This document is part of the Research and Innovation Action “Quality Translation 21 (QT21)”. This project has received funding from the European Union’s Horizon 2020 program for ICT under grant agreement no. 645452.

QT21

Deliverable D2.3

Final Report: Morphologically Rich Languages

Jan Niehues (KIT), Thanh-Le Ha (KIT), Jan-Thorsten Peter (RWTH), Rico Sennrich (UEDIN), Philip Williams (UEDIN), Stella Frank (UvA), Khalil Sima’an (UvA), Ondřej Bojar (CUNI), Tom Kocmi (CUNI), Dominik Macháček (CUNI), Tomáš Musil (CUNI), Dušan Variš (CUNI), Franck Burlot (LIMSI-CNRS), François Yvon (LIMSI-CNRS), Inguna Skadina (TILDE), Daiga Deksne (TILDE)

Dissemination Level: Public

Final (Revision 1.0), 26th January, 2018
<table>
<thead>
<tr>
<th>Grant agreement no.</th>
<th>645452</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project acronym</td>
<td>QT21</td>
</tr>
<tr>
<td>Project full title</td>
<td>Quality Translation 21</td>
</tr>
<tr>
<td>Type of action</td>
<td>Research and Innovation Action</td>
</tr>
<tr>
<td>Coordinator</td>
<td>Prof. Josef van Genabith (DFKI)</td>
</tr>
<tr>
<td>Start date, duration</td>
<td>1st February, 2015, 36 months</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>Public</td>
</tr>
<tr>
<td>Contractual date of delivery</td>
<td>31st January, 2018</td>
</tr>
<tr>
<td>Actual date of delivery</td>
<td>30th September, 2016</td>
</tr>
<tr>
<td>Deliverable number</td>
<td>D2.3</td>
</tr>
<tr>
<td>Deliverable title</td>
<td>Final Report: Morphologically Rich Languages</td>
</tr>
<tr>
<td>Type</td>
<td>Report</td>
</tr>
<tr>
<td>Status and version</td>
<td>Final (Revision 1.0)</td>
</tr>
<tr>
<td>Number of pages</td>
<td>223</td>
</tr>
<tr>
<td>Contributing partners</td>
<td>KIT, RWTH, UEDIN, UvA, CUNI, LMSI-CNRS, TILDE</td>
</tr>
<tr>
<td>WP leader</td>
<td>KIT</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Jan Niehues (KIT), Thanh-Le Ha (KIT), Jan-Thorsten Peter (RWTH), Rico Sennrich (UEDIN), Philip Williams (UEDIN), Stella Frank (UvA), Khalid Sima’an (UvA), Ondřej Bojar (CUNI), Tom Kocmi (CUNI), Dominik Macháček (CUNI), Tomáš Musil (CUNI), Dušan Varš (CUNI), Franck Burlot (LIMSI-CNRS), François Yvon (LIMSI-CNRS), Inguna Skadina (TILDE), Daiga Deksne (TILDE)</td>
</tr>
<tr>
<td>EC project officer</td>
<td>Susan Fraser</td>
</tr>
</tbody>
</table>

The partners in QT21 are:
- Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Germany
- Rheinisch-Westfälische Technische Hochschule Aachen (RWTH), Germany
- Universiteit van Amsterdam (UvA), Netherlands
- Dublin City University (DCU), Ireland
- University of Edinburgh (UEDIN), United Kingdom
- Karlsruher Institut für Technologie (KIT), Germany
- Centre National de la Recherche Scientifique (CNRS), France
- Univerzita Karlova v Praze (CUNI), Czech Republic
- Fondazione Bruno Kessler (FBK), Italy
- University of Sheffield (USFD), United Kingdom
- TAUS b.v. (TAUS), Netherlands
- text & form GmbH (TAF), Germany
- TILDE SIA (TILDE), Latvia
- Hong Kong University of Science and Technology (HKUST), Hong Kong

For copies of reports, updates on project activities and other QT21-related information, contact:
Prof. Stephan Busemann, DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
Phone: +49 (681) 85775 5286
Fax: +49 (681) 85775 5338

Copies of reports and other material can also be accessed via the project’s homepage:
http://www.qt21.eu/

© 2018, The Individual Authors

No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval.
system, without permission from the copyright owner.
## Contents

1 Executive Summary ................................................. 6

2 Word Representation in Phrase-based Machine Translation .......... 7
   2.1 Producing Unseen Morphological Variants in Statistical Machine Translation .................................................. 7
   2.2 Compound ....................................................... 7
   2.3 Word Segmentation to Address Sparsity ........................................ 7
   2.4 Source Words Representation in Phrase-based Machine Translation .................................................. 8
   2.5 Phrase Table Filtering for English-Latvian Translation System .................................................. 8

3 Word Representation in Neural Networks ........................................ 9
   3.1 Factored Representation .......................................... 9
      3.1.1 Neural Network Translation Models ........................................ 9
      3.1.2 Using Factored Word Representation in Neural Language Models .................................................. 10
      3.1.3 Neural Language Models with Morphology ........................................ 10
      3.1.4 Using Linguistically Motivated Input Factors in Neural Machine Translation .................................................. 10
   3.2 Word Segmentation ................................................ 11
      3.2.1 Neural Machine Translation of Rare Words with subword Units .................................................. 11
      3.2.2 SubGram ....................................................... 11
      3.2.3 Inferring Morphemes using Character-based Neural Network Language Models .................................................. 12
      3.2.4 Morphological Segmentation for Neural Machine Translation .................................................. 12
   3.3 Morphology Normalisation .......................................... 13
      3.3.1 Output Re-inflection for Rich Morphology ........................................ 13
      3.3.2 Rich Morphology normalisation ........................................ 13
      3.3.3 Morphology Normalisation in Factored Neural Machine Translation .................................................. 14
   3.4 Learning Word Representation using Multi-task Learning .................................................. 14
      3.4.1 Exploiting Linguistic Resources for Neural Machine Translation using Multi-task Learning .................................................. 14
      3.4.2 Multi-lingual Word Representations ........................................ 15
   3.5 Multi-Word Expressions .......................................... 15

4 Modelling Morphological Agreement ...................................... 17
   4.1 A Dependency Model of Morphological Structure ...................... 17
   4.2 Source Discriminative Word Lexicon .................................. 17
   4.3 Interaction Between Morphology and Word Order ...................... 17
   4.4 Pronoun Translation .............................................. 18
   4.5 Unification-based Constraints ....................................... 19
   4.6 Correction of Agreement Errors in Machine Translation Output ...................... 19

5 Morphology Prediction Evaluation ...................................... 20
   5.1 Automatic Test Suite Generation and Evaluation to Measure the Morphological Competence of a MT System .................................................. 20
   5.2 PROTEST: A Test Suite for Evaluating Pronouns in MT ...................... 20
   5.3 Assessing Morphological Agreement Using Contrastive Translation Pairs .................................................. 21

References ........................................................................... 21

A Annex ............................................................................ 27
   A.1 Linguistic Input Features Improve Neural Machine Translation ...................... 27
   A.2 Neural Machine Translation of Rare Words with Subword Units ...................... 36
   A.3 Neural Language Models with Morphology ........................................ 47
      A.3.1 Representing Morphology ........................................ 47
      A.3.2 Evaluation .............................................. 49
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.3.3 Results</td>
<td>51</td>
</tr>
<tr>
<td>A.3.4 Conclusion</td>
<td>52</td>
</tr>
<tr>
<td>A.4 SubGram: Extending Skip-Gram Word Representation with Substrings</td>
<td>54</td>
</tr>
<tr>
<td>A.5 Source Words Representation in Phrase-based Machine Translation</td>
<td>62</td>
</tr>
<tr>
<td>A.5.1 Combined Input</td>
<td>62</td>
</tr>
<tr>
<td>A.5.2 Hidden Combination</td>
<td>62</td>
</tr>
<tr>
<td>A.5.3 Evaluation</td>
<td>63</td>
</tr>
<tr>
<td>A.6 Morphology-aware Alignments</td>
<td>64</td>
</tr>
<tr>
<td>A.7 Output Re-inflection for Rich Morphology</td>
<td>72</td>
</tr>
<tr>
<td>A.8 Rich Morphology normalisation</td>
<td>80</td>
</tr>
<tr>
<td>A.9 Word Representations in Factored NMT</td>
<td>92</td>
</tr>
<tr>
<td>A.10 Morphological Evaluation</td>
<td>104</td>
</tr>
<tr>
<td>A.11 Phrase Table Filtering for English-Latvian Translation System</td>
<td>117</td>
</tr>
<tr>
<td>A.11.1 Filtering a Phrase Table</td>
<td>117</td>
</tr>
<tr>
<td>A.11.2 Modifying a Symmetrized Word Alignment File</td>
<td>117</td>
</tr>
<tr>
<td>A.11.3 Modifying a Phrase Extraction Module</td>
<td>118</td>
</tr>
<tr>
<td>A.11.4 Modifying Alignment Files prior to Symmetrization</td>
<td>119</td>
</tr>
<tr>
<td>A.12 Source Discriminative Word Lexicon</td>
<td>119</td>
</tr>
<tr>
<td>A.12.1 Structural Features</td>
<td>120</td>
</tr>
<tr>
<td>A.12.2 Word Representation</td>
<td>121</td>
</tr>
<tr>
<td>A.13 Paying Attention to Multi-Word Expressions in Neural Machine Translation</td>
<td>123</td>
</tr>
<tr>
<td>A.14 Producing Unseen Morphological Variants in Statistical Machine Translation</td>
<td>133</td>
</tr>
<tr>
<td>A.15 Exploiting Linguistic Resources for Neural Machine Translation using Multi-task Learning</td>
<td>140</td>
</tr>
<tr>
<td>A.16 Using Factored Word Representation in Neural Network Language Models</td>
<td>150</td>
</tr>
<tr>
<td>A.17 Automatic Post-Editing for the DiscoMT Pronoun Translation Task</td>
<td>159</td>
</tr>
<tr>
<td>A.18 Analysing ParCor and its Translations by State-of-the-art SMT Systems</td>
<td>166</td>
</tr>
<tr>
<td>A.19 PROTEST: A Test Suite for Evaluating Pronouns in Machine Translation</td>
<td>175</td>
</tr>
<tr>
<td>A.20 Edinburgh’s Statistical Machine Translation Systems at WMT 2016</td>
<td>183</td>
</tr>
<tr>
<td>A.21 A Joint Dependency Model of Morphological and Syntactic Structure for Statistical Machine Translation</td>
<td>195</td>
</tr>
<tr>
<td>A.22 How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs</td>
<td>202</td>
</tr>
<tr>
<td>A.23 Morphological Segmentation for Neural Machine Translation</td>
<td>209</td>
</tr>
<tr>
<td>A.24 KIT-Multi: A Translation-Oriented Multilingual Embedding Corpus</td>
<td>214</td>
</tr>
<tr>
<td>A.25 CUNI NMT System for WAT 2017 Translation Tasks</td>
<td>218</td>
</tr>
</tbody>
</table>
1 Executive Summary

This deliverable describes our work on morphologically rich languages within the QT21 project and it will make reference to the work done during the first reporting period.

QT21’s focus on morphologically rich languages (MRL) has been limited to German, Czech, and Latvian, covering 3 important European language families. Some of the work done for this deliverable began before the start of QT21 and was done on other morphologically rich languages as those chosen for QT21. As the work done on those languages (e.g. Russian, Finnish) is very relevant to QT21 and in order not to stop the analysis done, we decided during the first reporting period to finalise this work before applying it to the QT21 languages. In order to improve the translation quality for QT21 languages, new ways to represent words in a translation system were studied, along with modifications in the modelling of word agreement and methods to evaluate the approaches.

At the beginning of the project, we investigated several techniques to improve the representation of MRL within phrase-based statistical machine translation (PBMT) systems. Intrinsically, MRLs need larger vocabulary sizes, have higher out-of-vocabulary rates and model size increases. The work in this area is described in Chapter 2.

During the project, the shift to Neural Machine Translation (NMT) helped to significantly improve translation quality especially for MRLs. While building a state-of-the-art system, it is important to create a suitable vocabulary for the neural network. Inspired by techniques successfully used in PBMT, we developed several word representation techniques as described in Chapter 3. These techniques are essential for the success of NMT especially for MRLs. We developed methods using factored word representations, and further analysed different word segmentation strategies, performed morphology normalisation and used multi-task learning.

We also improved the modelling of the word agreement, as described in Chapter 4, as this is especially important for MRLs. Based on the work in the first period, which mainly focused on models in phrase-based and syntax-based models, we continued this work and extended it to correct agreement errors with automatic post-editing. Furthermore, the modelling of pronoun translation was addressed.

Finally, in order to make sure that all this work on MRLs is going in the right direction, we investigated methods to evaluate the morphological competence of a machine translation system.
# Word Representation in Phrase-based Machine Translation

Especially in the beginning of the project we investigated several approaches to represent words in PBMT systems. In addition to the work reported already in the previous deliverable D2.1 (here Sections 2.2 to 2.5), we investigated methods to produce unseen morphological variants as described in Section 2.1.

## 2.1 Producing Unseen Morphological Variants in Statistical Machine Translation

Even with very large training data, morphologically rich languages often lead to sparsity issues and can never provide all possible word forms.

In Huck et al. [31], we extend the log-linear model of PBMT implemented in Moses with a discriminative model to improve the scoring of considered translation options. In contrast to common n-gram language models, this discriminative model can rely on an arbitrary mix of information sources from the whole source sentence and a few preceding words from the target candidate translation. We experiment with target units of length one (words) or longer (minimum translation units (MTU)) and we use a morphological generator to generate all word forms permissible by the morphological system of our language of interest (Czech, in this case).

One of the benefits of the discriminative model is that it can be conditioned on all possible morphological features of words excluding the actual word form. This setting allows us to score even unseen word forms relatively reliably. Gains in BLEU scores are observed from small (50K sentence pairs; +0.9 BLEU) to very large (50M sentence pairs; +0.3 BLEU) training data conditions.

The work by Huck et al. [31] is reproduced in Annex A.14.

## 2.2 Compounds

At UvA, our research (published as Daiber et al. [13]) looked into using distributional semantics for derivational morphology. Dealing with productive word formation processes, such as noun compounding, is a key issue for MT. Compound splitting is a common preprocessing step when translating from languages with productive compounding, for example German or Hungarian; commonly used approaches use surface features of the word string, such as component frequency [36], or morphological analysers [21]. Distributional semantics captures the usage of the compound as well as proposed components and represent them in high dimensional semantic space, such that analogical relations like “handmade is to made as handwriting is to writing” hold. Exploiting these to identify compound splits can be done in a completely unsupervised way [50] and leads to far more accurate splits than the similarly unsupervised frequency-based methods, as well as improved translation performance between de-en (approx. 0.5 BLEU) when splitting is applied to OOV words only.

## 2.3 Word Segmentation to Address Sparsity

At the LIMSI-CNRS, a pilot study was run with the intention to improve translation from and into Finnish. The agglutinative character of Finnish morphology makes the task of translating into English very challenging, notably due to the very high number of forms that are rare or even unseen in the training set. These difficulties are common with other morphologically rich languages in which many forms correspond to one single English word. Finnish, in this respect, implies an increased difficulty, since it presents a higher quantity of possible morpheme combinations, compared to inflectional European languages, such as Czech or Latvian. Such complex forms plague the word alignments and generate a lot of noise during the translation process. A simple workaround is to segment the Finnish text into smaller units that make both languages look more similar. However, finding the right level of segmentation of Finnish words into morpheme-like units so as to produce more robust alignments with English is challenging, and can probably not be performed by looking at only the Finnish side. These early experiments were performed in the context of the WMT 2015 [38].
2.4 Source Words Representation in Phrase-based Machine Translation

One of the main problems in translating from a morphologically rich language is the number of Out-Of-Vocabulary (OOV) words. Every word stem has many different surface forms and many of them will only be seen rarely. When translating into a less morphologically rich language we often do not require all the morphological information. Therefore, KIT developed and evaluated different techniques to integrate word stemming into the translation system. While only relying on the word stems hurts the performance of the MT system, we could improve the translation of rare words by using a combination of stemmed and surface word representation in a Phrase-based SMT system. By using these methods, it is still easy to integrate any advanced model into the log-linear combination. A detailed description of the work can be found in Annex A.3.

2.5 Phrase Table Filtering for English-Latvian Translation System

The phrase tables built during the training process of a PBMT system are big in size and tend to contain a large portion of low quality data, which influences the quality of translation. For instance, the size of the phrase table of the English-Latvian general domain PBMT system trained on 4,874,892 unique parallel sentences is 17.5 GB. It contains 143,894,027 lines (bilingual pairs of phrases). Due to the rich morphology of the Latvian language, many phrases in the phrase table have several thousands of translations. In our experiments we made changes to the different modules and outputs of the Moses training pipeline to improve translation into the morphologically rich Latvian language:

- an extra function in the phrase table creation module was added to filter out some entries of the phrase table;
- the phrase extraction module was modified;
- the alignment file was modified prior to symmetrisation;
- the symmetrised alignment file was modified.

Although experiments described in this deliverable were performed for the English-Latvian pair, we believe that they are also applicable to other morphologically rich under-resourced languages. More details can be found in Annex A.11.
3 Word Representation in Neural Networks

The major focus in modelling MRLs in the second part of the project has been on word representations in neural networks. Based on the successful attempts in the first part of the project (Sections 3.1.1, 3.2.1, 3.2.2, 3.2.3, 3.3.1), different word representation techniques were further developed during the second reporting period of the project. The work on this topic can be grouped into the following lines of research: factored representations, word segmentation, morphology normalisation, as well as multi-task learning.

3.1 Factored Representation

One promising way to improve the word representation in neural networks is to explicitly encode additional information about the word. This can be linguistic information like PoS-tags as well as automatically learned representations like word classes. The idea was already successfully used in many PBMT translation systems and within the project we could show the potential of this approach also for neural network models.

In D2.1, we already reported good initial results for neural translation models used as a feature in an SMT system (Section 3.1.1). We continued this research and also successfully applied it to neural language models within a SMT system and in a purely neural machine translation system.

3.1.1 Neural Network Translation Models

Based on the observation on one-hot encoding that two words which only differ by a letter are treated the same way as two words that have nothing in common, one of the first attempts to enhance word encodings was to split words in prefix, body and suffix. RWTH tested prefix, suffix and class features as additional input features to the one-hot encoding (unpublished). The experiments were conducted on the German to English IWSLT 2013 translation task. The basic setup is a PBMT system similar to the system used in Wuebker et al. [57]. We reranked 500-best lists with an NMT similar to the Neural Network Joint Model in Devlin et al. [17] without the language model part and used this as our strong baseline.

In the first experiment on improving the baseline + NMT system by adding a prefix feature containing the first three letters ($P_3$), a suffix feature containing the last two letters ($S_2$) and the rest ($R$) which was not covered by the prefix and suffix in addition to the one-hot encoding. This resulted in a TER improvement of 0.3% absolute on the evaluation set.

In the second experiment we provided the neural network model with more information from the MRL by using a clustering algorithm to create word classes. These word classes can be used in addition to the word feature to get more reliable inputs since the classes are seen more frequently than the individual words. This yielded an absolute improvement of 0.2 BLEU points compared to the NMT model and a TER improvement of 0.3% on the evaluation set (Table 1).

<table>
<thead>
<tr>
<th>IWSLT</th>
<th>test</th>
<th>eval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU [%]</td>
<td>TER [%]</td>
</tr>
<tr>
<td>Phrase-based</td>
<td>30.7</td>
<td>49.1</td>
</tr>
<tr>
<td>+ NNTM (Baseline)</td>
<td>31.9</td>
<td>47.4</td>
</tr>
<tr>
<td>+ $P_3$-$R$-$S_2$</td>
<td>32.1 (+0.2)</td>
<td>47.2 (-0.2)</td>
</tr>
<tr>
<td>+ Class Features</td>
<td>32.1 (+0.2)</td>
<td>47.0 (-0.4)</td>
</tr>
</tbody>
</table>

Table 1: Experimental results of neural network translation model with word morphology and class features. The German to English IWSLT 2013 translation task was used for these experiments. test is the evaluation set from 2010 and eval the evaluation set from 2011. A bold font indicates results that are significantly better than the baseline system (Phrase-based + NNTM) with $p < 0.05$. 

---
3.1.2 Using Factored Word Representation in Neural Language Models

Neural networks have recently shown their great potential for language and translation modelling, specifically improving translation performance over PBMT. As word representation using different form factors has been successfully applied to PBMT, KIT decided to combine the idea with neural language models.

KIT combined these two ideas by investigating the combination of both techniques (NN and word factorisation). By representing words in neural network language models using different factors, we were able to improve the models themselves measured in perplexity, as well as their impact on the overall machine translation performance. This is especially helpful for MRLs as they have a large vocabulary size. Furthermore, it is easy to add additional knowledge to the model, such as source-side information.

Using this model we improved the translation quality of a state-of-the-art PBMT system by 0.7 BLEU points. We performed experiments on three language pairs for the WMT 2016 news translation task. A detailed description can be found in Annex A.16.

3.1.3 Neural Language Models with Morphology

Language models play an important role in many natural language processing tasks. In this work, we examined the possibilities of using morphological annotations within Neural Language Models (NLM). Various architectures of neural language models have been analysed, both theoretically and empirically. To take advantage of annotation tools and annotated datasets, we proposed a neural network architecture for a language model that explicitly makes use of morphological annotation of the input sentence: instead of word forms it processes lemmas together with morphological tags. Both the baseline and the proposed method were evaluated on their own by perplexity, and also in the context of machine translation by the means of automatic translation quality evaluation.

As opposed to simple factored neural language models, we decided not to embed the morphological annotation, but to represent it as a binary vector (concatenation of one-hot vectors for each feature in the morphological tag).

As a result our language models obtained a significantly lower perplexity. Unfortunately, when we employed this improved NLM to rescore n-best lists of Moses PBMT, we observed no statistically significant gains in BLEU.

Our results suggest that there is a potential to improve machine translation by including morphological information into language models. Recent developments in machine translation seem to favour neural machine translation fed by various subword types. For future experiments, we would thus recommend combining subword encoding with explicit morphological information and independently, we already saw such experiments being carried out [52].

Our work on neural language models with morphology is described in detail in Annex A.3.

3.1.4 Using Linguistically Motivated Input Factors in Neural Machine Translation

Taking inspiration from prior work in language modelling and statistical machine translation, UEDIN explored the use of factored word representations in fully neural machine translation models. Starting from the widely-used attentional encoder–decoder architecture, the embedding layer of the encoder was generalised to support the inclusion of arbitrary input factors. In this first study, UEDIN added several factors to supplement the original surface form, including factors for morphological features, part-of-speech tags, and syntactic dependency labels. These were included as input factors to English→German and English→Romanian neural machine translation systems. In experiments on WMT16 training and test sets, it was found that linguistic input factors improved model quality according to the three evaluation metrics: perplexity, CHRF3, and BLEU. For the latter, the improvements from using all proposed features were 1.5 BLEU for German→English, 0.6 BLEU for English→German, and 1.0 BLEU for English→Romanian. Further experiments to isolate the effects of individual features showed that the gain from different features was not fully cumulative since there is an overlap in the
Table 2: Experimental results of neural machine translation with subword units (BLEU scores).

<table>
<thead>
<tr>
<th>system</th>
<th>en-de</th>
<th>en-ru</th>
</tr>
</thead>
<tbody>
<tr>
<td>phrase-based</td>
<td>22.8</td>
<td>24.3</td>
</tr>
<tr>
<td>NMT baseline (large-vocabulary model with backoff dictionary)</td>
<td>24.2</td>
<td>22.8</td>
</tr>
<tr>
<td>character bigrams</td>
<td>25.3</td>
<td>24.1</td>
</tr>
<tr>
<td>byte pair encoding</td>
<td>24.7</td>
<td>24.1</td>
</tr>
</tbody>
</table>

information encoded in different features. For instance, both the dependency labels and the morphological features encode the distinction between German subjects and accusative objects, the former through different labels (subj and obj), the latter through grammatical case (nomina- tive and accusative).

This work is described in [48], which can be found in Annex A.1. The work was developed jointly within WP1 and WP2 due to its applicability to both the problems of semantically-annotated representation of language and morphologically-aware word representations. It is therefore also reported in deliverable D1.2.

### 3.2 Word Segmentation

A second line of research on word representation investigated different word segmentation strategies. Motivated by successful approaches reported in the first period (Section 3.2.1, 3.2.2, and initial experiments of Section 3.2.3), we continued this work. We explored statistical methods as well as methods that also take linguistic information into account.

#### 3.2.1 Neural Machine Translation of Rare Words with subword Units

Neural machine translation (NMT) models typically operate with a fixed vocabulary but translation is an open-vocabulary problem. Until now, translation of out-of-vocabulary words was performed using back-off models. UEDIN introduced a simpler and more effective approach, making the NMT model capable of open-vocabulary translation by encoding rare and unknown words as sequences of subword units, based on the intuition that various word classes are translatable via smaller units than words, for instance names (via character copying or transliteration), compounds (via compositional translation), and cognates and loanwords (via phonological and morphological transformations). UEDIN investigated the suitability of different word segmentation techniques, including simple character n-gram models and a segmentation based on the byte pair encoding (BPE) compression algorithm. UEDIN showed empirically that subword models improve over a back-off dictionary baseline for the WMT 2015 translation tasks English-to-German and English-to-Russian by up to 1.1 and 1.3 BLEU, respectively (Table 2).

This work was described in [49], which can be found in Annex A.2.

#### 3.2.2 SubGram

One of the popular vector embeddings for words is word2vec based on the so-called Skip-gram model [39]. Before applying this model to machine translation, CUNI wanted to avoid one of the clear limitations of the model for morphologically rich languages: treating word forms as atomic units.

CUNI proposed a substring-oriented extension of Skip-gram model called “SubGram” which induces vector embeddings from character-level structure of individual words. This approach gives the NN more information about the examined word without any drawbacks in data sparsity or reliance on explicit linguistic markup. In future work, we will compare this totally linguistically uninformed approach with a setup where morphological features are provided to the NN in various ways.

We evaluate the new model on the same set of “semantic” and “syntactic” word tuples created by Mikolov et al. [39]. Each tuple checks if the model is able to identify the word related
to a given input word in the same way as another pair of words suggests. For example, given the pair “boy–girl” and the query “father”, the model is expected to find the word “mother” without any supervision and instead inducing it from observing word co-occurrences in a monolingual corpus.

While the “semantic” relations are idiomatic and not related to the spelling of the words, the “syntactic” part of the test set covers several phenomena clearly described by the processes of word formation and our SubGram can benefit from word-form similarities. As shown in the experiments, the technique is able to improve the embeddings for morphologically rich languages. It is possible to use these word embeddings in NMT, but we did not pursue this line of research.

This work was described in [34], which is included in Annex A.4.

3.2.3 Inferring Morphemes using Character-based Neural Network Language Models

Earlier work shows that language models over characters, rather than words, have the potential to alleviate the problems of large vocabulary/data sparsity that arise when dealing with morphologically rich languages. In [19], UvA investigated whether neural languages models over characters implicitly capture morphemes, and how they can then be used to generate new words, i.e. via novel combinations of latent morphemes. Initial experiments were performed on languages with highly-concatenative morphology, in order to evaluate on segmentation points (English, Finnish, and Turkish). However, by the end of this project the experiments on unsupervised learning of morphemes for languages with more syncretic morphology (e.g., German, Czech) showed mixed results and this work was discontinued.

3.2.4 Morphological Segmentation for Neural Machine Translation

The current state of the art of word handling in NMT is to break words into short character sequences. The exact breaks are determined automatically based on corpus statistics, disregarding morphological properties of words. Also, the exact marking of the newly introduced boundaries can affect the performance of the method.

We compared several types of subword boundary marks for byte-pair-encoding for German and Czech NMT, ambiguating or disambiguating the first, middle and last subword of a word. According to our results, the differences are not remarkable, but adding “@@” to every left side of a subword split seems to be the best. This setting is the default behaviour of BPE implementation by Sennrich et al. [49].

Then we compared the Sennrich et al.’s BPE implementation and Google’s open-sourced toolkit Tensor2Tensor, namely the SubwordTextEncoder class and found out the latter one gives about 4.9(!) BLEU points better results than default BPE, but we also showed that BPE can be slightly modified to give nearly the same performance as SubwordTextEncoder. Adding an underscore to the end of every word (as a “zero suffix mark”) before the BPE learning phase and sharing vocabulary increased BLEU about 4.5 points.

To exploit morphologically-motivated word segmentation, we used Morfessor and DeriNet morphology segmentation combined with SubwordTextEncoder to introduce subword boundaries reflecting morphological composition of words. Unfortunately, we didn’t see any improvements to plain SubwordTextEncoder with the settings we used.

As a part of our submission to the Workshop on Asian Translation 2017 (WAT 2017), we also explored various segmentation strategies when training Japanese-English translation NMT systems. Japanese texts (and texts in other Eastern Asian languages) are known to lack explicit word segmentation. Given that the widely used byte-pair encoding (BPE) algorithm can easily handle the segmentation without prior preprocessing, we examined whether an explicit segmentation performed by a separate tokeniser can help produce better subword units which can benefit the overall performance of the NMT system. The results showed that tokenisation
of Japanese sentences before applying BPE can significantly improve the target side BLEU (around 2 BLEU points). On a side note, we also tried to transcribe Japanese characters into Latin characters, however, this seemed to be harmful to system performance.

The details on these experiments are available in Appendices A.23 and A.25, the second of which was published as Kocmi et al. [35].

### 3.3 Morphology Normalisation

Morphologically rich languages exacerbate the well-known issues incurred by low-resource language conditions. Sparsity problems are worsened when a lexical unit has many morpho-syntactic realisations as different word forms, leading to many out-of-vocabulary words or inaccurately-estimated translation probabilities. To mitigate such issues, LIMSI has been considering factored representations of a word consisting of a lemma and a PoS-tag. This representation was first intended to learn better alignments between a morphologically rich language and an analytical language like English. This setup enables the alignment of a case marker (e.g. genitive of a noun) with its English translation as a preposition (e.g. of). In a phrase-based MT framework, this alignment procedure resulted in local improvements and considerably lighter phrase tables (see A.6).

Further study on the subject led to consideration that some information within an MRL has no counterpart in English. A way to reduce data sparsity then consists in removing this “redundant” part of the rich morphology from within the MRL, which we call normalisation.

#### 3.3.1 Output Re-inflection for Rich Morphology

LIMSI has investigated a two-step MT approach for English-to-Czech [18, 20], where all target data is normalised, i.e. morphological information that can not be predicted from the source is removed (such as case information for nouns). The MT system trained on such data thus computes translations from English into a normalised version of Czech. In the second step, the missing morphological attributes are predicted in order to generate fully inflected Czech using a dictionary containing the paradigms of all (known) Czech words. Using this dictionary provides the MT system with more word forms than is observed at training time.

Various models have been investigated for re-inflection and the impact of different amounts of data on translation quality has been evaluated [6, 7]. Re-inflection based on conditional random fields (CRFs) works well on small amounts of parallel data: our experiments showed that re-inflection in a system trained on 117k sentences from TED Talk data gives an improvement of 1.5 BLEU. The use of a dictionary enabled the generation of 12.45% word types unseen in the training set (3.05% tokens) at test time. When larger training data is available, other methods need to be applied, such as the use of n-gram language models. See the publication on this work in A.7.

#### 3.3.2 Rich Morphology normalisation

The normalisation of source-side morphologically rich languages usually consists of hand-crafted rules to remove morphological information that is redundant with respect to the target language. LIMSI developed a new method to automate the process based on the entropy of English word distributions when aligned for example to morphologically rich Czech words. The proposed model overcomes time and resource challenges incurred by writing manual rules and improves the MT output. The goal of this approach is to perform a clustering of source word forms, merging those words that translate into the same target word(s). We assume that each source word form is a combination of a lemma, a part of speech (PoS) and a sequence of morphological tags, and that the parallel corpus has been aligned at the word level. Using these alignments, it is possible to compute lexical translation probabilities \( p(e|f) \) and unigram probabilities \( p(f) \), which are inputs to our algorithm. These counts are first used for the computation of the conditional entropy of the translation model where each individual word form is a singleton
cluster. Then, proceeding in a bottom-up fashion, we look for pairs of clusters, the merging of which reduces the conditional entropy. This search for the optimal normalisation is a variant of hierarchical clustering.

Experiments were run for SMT systems from Czech into English (1M parallel sentences), showing that manual normalisation gives an improvement of 0.61 BLEU, while automatic normalisation almost doubles this improvement (+1.0 BLEU). This normalised representation of rich morphology enables a more efficient use of few parallel data, providing an improvement of 1.6 BLEU for a system trained on only 190k parallel sentences. For the translation into a morphologically rich language, LIMSI has developed a neural model that considers the source (English) sentence, the normalised target (Czech) sentence and predicts the morphology information that was removed during normalisation. This model provides significant improvements over strong baselines (>0.8 BLEU). See the publication on this work in A.8.

### 3.3.3 Morphology Normalisation in Factored Neural Machine Translation

The latter two studies have been reported during the first reporting period. The two-step approach above contains an important drawback: the predictions of lexical units and morphological information are strictly independent. To avoid this over-simplistic hypothesis, LIMSI has been investigating multi-task learning in an NMT framework, where both predictions are simultaneous and joint. When translating into a morphologically rich language, the model predicts, at each time-step in the output generation, a normalised word and a fine-grained PoS-tag.

The MT model thus predicts two sequences of the same length: one consists of normalised words and the second of fine-grained PoS-tags. A last step is required to predict final target word forms, using a dictionary. One issue is that this dictionary often returns several word forms when given a specific lemma and PoS-tag. To select the right word form in this list, we have investigated different techniques. First, a simple unigram model on words can be used to select the most frequent word. Secondly, LIMSI has designed a neural network architecture that takes as input lemmas, PoS-tags, and lists of candidate word forms. The network decomposes these candidate word forms into characters using a convolutional filter and returns a score for each one of them. However, the latter method did not show any improvement over the naive unigram model.

Experiments have been conducted for English-to-Czech and English-to-Latvian, and systems for both languages were submitted at WMT’17. The multi-task approach provided significant improvements over a vanilla NMT setup. The best results obtained over a state-of-the-art system applied normalisation to subword units (section 3.2.1), which shows an efficient pre-processing combination for rich morphology generation. It was finally shown that predicting normalised words resulted in higher performance than when predicting bare lemmas, which proves that removing strictly all the rich morphology from the target language during normalisation is not a good solution and that there is a subset of morphological information that should be kept. See the publication on this work in A.8, reported in A.9.

### 3.4 Learning Word Representation using Multi-task Learning

In addition, the project also explored ways to use several tasks to learn better word representation. This was done for different NLP tasks as well as training machine translation systems on several language pairs.

#### 3.4.1 Exploiting Linguistic Resources for Neural Machine Translation using Multi-task Learning

KIT could show that multi-task learning is a successful and an easy approach to introduce additional knowledge into an end-to-end neural attentional model. By jointly training several natural language processing (NLP) tasks in one system, we are able to leverage common information and improve the performance of the individual task. Thereby, the representations learned within the neural network are able to better learn properties of the words.
KIT analysed the impact of three design decisions in multi-task learning: the tasks used in training, the training schedule, and the degree of parameter sharing across the tasks, which is defined by the network architecture. The experiments are conducted for a German to English translation task. As additional linguistic resources, we exploit PoS information and named-entities (NE). Experiments show that the translation quality can be improved by up to 1.5 BLEU points under low-resource conditions. The performance of the PoS tagger is also improved using the multi-task learning scheme. A detailed description can be found in (see Appendix A.15).

3.4.2 Multi-lingual Word Representations

Cross-lingual word embeddings are the representations of words across languages in a shared continuous vector space. Cross-lingual word embeddings have been shown to be helpful in the development of cross-lingual natural language processing tools. There have been various methods of cross-lingual embedding induction being proposed, but most of them are essentially bilingual in the perspective that they learn to induce bilingual embeddings from bilingual data. Basically these methods optimise some cross-lingual constraints so that the semantic similarity between words corresponds to the closeness of these representations in a common vector space. Consequently, if they need cross-lingual embeddings for a new language pair, they must apply their inducing method on that new bilingual data. Furthermore, there would be some domain mismatch between the new acquired embeddings and the others if the new bilingual data are from a different domain. The aforementioned limitations of those cross-lingual corpora motivates us to design a multilingual embedding inducing method from a single corpus which is available in as many languages as possible.

In this work, KIT proposed such an approach utilising a multilingual neural machine translation (NMT) system to constrain the embeddings from n source languages while translating into the same target language (as we call it, multi-source NMT). The source embeddings employed in this model were implicitly forced to learn the common semantic regularities in order to maximise the translation quality of every language pair in the system. Once the multi-source NMT model is trained to a good state, the source word embeddings can be simply extracted from the model and used as a multilingual word embedding corpus. It has been shown that our multilingual corpus achieves competitive performances in standard evaluations and it also has better coverage while using ten times less data (around two hundred thousand from the TED corpus for each language pair versus two million from Europarl) for the training process.

The work will appear in LREC 2018; the abstract can be found as Annex A.24. The work was developed jointly within WP1 and WP2 due to its applicability to both the problems of deep semantic representations of language and translation involving under-resourced languages and multi-linguality.

3.5 Multi-Word Expressions

Correct treatment of multi-word expressions (MWE) in MT is conceptually very closely related to word segmentation strategies. MWEs are semantically atomic, so gains could be expected from handling them as units on their own. On the other hand, it is not uncommon that MWEs undergo certain form variations, especially in morphologically rich languages. Translating MWEs has been a challenge in PBMT which relies on very large automatic (multi-word) translation dictionaries. The allowable vocabulary size in NMT is much smaller and while NMT is generally substantially better at predicting the correct word forms, the translation of MWEs is still far from perfect.

In Rikters and Bojar [14], we thus investigated techniques of improving translation of MWEs in NMT. The methods are limited to data manipulation; we stick to the common sequence-to-sequence model with attention [1] as implemented in Neural Monkey [30]. We devise a modified training regime, where natural training sentences are mixed with segments consisting of automatically extracted MWEs, obtaining almost a 1.0 BLEU improvement on a subset of sentences containing MWEs in the source or 0.65 BLEU on the full Latvian newsdev2017.
The full paper is reprinted in Annex A.13.
4 Modelling Morphological Agreement

Working on morphologically rich languages also has to deal with complex word agreement. We therefore investigated different techniques which are devised to improve the agreement between the words on the target side. In the beginning of the project, we worked on modelling agreement in phrase-based MT. The work in Section 4.1 and Section 4.2 was already done in the first period and not continued due to the switch to NMT. The work in Section 4.3 and 4.4 was started in the first period and continued in the second.

In addition, UEDIN and CNU started to collaborate on using unification-based constraints for English-Czech translation and we used NMT to correct agreement errors in automatic post-editing.

4.1 A Dependency Model of Morphological Structure

When translating between two languages that differ in their degree of morphological synthesis, syntactic structures in one language may be realised as morphological structures in the other. In this case, PBMT models require a mechanism to learn such translations. Morpheme segmentation is commonly employed to allow the translation of morphologically complex words. In a flat representation, i.e. as a sequence of morphemes, however, the hierarchical structure of words is lost, which is especially damaging when translating into a morphologically complex language. For models operating on sequences of surface tokens, such as n-gram language models, splitting morphemes increases the distance between tokens which are inter-dependent. For example, this will be the case for the morpho-syntactic agreement between a determiner and the head of a German compound, which is the last segment at the same time. We wish to utilise the morphological structure to better model morpho-syntactic constraints and selectional preferences. In Sennrich and Haddow [47], UEDIN has presented a quasi-syntactic dependency annotation of selected morphological structures in German, which allows for a translation model that models the translation of both syntactic and morphological structures. It informs the dependency language model described previously in Sennrich [45] about the relationship between morphemes, and facilitates the modelling of agreement and selectional preferences involving morphologically complex words. Improvements in translation quality of up to 1.8 BLEU are achieved on the WMT English-to-German translation task.

This work is described in [47], which can be found in Annex A.21. The work was developed jointly within WP1 and WP2 due to its applicability to both the problems of semantics in shallow syntactic models and morphological representations. It is therefore also reported in deliverable D1.4.

4.2 Source Discriminative Word Lexicon

The correct word form in a morphologically rich language often has long-range dependencies. In order to better disambiguate the translations, KIT proposed modelling the translation as a prediction task. The source discriminative word lexicon was developed to predict the correct translation given the source word or contextual information. For every source word, a maximum entropy classifier was trained using local context features and dependency information. We investigated different features as well as different word representations. Using this additional information, we could improve the prediction of the correct morphological form of words in the target language. When adding the model to a phrase-based machine translation, the performance of the English to German translation task of IWSLT was improved by up to 0.6 BLEU points [26]. A detailed description of the technique can be found in Annex A.12.

4.3 Interaction Between Morphology and Word Order

Correctly modelling morphological agreement often requires knowledge about the structure of the entire sentence, which can also be used to improve word order in translation. UvA has
been investigating the interaction between morphology and word order for English-German translation. Based on the promising initial results in the first period, we continued this line of research in the second one. The focus has been on pre-ordering and on target morphology prediction on the source sentence before reordering and translation.

In [12], UvA explored a novel pipeline for translation into morphologically rich languages which consists of two steps: initially, the source string is enriched with target morphological features and then fed into a translation model which takes care of reordering and lexical choice that matches the provided morphological features. As a proof of concept we first show improved translation performance for a phrase-based model translating source strings enriched with morphological features projected through the word alignments from target words to source words. Given this potential, we present a model for predicting target morphological features on the source string and its predicate-argument structure, and tackle two major technical challenges: (1) how to fit the morphological feature set to training data and (2) how to integrate the morphology into the back-end phrase-based model such that it can also be trained on projected (rather than predicted) features for a more efficient pipeline. For the first challenge we present a latent variable model, and show that it learns a feature set with quality comparable to a manually selected set for German. And for the second challenge we present results showing that it is possible to bridge the gap between a model trained on a predicted and another model trained on a projected morphologically enriched parallel corpus. Finally we exhibit final translation results showing promising improvement over the baseline phrase-based system.

In [14], UvA study the relationship between word order freedom and pre-ordering in statistical machine translation. To assess word order freedom, we first introduce a novel entropy measure which quantifies how difficult it is to predict word order given a source sentence and its syntactic analysis. We then address pre-ordering for two target languages at the far ends of the word order freedom spectrum, German and Japanese, and argue that for languages with more word order freedom, attempting to predict a unique word order given only source clues is less justified. Subsequently, we examine lattices of n-best word order predictions as a unified representation for languages from across this broad spectrum and present an effective solution to a resulting technical issue, namely how to select a suitable source word order from the lattice during training. Our experiments show that lattices are crucial for enabling empirical performance improvements for languages with freer word order (English–German) and can provide additional improvements for fixed word order languages (English–Japanese).

In [15], UvA explore the novel idea of building a single universal reordering model from English to a large number of target languages. To build this model, we exploit typological features of word order and morphology for a large number of target languages together with source (English) syntactic features and we train this model on a single combined parallel corpus representing all (22) involved language pairs. The work contributes experimental evidence for the usefulness of linguistically defined typological features for building such a model. When the universal reordering model is used for preordering followed by monotone translation (no reordering inside the decoder), various experiments show that this pipeline gives comparable or improved translation performance to a phrase-based baseline for a large number of language pairs (12 out of 22) from diverse language families.

### 4.4 Pronoun Translation

Morphology is crucial to the translation of pronouns in morphologically rich languages. For instance, the choice of an anaphoric pronoun needs to agree in morphological realisation with the noun to which it refers (its antecedent) and the verb for which it is an argument. Agreement requirements such as this pose a problem for machine translation, not only within individual sentences, but also at the discourse level.

As part of a line of research in discourse-aware SMT, UEDIN conducted work on the specific problem of pronoun translation. This included the development of an automated post-editing system submitted to the shared task on pronoun translation at the 2nd Workshop on Discourse in Machine Translation [22]. The system categorises pronouns according to their function and
uses simple, hand-crafted rules to detect and amend pronoun translation in SMT output. With sub-baseline-performance observed for all shared task systems, pronoun translation remains an unsolved task. In order to better understand the problem, UEDIN conducted analyses of manual and automated translation [23]. The ParCor corpus [24] of parallel English-German texts and their annotations was used to categorise pronouns according to their function. UEDIN identified significant differences in pronoun usage between English and German for the anaphoric type in manual translation. In addition to this, they were able to make recommendations for further sub-categorisation of this function with respect to system development and evaluation.

This work is described in [22] and [23], which can be found in Annex A.17 and Annex A.18.

The transition to neural translation models raised the question of whether pronoun translation remained such a serious problem. UEDIN subsequently developed a test suite for evaluating pronoun translation (see Section 5.2). While pronoun translation remains an interesting problem, particularly at the discourse-level, it is a lesser concern in the era of neural translation models due to their ability to model intra-sentential dependencies and so we did not continue work on modelling pronoun agreement.

4.5 Unification-based Constraints

[55] and [54] have proposed the use of unification-based constraints as a means of improving morphological agreement in string-to-tree models. Continuing this line of work, UEDIN and CUNI collaborated to develop an English-Czech system with agreement and case government constraints. Specifically, the constraints were designed to enforce: i) case, gender, and number agreement between nouns and prenominal adjectival modifiers; ii) number and person agreement between subjects and verbs; iii) case agreement between prepositions and nouns; iv) use of nominative case for subject nouns. In preliminary experiments, small but consistent gains of between 0.1-0.3 BLEU were obtained. Previous analysis for German has shown that BLEU lacks sensitivity to grammatical improvements when compared to human evaluators and so CUNI carried out a small manual analysis of the submitted system with and without constraints. The comparison of 100 segments by a native Czech speaker found that while the use of hard constraints sometimes forced the system to select a worse translation, on average the constraints led to better translations.

This work is described in [56], which can be found in Annex A.20. The work was developed jointly within WP1 and WP2 due to its applicability to both the problems of semantics in shallow syntactic models and morphological agreement. It is therefore also reported in deliverable D1.4.

4.6 Correction of Agreement Errors in Machine Translation Output

In the WMT17 Automatic Post-Editing (APE) task, we focused on experimenting with neural network sequence-to-sequence architectures for automatic post-editing. These experiments were focused on post-editing in general and were trained using machine translated data that were post-edited by human translators. We did not focused on the specific phenomena, e.g. agreement errors.

As a part of the project Health in my Language (HimL), we experimented with a machine learning framework for correcting morphological errors, MLFix. These experiments mainly focused on post-editing of the statistical machine translation (SMT) outputs. We also applied trained models on the NMT output with negative results. For more detailed report on the MLFix experiments, please refer to the HimL project deliverables.

Recent reports on the analysis of the state-of-the-art NMT systems showed that these systems are very good at handling target-side morphology including agreement. This is due to the decoder being able to learn a strong target-side language model. These results suggest, that further development of automatic post-editing tools focused on correcting morphology might not be feasible in the future.
5  Morphology Prediction Evaluation

In addition to developing methods to improve the translation of morphologically rich languages, we also developed several approaches to perform a focused evaluation on morphological challenges.

5.1 Automatic Test Suite Generation and Evaluation to Measure the Morphological Competence of a MT System

Standard automatic metrics are useful to assess the overall quality of a sentence or a document. They, however, do not give any insights regarding the modelling of a specific phenomenon. In Work Package 2, one essential goal is to improve the morphological quality of MT outputs, and we needed a way to measure whether we actually are achieving this goal. To tackle this problem, LIMSI proposed a morphological evaluation method that reports on the performance of a system \( \text{wrt.} \) diverse linguistic phenomena. This method is based on automatically generated pairs of source sentences, where each pair tests one morphological contrast. Once the pair is translated, the method reports whether the MT output reflects the contrast introduced in the source.

The test suite is divided into three main sets of tests. The first one contains minimal pairs of sentences that differ in, e.g., the tense of a verb, or the number of a noun. It checks whether a morphological feature has been correctly conveyed from source to target (adequacy). The second one contains pairs where a pronoun has been changed to, e.g., a noun phrase. The evaluation then consists in checking if agreement is correctly modelled within the noun phrase (fluency). A third set of the test suite focuses on a word in the source and generates several of its hyponyms (or synonyms, antonyms). The test then consists in checking whether the MT system made the same morphological prediction for all the hyponyms. Indeed, when changing a single noun to a synonym in the source sentence, we should obtain different translations of this noun, but with the same inflection (e.g. same case mark). This last evaluation type tests the consistency of an MT system.

This evaluation method has been applied to all the systems submitted by QT21 partners at WMT’17 for English-to-Latvian and English-to-Czech (see publication [5] reproduced in A.10). LIMSI is currently working in collaboration with CUNI and others on new features to evaluate and on adding new language pairs (English to German, Finnish, Turkish and Kazakh). The resulting test suite will be submitted to the WMT’18 organizers with a proposal to integrate it in the official test data for the news translation shared task.

5.2 PROTEST: A Test Suite for Evaluating Pronouns in MT

Morphological agreement poses a particularly challenging problem for evaluation. Most automatic metrics assume that overlap of the MT output with a human-generated reference translation may be used as a proxy for correctness. When it comes to agreement, this assumption breaks down. For instance, if an anaphoric pronoun antecedent is translated in a way that differs from the reference translation, a different pronoun may be required: one that matches the reference translation may in fact be wrong. In the particular case of pronoun translation, Hardmeier [29] suggests using a test suite composed of carefully selected pronoun tokens which can be checked individually using an automatic evaluation script. In [24], UEDIN collaborated with Hardmeier to develop PROTEST, a test suite for the evaluation of pronoun translation by MT systems. The test suite comprises 250 hand-selected pronoun tokens and an automatic evaluation method which compares the translations of pronouns in MT output with those in the reference translation. Pronoun translations that do not match the reference are referred for manual evaluation. A major aim of this work is to encourage detailed analyses to highlight issues in the handling of specific linguistic mechanisms by MT systems, thereby contributing to a better understanding of those problems involved in translating pronouns.

This work is described in [24], which can be found in Annex A.19.
5.3 Assessing Morphological Agreement Using Contrastive Translation Pairs

Analysing translation quality with regards to specific linguistic phenomena has historically been difficult and time-consuming. Neural machine translation has the attractive property that it can produce scores for arbitrary translations, and UEDIN propose a novel method to assess how well NMT systems model specific linguistic phenomena including agreement over long distances and the morphological production of novel words. The core idea is that we can measure whether a reference translation is more probable under a NMT model than a contrastive translation which introduces a specific type of error. With this work, UEDIN introduce LingEval97, a large-scale data set of 97,000 contrastive translation pairs based on the WMT English→German translation task, with errors automatically created using simple rules. UEDIN report results for a number of systems, and find that recently introduced character-level NMT systems perform better at transliteration than models with byte-pair encoding (BPE) segmentation, but perform more poorly at morpho-syntactic agreement, and translating discontiguous units of meaning.

This work is described in [46], which can be found in Annex A.22. The work was developed jointly within WP1 and WP2 due to its applicability to both the problems of semantic and morphological phenomena. It is also reported in deliverable D1.2.
References


[38] Benjamin Marie, Alexandre Allauzen, Franck Burlot, Quoc-Khanh Do, Julia Ive, Elena Knyazeva, Matthieu Labeau, Thomas Lavergne, Kevin Löser, Nicolas Pécheux, and François Yvon. LIMSI@WMT’15 : Translation task. In *Proceedings of the Tenth Workshop on


A Annex

A.1 Linguistic Input Features Improve Neural Machine Translation

Linguistic Input Features Improve Neural Machine Translation

Rico Sennrich and Barry Haddow
School of Informatics, University of Edinburgh
rico.sennrich@ed.ac.uk, bhaddow@inf.ed.ac.uk

Abstract

Neural machine translation has recently achieved impressive results, while using little in the way of external linguistic information. In this paper we show that the strong learning capability of neural MT models does not make linguistic features redundant; they can be easily incorporated to provide further improvements in performance. We generalize the embedding layer of the encoder in the attentional encoder–decoder architecture to support the inclusion of arbitrary features, in addition to the baseline word feature. We add morphological features, part-of-speech tags, and syntactic dependency labels as input features to English→German and English→Romanian neural machine translation systems. In experiments on WMT16 training and test sets, we find that linguistic input features improve model quality according to three metrics: perplexity, BLEU and chrF3.

1 Introduction

Neural machine translation has recently achieved impressive results (Bahdanau et al., 2015; Jean et al., 2015), while learning from raw, sentence-aligned parallel text and using little in the way of external linguistic information. However, we hypothesize that various levels of linguistic annotation can be valuable for neural machine translation. Lemmatization can reduce data sparseness, and allow inflectional variants of the same word to explicitly share a representation in the model. Other types of annotation, such as parts-of-speech (POS) or syntactic dependency labels, can help in disambiguation. In this paper we investigate whether linguistic information is beneficial to neural translation models, or whether their strong learning capability makes explicit linguistic features redundant.

Let us motivate the use of linguistic features using examples of actual translation errors by neural MT systems. In translation out of English, one problem is that the same surface word form may be shared between several word types, due to homonymy or word formation processes such as conversion. For instance, close can be a verb, adjective, or noun, and these different meanings often have distinct translations into other languages. Consider the following English→German example:

1. We thought a win like this might be close.
2. Wir dachten, dass ein solcher Sieg nah sein könnte.

For the English source sentence in Example 1 (our translation in Example 2), a neural MT system (our baseline system from Section 4) mis-translates close as a verb, and produces the German verb schließen (Example 3), even though close is an adjective in this sentence, which has the German translation nah. Intuitively, part-of-speech annotation of the English input could disambiguate between verb, noun, and adjective meanings of close.

As a second example, consider the following German→English example:

4. Gefährlich ist die Route aber dennoch .

dangerous is the route but still.


83

Page 27 of 223
5. However the route is dangerous.
6. * Dangerous is the route, however.

German main clauses have a verb-second (V2) word order, whereas English word order is generally SVO. The German sentence (Example 4; English reference in Example 5) topicalizes the predicate gefährlich 'dangerous', putting the subject die Route 'the route' after the verb. Our baseline system (Example 6) retains the original word order, which is highly unusual in English, especially for prose in the news domain. A syntactic annotation of the source sentence could support the attentional encoder-decoder in learning which words in the German source to attend (and translate) first.

We will investigate the usefulness of linguistic features for the language pair German–English, considering the following linguistic features:

- lemmas
- subword tags (see Section 3.2)
- morphological features
- POS tags
- dependency labels

The inclusion of lemmas is motivated by the hope for a better generalization over inflectional variants of the same word form. The other linguistic features are motivated by disambiguation, as discussed in our introductory examples.

2 Neural Machine Translation

We follow the neural machine translation architecture by Bahdanau et al. (2015), which we will briefly summarize here.

The neural machine translation system is implemented as an attentional encoder-decoder network with recurrent neural networks.

The encoder is a bidirectional neural network with gated recurrent units (Cho et al., 2014) that reads an input sequence $x = (x_1, ..., x_m)$ and calculates a forward sequence of hidden states $(\hat{h}_1, ..., \hat{h}_m)$, and a backward sequence $(\hat{h}_1, ..., \hat{h}_m)$. The hidden states $\hat{h}_j$ and $\hat{h}_j$ are concatenated to obtain the annotation vector $h_j$.

The decoder is a recurrent neural network that predicts a target sequence $y = (y_1, ..., y_n)$. Each word $y_i$ is predicted based on a recurrent hidden state $s_i$, the previously predicted word $y_{i-1}$, and a context vector $c_i$. $c_i$ is computed as a weighted sum of the annotations $h_j$. The weight of each annotation $h_j$ is computed through an alignment model $\alpha_{ij}$, which models the probability that $y_i$ is aligned to $x_j$. The alignment model is a single-layer feedforward neural network that is learned jointly with the rest of the network through backpropagation.

A detailed description can be found in (Bahdanau et al., 2015), although our implementation is based on a slightly modified form of this architecture, released for the dl4mt tutorial². Training is performed on a parallel corpus with stochastic gradient descent. For translation, a beam search with small beam size is employed.

2.1 Adding Input Features

Our main innovation over the standard encoder-decoder architecture is that we represent the encoder input as a combination of features (Alexandrescu and Kirchhoff, 2006).

We here show the equation for the forward states of the encoder (for the simple RNN case; consider (Bahdanau et al., 2015) for GRU):

$$h_j = \tanh(W^x x_j + U^h h_{j-1})$$  \hspace{1cm} (1)

where $E \in \mathbb{R}^{m \times K_x}$ is a word embedding matrix, $W \in \mathbb{R}^{m \times m}$, $U \in \mathbb{R}^{m \times m}$ are weight matrices, with $m$ and $n$ being the vocabulary size and number of hidden units, respectively, and $K_x$ being the vocabulary size of the source language.

We generalize this to an arbitrary number of features $|F|:

$$\hat{h}_j = \tanh(W^x \large{\|} E_k x_j \large{\|} + U^h h_{j-1})$$  \hspace{1cm} (2)

where $\| \cdot \|$ is the vector concatenation, $E_k \in \mathbb{R}^{m_k \times K_x}$ are the feature embedding matrices, with $\sum_{k=1}^{|F|} m_k = m$, and $K_x$ is the vocabulary size of the $k$th feature. In other words, we look up separate embedding vectors for each feature, which are then concatenated. The length of the concatenated vector matches the total embedding size, and all other parts of the model remain unchanged.

²https://github.com/nyu-dl/dl4mt-tutorial
3 Linguistic Input Features

Our generalized model of the previous section supports an arbitrary number of input features. In this paper, we will focus on a number of well-known linguistic features. Our main empirical question is if providing linguistic features to the encoder improves the translation quality of neural machine translation systems, or if the information emerges from training encoder-decoder models on raw text, making its inclusion via explicit features redundant. All linguistic features are predicted automatically; we use Stanford CoreNLP (Toutanova et al., 2003; Minnen et al., 2001; Chen and Manning, 2014) to annotate the English input for English→German, and ParZu (Sennrich et al., 2013) to annotate the German input for German→English. We here discuss the individual features in more detail.

3.1 Lemma

Using lemmas as input features guarantees sharing of information between word forms that share the same base form. In principle, neural models can learn that inflectional variants are semantically related, and represent them as similar points in the continuous vector space (Mikolov et al., 2013). However, while this has been demonstrated for high-frequency words, we expect that a lemmatized representation increases data efficiency; low-frequency variants may even be unknown to word-level models. With character- or subword-level models, it is unclear to what extent they can learn the similarity between low-frequency word forms that share a lemma, especially if the word forms are superficially dissimilar. Consider the following two German word forms, which share the lemma *liegen* ‘lie’:

- *liegt* ‘lies’ (3.p.sg. present)
- *läge* ‘lay’ (3.p.sg. subjunctive II)

The lemmatisers we use are based on finite-state methods, which ensures a large coverage, even for infrequent word forms. We use the Zmorge analyzer for German (Schmid et al., 2004; Sennrich and Kunz, 2014), and the lemmatiser in the Stanford CoreNLP toolkit for English (Minnen et al., 2001).

3.2 Subword Tags

In our experiments, we operate on the level of subwords to achieve open-vocabulary translation with a fixed symbol vocabulary, using a segmentation based on byte-pair encoding (BPE) (Sennrich et al., 2016c). We note that in BPE segmentation, some symbols are potentially ambiguous, and can either be a separate word, or a subword segment of a larger word. Also, text is represented as a sequence of subword units with no explicit word boundaries, but word boundaries are potentially helpful to learn which symbols to attend to, and when to forget information in the recurrent layers. We propose an annotation of subword structure similar to popular IOB format for chunking and named entity recognition, marking if a symbol in the text forms the beginning (B), inside (I), or end (E) of a word. A separate tag (O) is used if a symbol corresponds to the full word.

3.3 Morphological Features

For German→English, the parser annotates the German input with morphological features. Different word types have different sets of features – for instance, nouns have case, number and gender, while verbs have person, number, tense and aspect – and features may be underspecified. We treat the concatenation of all morphological features of a word, using a special symbol for underspecified features, as a string, and treat each such string as a separate feature value.

3.4 POS Tags and Dependency Labels

In our introductory examples, we motivated POS tags and dependency labels as possible disambiguators. Each word is associated with one POS tag, and one dependency label. The latter is the label of the edge connecting a word to its syntactic head, or ‘ROOT’ if the word has no syntactic head.

3.5 On Using Word-level Features in a Subword Model

We segment rare words into subword units using BPE. The subword tags encode the segmentation of words into subword units, and need no further modification. All other features are originally word-level features. To annotate the segmented source text with features, we copy the word’s feature value to all its subword units. An example is shown in Figure 1.

4 Evaluation

We evaluate our systems on the WMT16 shared translation task English→German. The parallel
Leonidas begged in the arena.

words lemmas
Leonidas one oni das beg ged
B I E B E O O O

subword tags POS dep
B NNP NNP NNP IN DT NN

Figure 1: Original dependency tree for sentence Leonidas begged in the arena, and our feature representation after BPE segmentation.

training data consists of about 4.2 million sentence pairs.

To enable open-vocabulary translation, we encode words via joint BPE5 (Sennrich et al., 2016c), learning 89 500 merge operations on the concatenation of the source and target side of the parallel training data. We use minibatches of size 80, a maximum sentence length of 50, word embeddings of size 500, and hidden layers of size 1024. We clip the gradient norm to 1.0 (Pascanu et al., 2013). We train the models with Adadelta (Zeiler, 2012), reshuffling the training corpus between epochs. We validate the model every 10 000 minibatches via BLEU and perplexity on a validation set (newstest2013).

For neural MT, perplexity is a useful measure of how well the model can predict a reference translation given the source sentence. Perplexity is thus a good indicator of whether input features provide any benefit to the models, and we report the best validation set perplexity of each experiment. To evaluate whether the features also increase translation performance, we report case-sensitive BLEU scores with mteval-13b.perl on two test sets, newstest2015 and newstest2016. We also report CHRF3 (Popović, 2015), a character n-gram F3 score which was found to correlate well with human judgments, especially for translations out of English (Stanojević et al., 2015). The two metrics may occasionally disagree, partly because they are highly sensitive to the length of the output. BLEU is precision-based, whereas CHRF3 considers both precision and recall, with a bias for recall. For BLEU, we also report whether differences between systems are statistically significant according to a bootstrap resampling significance test (Riezler and Maxwell, 2005).

We train models for about a week, and report results for an ensemble of the 4 last saved models (with models saved every 12 hours). The ensemble serves to smooth the variance between single models.

Decoding is performed with beam search with a beam size of 12.

To ensure that performance improvements are not simply due to an increase in the number of model parameters, we keep the total size of the embedding layer fixed to 500. Table 1 lists the embedding size we use for linguistic features – the embedding layer size of the word-level feature varies, and is set to bring the total embedding size to 500. If we include the lemma feature, we roughly split the embedding vector one-to-two between the lemma feature and the word feature.

The table also shows the network vocabulary size; for all features except lemmas, we can represent all feature values in the network vocabulary – in the case of words, this is due to BPE segmentation. For lemmas, we choose the same vocabulary size as for words, replacing rare lemmas with a

1 https://github.com/rsennrich/subword-nmt
2 We use the re-implementation included with the subword code
special UNK symbol.

Sennrich et al. (2016b) report large gains from using monolingual in-domain training data, automatically back-translated into the source language to produce a synthetic parallel training corpus. We use the synthetic corpora produced in these experiments\(^7\) (3.6–4.2 million sentence pairs), and we trained systems which include this data to compare against the state of the art. We note that our experiments with this data entail a syntactic annotation of automatically translated data, which may be a source of noise. For the systems with synthetic data, we double the training time to two weeks.

We also evaluate linguistic features for the lower-resourced translation direction English→Romanian, with 0.6 million sentence pairs of parallel training data, and 2.2 million sentence pairs of synthetic parallel data. We use the same linguistic features as for English→German. We follow Sennrich et al. (2016a) in the configuration, and use dropout for the English→Romanian systems. We drop out full words (both on the source and target side) with a probability of 0.1. For all other layers, the dropout probability is set to 0.2.

### 4.1 Results

Table 2 shows our main results for German→English, and English→German. The baseline system is a neural MT system with only one input feature, the (sub)words themselves. For both translation directions, linguistic features improve the best perplexity on the development data (47.3 → 46.2, and 54.9 → 52.9, respectively). For German→English, the linguistic features lead to an increase of 1.5 BLEU (31.4→32.9) and 0.5 CHRF3 (58.0→58.5), on the newstest2016 test set. For English→German, we observe improvements of 0.6 BLEU (27.8→28.4) and 1.2 CHRF3 (56.0→57.2).

To evaluate the effectiveness of different linguistic features in isolation, we performed contrastive experiments in which only a single feature was added to the baseline. Results are shown in Table 3. Unsurprisingly, the combination of all features (Table 2) gives the highest improvement, averaged over metrics and test sets, but most features are beneficial on their own. Subword tags give small improvements for English→German, but not for German→English. All other features outperform the baseline in terms of perplexity, and yield significant improvements in BLEU on at least one test set. The gain from different features is not fully cumulative; we note that the information encoded in different features overlaps. For instance, both the dependency labels and the morphological features encode the distinction between German subjects and accusative objects, the former through different labels (ssubj and sobj), the latter through grammatical case (nominative and accusative).

We also evaluated adding linguistic features to a stronger baseline, which includes synthetic parallel training data. In addition, we compare our neural systems against phrase-based (PBSMT) and syntax-based (SBSMT) systems by (Williams et al., 2016), all of which make use of linguistic annotation on the source and/or target side. Results are shown in Table 4. For German→English, we observe similar improvements in the best development perplexity (45.2 → 44.1), test set BLEU (37.5→38.5) and CHRF3 (62.2→62.8). Our test set BLEU is on par to the best submitted system this year’s WMT 16 shared translation task, which is similar to our baseline MT system, but which also uses a right-to-left decoder for reranking (Sennrich et al., 2016a). We expect that linguistic input features and bidirectional decoding are orthogonal, and that we could obtain further improvements by combining the two.

For English→German, improvements in development set perplexity carry over (49.7 → 48.4), but we see only small, non-significant differences in BLEU and CHRF3. While we cannot clearly account for the discrepancy between perplexity and translation metrics, factors that potentially lower the usefulness of linguistic features in this setting are the stronger baseline, trained on more data, and the low robustness of linguistic tools in the annotation of the noisy, synthetic data sets. Both our baseline neural MT systems and the systems with linguistic features substantially outperform phrase-based and syntax-based systems for both translation directions.

In the previous tables, we have reported the best perplexity. To address the question about the randomness in perplexity, and whether the best perplexity just happened to be lower for the systems with linguistic features, we show perplexity on our development set as a function of training time.
Table 2: German↔English translation results: best perplexity on dev (newstest2013), and BLEU and CHR F3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different (p < 0.05) from respective baseline are marked with (*).

<table>
<thead>
<tr>
<th>System</th>
<th>English→German</th>
<th>German→English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>CHR F3</td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>test15</td>
</tr>
<tr>
<td>Baseline</td>
<td>47.3</td>
<td>27.9</td>
</tr>
<tr>
<td>All features</td>
<td>46.2</td>
<td>28.7*</td>
</tr>
</tbody>
</table>

Table 3: Contrastive experiments with individual linguistic features: best perplexity on dev (newstest2013), and BLEU and CHR F3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different (p < 0.05) from respective baseline are marked with (*).

<table>
<thead>
<tr>
<th>System</th>
<th>English→German</th>
<th>German→English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>CHR F3</td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>test15</td>
</tr>
<tr>
<td>Baseline</td>
<td>47.3</td>
<td>27.9</td>
</tr>
<tr>
<td>Lemmas</td>
<td>47.1</td>
<td>28.4</td>
</tr>
<tr>
<td>Subword tags</td>
<td>47.3</td>
<td>27.7</td>
</tr>
<tr>
<td>Morph. features</td>
<td>47.1</td>
<td>28.2</td>
</tr>
<tr>
<td>POS tags</td>
<td>46.9</td>
<td>28.1</td>
</tr>
<tr>
<td>Dependency labels</td>
<td>46.9</td>
<td>28.1</td>
</tr>
</tbody>
</table>

Table 4: German↔English translation results with additional, synthetic training data: best perplexity on dev (newstest2013), and BLEU and CHR F3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different (p < 0.05) from respective baseline are marked with (*).

<table>
<thead>
<tr>
<th>System</th>
<th>English→German</th>
<th>German→English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>CHR F3</td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>test15</td>
</tr>
<tr>
<td>PBSMT (Williams et al., 2016)</td>
<td>-</td>
<td>29.9</td>
</tr>
<tr>
<td>SBSMT (Williams et al., 2016)</td>
<td>-</td>
<td>29.5</td>
</tr>
<tr>
<td>Baseline</td>
<td>45.2</td>
<td>31.5</td>
</tr>
<tr>
<td>All features</td>
<td>44.1</td>
<td>32.1*</td>
</tr>
</tbody>
</table>
Figure 2: English→German (black) and German→English (red) development set perplexity as a function of training time (number of minibatches) with and without linguistic features.

Table 5: English→German (red) development set perplexity as a function of training time (number of minibatches) with and without linguistic features.

Table 6: Translation examples illustrating the effect of adding linguistic input features.

5 Related Work

Linguistic features have been used in neural language modelling (Alexandrescu and Kirchhoff, 2006), and are also used in other tasks for which neural models have recently been employed, such as syntactic parsing (Chen and Manning, 2014). This paper addresses the question whether linguistic features on the source side are beneficial for neural machine translation. On the target side, linguistic features are harder to obtain for a generation task such as machine translation, since this would require incremental parsing of the hypotheses at test time, and this is possible future work.

Among others, our model incorporates information from a dependency annotation, but is still a sequence-to-sequence model. Eriguchi et al. (2016) propose a tree-to-sequence model whose encoder computes vector representations for each phrase in the source tree. Their focus is on exploiting the (unlabelled) structure of a syntactic annotation, whereas we are focused on the disambiguation power of the functional dependency labels.

Factored translation models are often used in phrase-based SMT (Koehn and Hoang, 2007) as a means to incorporate extra linguistic information. However, neural MT can provide a much more flexible mechanism for adding such information. Because phrase-based models cannot easily generalize to new feature combinations, the individual models either treat each feature combination as an atomic unit, resulting in data sparsity, or assume independence between features, for instance by having separate language models for words and POS tags. In contrast, we exploit the strong generalization ability of neural networks, and expect that even new feature combinations, e.g. a word that appears in a novel syntactic function, are handled gracefully.

One could consider the lemmatized representation of the input as a second source text, and per-
6 Conclusion

In this paper we investigate whether linguistic input features are beneficial to neural machine translation, and our empirical evidence suggests that this is the case.

We describe a generalization of the encoder in the popular attentional encoder-decoder architecture for neural machine translation that allows for the inclusion of an arbitrary number of input features. We empirically test the inclusion of various linguistic features, including lemmas, part-of-speech tags, syntactic dependency labels, and morphological features, into English→German, and English→Romanian neural MT systems. Our experiments show that the linguistic features yield improvements over our baseline, resulting in improvements on newstest2016 of 1.5 BLEU for German→English, 0.6 BLEU for English→German, and 1.0 BLEU for English→Romanian.

In the future, we expect several developments that will shed more light on the usefulness of linguistic (or other) input features, and whether they will establish themselves as a core component of neural machine translation. On the one hand, the machine learning capability of neural architectures is likely to increase, decreasing the benefit provided by the features we tested. On the other hand, there is potential to explore the inclusion of novel features for neural MT, which might prove to be even more helpful than the ones we investigated, and the features we investigated may prove especially helpful for some translation settings, such as very low-resourced settings and/or translation settings with a highly inflected source language.

Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements 645452 (QT21), and 644402 (HimL).

References


Neural Machine Translation of Rare Words with Subword Units

Rico Sennrich and Barry Haddow and Alexandra Birch
School of Informatics, University of Edinburgh
{rico.sennrich,a.birch}@ed.ac.uk, bhaddow@inf.ed.ac.uk

Abstract

Neural machine translation (NMT) models typically operate with a fixed vocabulary, but translation is an open-vocabulary problem. Previous work addresses the translation of out-of-vocabulary words by backing off to a dictionary. In this paper, we introduce a simpler and more effective approach, making the NMT model capable of open-vocabulary translation by encoding rare and unknown words as sequences of subword units. This is based on the intuition that various word classes are translatable via smaller units than words, for instance names (via character copying or transliteration), compounds (via compositional translation), and cognates and loanwords (via phonological and morphological transformations). We discuss the suitability of different word segmentation techniques, including simple character n-gram models and a segmentation based on the byte pair encoding compression algorithm, and empirically show that subword models improve over a back-off dictionary baseline for the WMT 15 translation tasks English—German and English—Russian by up to 1.1 and 1.3 BLEU, respectively.

1 Introduction

Neural machine translation has recently shown impressive results (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). However, the translation of rare words is an open problem. The vocabulary of neural models is typically limited to 30,000–50,000 words, but translation is an open-vocabulary problem, and especially for languages with productive word formation processes such as agglutination and compounding, translation models require mechanisms that go below the word level. As an example, consider compounds such as the German Abwasser|behandlungs|anlage ‘sewage water treatment plant’, for which a segmented, variable-length representation is intuitively more appealing than encoding the word as a fixed-length vector.

For word-level NMT models, the translation of out-of-vocabulary words has been addressed through a back-off to a dictionary look-up (Jean et al., 2015; Luong et al., 2015b). We note that such techniques make assumptions that often do not hold true in practice. For instance, there is not always a 1-to-1 correspondence between source and target words because of variance in the degree of morphological synthesis between languages, like in our introductory compounding example. Also, word-level models are unable to translate or generate unseen words. Copying unknown words into the target text, as done by (Jean et al., 2015; Luong et al., 2015b), is a reasonable strategy for names, but morphological changes and transliteration is often required, especially if alphabets differ.

We investigate NMT models that operate on the level of subword units. Our main goal is to model open-vocabulary translation in the NMT network itself, without requiring a back-off model for rare words. In addition to making the translation process simpler, we also find that the subword models achieve better accuracy for the translation of rare words than large-vocabulary models and back-off dictionaries, and are able to productively generate new words that were not seen at training time. Our analysis shows that the neural networks are able to learn compounding and transliteration from subword representations.

This paper has two main contributions:

• We show that open-vocabulary neural ma...
chine translation is possible by encoding (rare) words via subword units. We find our architecture simpler and more effective than using large vocabularies and back-off dictionaries (Jean et al., 2015; Luong et al., 2015b).

- We adapt byte pair encoding (BPE) (Gage, 1994), a compression algorithm, to the task of word segmentation. BPE allows for the representation of an open vocabulary through a fixed-size vocabulary of variable-length character sequences, making it a very suitable word segmentation strategy for neural network models.

2 Neural Machine Translation

We follow the neural machine translation architecture by Bahdanau et al. (2015), which we will briefly summarize here. However, we note that our approach is not specific to this architecture.

The neural machine translation system is implemented as an encoder-decoder network with recurrent neural networks.

The encoder is a bidirectional neural network with gated recurrent units (Cho et al., 2014) that reads an input sequence $x = (x_1, ..., x_m)$ and calculates a forward sequence of hidden states $(\bar{h}_1, ..., \bar{h}_m)$, and a backward sequence $(\bar{h}_1, ..., \bar{h}_m)$. The hidden states $\bar{h}_j$ and $\bar{h}_j$ are concatenated to obtain the annotation vector $h_j$.

The decoder is a recurrent neural network that predicts a target sequence $y = (y_1, ..., y_n)$. Each word $y_i$ is predicted based on a recurrent hidden state $h_i$, the previously predicted word $y_{i-1}$, and a context vector $c_i$. $c_i$ is computed as a weighted sum of the annotations $h_j$. The weight of each annotation $h_j$ is computed through an alignment model $\alpha_{ij}$, which models the probability that $y_i$ is aligned to $x_j$. The alignment model is a single-layer feedforward neural network that is learned jointly with the rest of the network through back-propagation.

A detailed description can be found in (Bahdanau et al., 2015). Training is performed on a parallel corpus with stochastic gradient descent. For translation, a beam search with small beam size is employed.

3 Subword Translation

The main motivation behind this paper is that the translation of some words is transparent in that they are translatable by a competent translator even if they are novel to him or her, based on a translation of known subword units such as morphemes or phonemes. Word categories whose translation is potentially transparent include:

- named entities. Between languages that share an alphabet, names can often be copied from source to target text. Transcription or transliteration may be required, especially if the alphabets or syllabaries differ. Example: Barack Obama (English; German)
- morphologically complex words. Words containing multiple morphemes, for instance formed via compounding, affixation, or inflection, may be translatable by translating the morphemes separately. Example: solar system (English)

In an analysis of 100 rare tokens (not among the 50,000 most frequent types) in our German training data\(^2\), the majority of tokens are potentially translatable from English through smaller units. We find 56 compounds, 21 names, 6 loanwords with a common origin (emancipate — emancipieren), 5 cases of transparent affixation (sweetish ‘sweet’ + ‘-ish’ → süßlich ‘süß’ + ‘-lich’), 1 number and 1 computer language identifier.

Our hypothesis is that a segmentation of rare words into appropriate subword units is sufficient to allow for the neural translation network to learn transparent translations, and to generalize this knowledge to translate and produce unseen words.\(^3\) We provide empirical support for this hy-
pothesis in Sections 4 and 5. First, we discuss different subword representations.

3.1 Related Work

For Statistical Machine Translation (SMT), the translation of unknown words has been the subject of intensive research. A large proportion of unknown words are names, which can just be copied into the target text if both languages share an alphabet. If alphabets differ, transliteration is required (Durrani et al., 2014). Character-based translation has also been investigated with phrase-based models, which provided especially successful for closely related languages (Vilár et al., 2007; Tiedemann, 2009; Neubig et al., 2012).

The segmentation of morphologically complex words such as compounds is widely used for SMT, and various algorithms for morpheme segmentation have been investigated (Nießen and Ney, 2000; Koehn and Knight, 2003; Virpioja et al., 2007; Stallard et al., 2012). Segmentation algorithms commonly used for phrase-based SMT tend to be conservative in their splitting decisions, whereas we aim for an aggressive segmentation that allows for open-vocabulary translation with a compact network vocabulary, and without having to resort to back-off dictionaries.

The best choice of subword units may be task-specific. For speech recognition, phone-level language models have been used (Bazzi and Glass, 2000). Mikolov et al. (2012) investigate subword language models, and propose to use syllables. For multilingual segmentation tasks, multilingual algorithms have been proposed (Snyder and Barzilay, 2008). We find these intriguing, but inapplicable at test time.

Various techniques have been proposed to produce fixed-length continuous word vectors based on characters or morphemes (Luong et al., 2013; Botha and Blunsom, 2014; Ling et al., 2015a; Kim et al., 2015). An effort to apply such techniques to NMT, parallel to ours, has found no significant improvement over word-based approaches (Ling et al., 2015b). One technical difference from our work is that the attention mechanism still operates on the level of words in the model by Ling et al. (2015b), and that the representation of each word is fixed-length. We expect that the attention mechanism benefits from our variable-length representation: the network can learn to place attention on different subword units at each step. Recall our introductory example Abwasserbehandlungsanlage, for which a subword segmentation avoids the information bottleneck of a fixed-length representation.

Neural machine translation differs from phrase-based methods in that there are strong incentives to minimize the vocabulary size of neural models to increase time and space efficiency, and to allow for translation without back-off models. At the same time, we also want a compact representation of the text itself, since an increase in text length reduces efficiency and increases the distances over which neural models need to pass information.

A simple method to manipulate the trade-off between vocabulary size and text size is to use shortlists of unsegmented words, using subword units only for rare words. As an alternative, we propose a segmentation algorithm based on byte pair encoding (BPE), which lets us learn a vocabulary that provides a good compression rate of the text.

3.2 Byte Pair Encoding (BPE)

Byte Pair Encoding (BPE) (Gage, 1994) is a simple data compression technique that iteratively replaces the most frequent pair of bytes in a sequence with a single, unused byte. We adapt this algorithm for word segmentation. Instead of merging frequent pairs of bytes, we merge characters or character sequences.

Firstly, we initialize the symbol vocabulary with the character vocabulary, and represent each word as a sequence of characters, plus a special end-of-word symbol ‘~’, which allows us to restore the original tokenization after translation. We iteratively count all symbol pairs and replace each occurrence of the most frequent pair (‘A’, ‘B’) with a new symbol ‘AB’. Each merge operation produces a new symbol which represents a character n-gram. Frequent character n-grams (or whole words) are eventually merged into a single symbol, thus BPE requires no shortlist. The final symbol vocabulary size is equal to the size of the initial vocabulary, plus the number of merge operations -- the latter is the only hyperparameter of the algorithm.

For efficiency, we do not consider pairs that cross word boundaries. The algorithm can thus be run on the dictionary extracted from a text, with each word being weighted by its frequency. A minimal Python implementation is shown in Al-
Algorithm 1 Learn BPE operations

```python
import re, collections
def get_stats(vocab):
    pairs = collections.defaultdict(int)
    for word, freq in vocab.items():
        symbols = word.split()
        pairs[(symbols[i], symbols[i+1])] += freq
    return pairs

def merge_vocab(pair, v_in):
    v_out = {}
    p = re.compile('\b{}\b'.format(re.escape(''.join(pair))))
    for word in v_in:
        w_out = p.sub(''.join(pair), word)
        v_out[w_out] = v_in[word]
    return v_out

vocab = {'l o w </w>': 5, 'l o w e r </w>': 2, 'low er': 1}
best = max(vocab.items(), key=vocab.get)
vocab = merge_vocab(best, vocab)

Algorithm 1
```

Figure 1: BPE merge operations learned from dictionary {‘low’, ‘lowest’, ‘newer’, ‘wider’}.

The main difference to other compression algorithms, such as Huffman encoding, which have been proposed to produce a variable-length encoding of words for NMT (Chininis and Denero, 2015), is that our symbol sequences are still interpretable as subword units, and that the network can generalize to translate and produce new words (unseen at training time) on the basis of these subword units. Figure 1 shows a toy example of learned BPE operations. At test time, we first split words into sequences of characters, then apply the learned operations to merge the characters into larger, known symbols. This is applicable to any word, and allows for open-vocabulary networks with fixed symbol vocabularies. In our example, the OOV ‘lower’ would be segmented into ‘low er’.

We evaluate two methods of applying BPE: learning two independent encodings, one for the source, one for the target vocabulary, or learning the encoding on the union of the two vocabularies (which we call joint BPE). The former has the advantage of being more compact in terms of text and vocabulary size, and having stronger guarantees that each subword unit has been seen in the training text of the respective language, whereas the latter improves consistency between the source and the target segmentation. If we apply BPE independently, the same name may be segmented differently in the two languages, which makes it harder for the neural models to learn a mapping between the subword units. To increase the consistency between English and Russian segmentation despite the differing alphabets, we transliterate the Russian vocabulary into Latin characters with ISO-9 to learn the joint BPE encoding, then transliterate the BPE merge operations back into Cyrillic to apply them to the Russian training text.3

### 4 Evaluation

We aim to answer the following empirical questions:

- Can we improve the translation of rare and unseen words in neural machine translation by representing them via subword units?
- Which segmentation into subword units performs best in terms of vocabulary size, text size, and translation quality?

We perform experiments on data from the shared translation task of WMT 2015. For English—German, our training set consists of 4.2 million sentence pairs, or approximately 100 million tokens. For English—Russian, the training set consists of 2.6 million sentence pairs, or approximately 50 million tokens. We tokenize and truecase the data with the scripts provided in Moses (Koehn et al., 2007). We use newstest2013 as development set, and report results on newstest2014 and newstest2015.

We report results with BLEU (mteval-v13a.pl), and CHRF3 (Popović, 2015), a character n-gram F3 score which was found to correlate well with

---

2The only symbols that will be unknown at test time are unknown characters, or symbols of which all occurrences in the training text have been merged into larger symbols, like ‘safeguard’, which has all occurrences in our training text merged into ‘safeguard’. We observed no such symbols at test time, but the issue could be easily solved by recursively reversing specific merges until all symbols are known.

3In practice, we simply concatenate the source and target side of the training set to learn joint BPE.

4Since the Russian training text also contains words that use the Latin alphabet, we also apply the Latin BPE operations.
human judgments, especially for translations out
of English (Stanojević et al., 2015). Since our
main claim is concerned with the translation of
rare and unseen words, we report separate sta-
tistics for these. We measure these through unigram
F1, which we calculate as the harmonic mean of
clipped unigram precision and recall. 6

We perform all experiments with Groundhog7
(Bahdanau et al., 2015). We generally follow set-
tings by previous work (Bahdanau et al., 2015;
Jean et al., 2015). All networks have a hidden
layer size of 1000, and an embedding layer size
of 620. Following Jean et al. (2015), we only keep
a shortlist of \( \tau = 30000 \) words in memory.

During training, we use Adadelta (Zeiler, 2012),
a minibatch size of 80, and reshuffle the train-
ing set between epochs. We train a network for
approximately 7 days, then take the last 4 saved
models (models being saved every 12 hours), and
continue training each with a fixed embedding
layer (as suggested by (Jean et al., 2015)) for 12
hours. We perform two independent training runs
for each models, once with cut-off for gradient
clipping (Pascanu et al., 2013) of 5.0, once with
a cut-off of 1.0 – the latter produced better single
models for most settings. We report results of the
system that performed best on our development set
(newstest2013), and of an ensemble of all 8 mod-
els.

We use a beam size of 12 for beam search,
with probabilities normalized by sentence length.
We use a bilingual dictionary based on fast-align
(Dyer et al., 2013). For our baseline, this serves
as back-off dictionary for rare words. We also use
the dictionary to speed up translation for all ex-
periments, only performing the softmax over a fil-
tered list of candidate translations (like Jean et al.
(2015), we use \( K = 30000 \); \( K^2 = 10 \)).

4.1 Subword statistics

Apart from translation quality, which we will ver-
ify empirically, our main objective is to represent
an open vocabulary through a compact fixed-size
subword vocabulary, and allow for efficient train-
ing and decoding. 8

Statistics for different segmentations of the Ger-
man side of the parallel data are shown in Table
1. A simple baseline is the segmentation of words
into character \( n \)-grams. 9 Character \( n \)-grams allow
for different trade-offs between sequence length
(\# tokens) and vocabulary size (\# types), depend-
ing on the choice of \( n \). The increase in sequence
length is substantial; one way to reduce sequence
length is to leave a shortlist of the \( k \) most frequent
word types unsegmented. Only the unigram repre-
sentation is truly open-vocabulary. However, the
unigram representation performed poorly in pre-
liminary experiments, and we report translation re-
sults with a bigram representation, which is empir-
ically better, but unable to produce some tokens in
the test set with the training set vocabulary.

We report statistics for several word segmen-
tation techniques that have proven useful in previous
SMT research, including frequency-based comp-
ound splitting (Koehn and Knight, 2003), rule-
based hyphenation (Liang, 1983), and Morfessor
(Creutz and Lagus, 2002). We find that they only
moderately reduce vocabulary size, and do not
solve the unknown word problem, and we thus find
them unsuitable for our goal of open-vocabulary
translation without back-off dictionary.

BPE meets our goal of being open-vocabulary,
and the learned merge operations can be applied
to the test set to obtain a segmentation with no
unknown symbols. 10 Its main difference from
the character-level model is that the more com-
 pact representation of BPE allows for shorter se-
quencies, and that the attention model operates
on variable-length units. 11 Table 1 shows BPE
with 59 500 merge operations, and joint BPE with
89 500 operations.

In practice, we did not include infrequent sub-
word units in the NMT network vocabulary, since
there is noise in the subword symbol sets, e.g.
because of characters from foreign alphabets.
Hence, our network vocabularies in Table 2 are
typically slightly smaller than the number of types
in Table 1.

---

6Clipped unigram precision is essentially 1-gram BLEU
without brevity penalty.

7github.com/sebastien-J/LV_groundhog

8The time complexity of encoder-decoder architectures is
at least linear to sequence length, and oversplitting harms ef-


ciency.

9Our character \( n \)-grams do not cross word boundaries. We
mark whether a subword is word-final or not with a special
character, which allows us to restore the original tokenization.

10Joint BPE can produce segments that are unknown be-
cause they only occur in the English training text, but these
are rare (0.05% of test tokens).

11We highlighted the limitations of word-level attention in
section 3.1. At the other end of the spectrum, the character
level is suboptimal for alignment (Tiedemann, 2009).
Table 1: Corpus statistics for German training
corpus with different word segmentation tech-
niques. \( \# \text{UNK} \): number of unknown tokens in the
corpus with different word segmentation tech-
niques. \( \# \text{tokens} \): number of tokens; \( \# \text{types} \): number of types.

<table>
<thead>
<tr>
<th>segmentation</th>
<th># tokens</th>
<th># types</th>
<th># UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>100 m</td>
<td>1 750 000</td>
<td>1 079</td>
</tr>
<tr>
<td>characters</td>
<td>550 m</td>
<td>3 000</td>
<td>0</td>
</tr>
<tr>
<td>character bigrams</td>
<td>306 m</td>
<td>20 000</td>
<td>34</td>
</tr>
<tr>
<td>character trigrams</td>
<td>214 m</td>
<td>120 000</td>
<td>59</td>
</tr>
<tr>
<td>compound splitting</td>
<td>102 m</td>
<td>110 000</td>
<td>643</td>
</tr>
<tr>
<td>morfessor*</td>
<td>109 m</td>
<td>544 000</td>
<td>237</td>
</tr>
<tr>
<td>hyphenation*</td>
<td>186 m</td>
<td>404 000</td>
<td>230</td>
</tr>
<tr>
<td>BPE</td>
<td>112 m</td>
<td>63 000</td>
<td>122</td>
</tr>
<tr>
<td>BPE (joint)</td>
<td>111 m</td>
<td>82 000</td>
<td>32</td>
</tr>
<tr>
<td>character bigrams</td>
<td>129 m</td>
<td>69 000</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 2: English—German translation performance (BLEU, CHRF3 and unigram F1) on newstest2015.

<table>
<thead>
<tr>
<th>segmentation</th>
<th># tokens</th>
<th># types</th>
<th># UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDict</td>
<td>300 000</td>
<td>3 000</td>
<td>0</td>
</tr>
<tr>
<td>C2-50k</td>
<td>50 000</td>
<td>120 000</td>
<td>59</td>
</tr>
<tr>
<td>BPE-60k &amp; BPE</td>
<td>60 000</td>
<td>60 000</td>
<td>59</td>
</tr>
<tr>
<td>BPE-J90k &amp; BPE (joint)</td>
<td>90 000</td>
<td>90 000</td>
<td>59</td>
</tr>
</tbody>
</table>

Unigram F1 scores indicate that learning the BPE symbols on the vocabulary union (BPE-J90k) is more effective than learning them separately (BPE-60k), and more effective than using character bigrams with a shortlist of 50 000 unsegmented words (C2-50k), but all reported subword segmentations are viable choices and outperform the back-off dictionary baseline.

Our subword representations cause big improvements in the translation of rare and unseen words, but these only constitute 9-11% of the test sets. Since rare words tend to carry central information in a sentence, we suspect that BLEU and CHRF3 underestimate their effect on translation quality. Still, we also see improvements over the baseline in total unigram F1, as well as BLEU and CHRF3, and the subword ensembles outperform the WDict baseline by 0.3–1.3 BLEU and 0.6–2 CHRF3. There is some inconsistency between BLEU and CHRF3, which we attribute to the fact that BLEU has a precision bias, and CHRF3 a recall bias.

For English—German, we observe the best BLEU score of 25.3 with C2-50k, but the best CHRF3 score of 54.1 with BPE-J90k. For comparison to the (to our knowledge) best non-neural MT system on this data set, we report syntax-based SMT results (Sennrich and Haddow, 2015). We observe that our best systems outperform the syntax-based system in terms of BLEU, but not in terms of CHRF3. Regarding other neural systems, Luong et al. (2015a) report a BLEU score of 25.9 on newstest2015, but we note that they use an ensemble of 8 independently trained models, and also report strong improvements from applying dropout, which we did not use. We are confident that our improvements to the translation of rare words are orthogonal to improvements achievable through other improvements in the network archi-
tecture, training algorithm, or better ensembles.
For English → Russian, the state of the art is
the phrase-based system by Haddow et al. (2015).
It outperforms our WDict baseline by 1.5 BLEU.
The subword models are a step towards closing
this gap, and BPE-J90k yields an improvement of
1.3 BLEU, and 2.0 CHRF3, over WDict.

As a further comment on our translation results,
we want to emphasize that performance variabil-
ity is still an open problem with NMT. On our de-
velopment set, we observe differences of up to 1
BLEU between different models. For single sys-
tems, we report the results of the model that per-
forms best on dev (out of 8), which has a stabil-
izing effect, but how to control for randomness
deserves further attention in future research.

5 Analysis

5.1 Unigram accuracy

Our main claims are that the translation of rare and
unknown words is poor in word-level NMT mod-
els, and that subword models improve the trans-
lation of these word types. To further illustrate
the effect of different subword segmentations on
the translation of rare and unseen words, we plot
target-side words sorted by their frequency in the
training set.\footnote{We perform binning of words with the same training set
frequency, and apply bezier smoothing to the graph.}

Figure 2 shows results for the English–German
ensemble systems on newstest2015. Unigram
F1 of all systems tends to decrease for lower-
frequency words. The baseline system has a spike
in F1 for OOVs, i.e. words that do not occur in
the training text. This is because a high propor-
tion of OOVs are names, for which a copy from
the source to the target text is a good strategy for
English–German.

The systems with a target vocabulary of 500,000
words mostly differ in how well they translate
words with rank > 500,000. A back-off dictionary
is an obvious improvement over producing UNK,
but the subword system C2-3/500k achieves better
performance. Note that all OOVs that the back-
off dictionary produces are words that are copied
from the source, usually names, while the subword
systems can productively form new words such as
compounds.

For the 50,000 most frequent words, the repre-
sentation is the same for all neural networks, and
all neural networks achieve comparable unigram
F1 for this category. For the interval between fre-
frequency rank 50,000 and 500,000, the comparison
between C2-3/500k and C2-50k unveils an inter-
esting difference. The two systems only differ in
the size of the shortlist, with C2-3/500k represent-
ing words in this interval as single units, and C2-
50k via subword units. We find that the perfor-
mance of C2-3/500k degrades heavily up to fre-
frequency rank 500,000, at which point the model
switches to a subword representation and perform-
ance recovers. The performance of C2-50k re-
mains more stable. We attribute this to the fact
that subword units are less sparse than words. In
our training set, the frequency rank 50,000 corre-
sponds to a frequency of 60 in the training data;
the frequency rank 500,000 to a frequency of 2.
Because subword representations are less sparse,
reducing the size of the network vocabulary, and
representing more words via subword units, can
lead to better performance.

The \( F_1 \) numbers hide some qualitative differ-
ences between systems. For English–German, WDict
produces few OOVs (26.5% recall), but with high precision (60.6%) , whereas the subword
systems achieve higher recall, but lower precision.
We note that the character bigram model C2-50k
produces the most OOV words, and achieves rela-
tively low precision of 29.1% for this category.
However, it outperforms the back-off dictionary
in recall (33.0%). BPE-60k, which suffers from
transliteration (or copy) errors due to segmenta-
tion inconsistencies, obtains a slightly better pre-
cision (32.4%), but a worse recall (26.6%). In con-
trast to BPE-60k, the joint BPE encoding of BPE-
J90k improves both precision (38.6%) and recall
(29.8%).

For English–Russian, unknown names can only rarely be copied, and usually require translit-
eration. Consequently, the WDict baseline per-
forms more poorly for OOVs (9.2% precision; 5.2% recall), and the subword models improve
both precision and recall (21.9% precision and
15.6% recall for BPE-J90k). The full unigram \( F_1 \)
plot is shown in Figure 3.
D2.3: Final Report: Morphologically Rich Languages

Table 3: English—Russian translation performance (BLEU, CHR F3 and unigram F1) on newstest2015. Ens-8: ensemble of 8 models. Best NMT system in bold. Unigram F1 (with ensembles) is computed for all words (n = 55654), rare words (not among top 50000 in training set; n = 5442), and OOVs (not in training set; n = 851).

<table>
<thead>
<tr>
<th>Name segmentation shortlist source</th>
<th>vocabulary BLEU</th>
<th>CHR F3</th>
<th>unigram F1 (%)</th>
<th>name segmentation shortlist target</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDict</td>
<td>24.3</td>
<td>53.8</td>
<td>56.0</td>
<td>31.3</td>
</tr>
<tr>
<td>C2-50k char-bigram 50000 60000 60000</td>
<td>20.9</td>
<td>54.0</td>
<td>56.0</td>
<td>57.2</td>
</tr>
<tr>
<td>BPE-60k BPE - 60000 60000</td>
<td>20.5</td>
<td>49.8</td>
<td>52.7</td>
<td>55.3</td>
</tr>
<tr>
<td>BPE-J90k BPE (joint) - 90000 100000</td>
<td>20.4</td>
<td>49.7</td>
<td>53.0</td>
<td>55.8</td>
</tr>
</tbody>
</table>

5.2 Manual Analysis

Table 4 shows two translation examples for the translation direction English—German, Table 5 for English—Russian. The baseline system fails for all of the examples, either by deleting content (health), or by copying source words that should be translated or transliterated. The subword translations of health research institutes show that the subword systems are capable of learning translations when oversplitting (research → Forschung), or when the segmentation does not match morpheme boundaries: the segmentation Forschungs|instituten would be linguistically more plausible, and simpler to align to the English research institutes, than the segmentation Forsch|ungsinstitu|ten in the BPE-60k system, but still, a correct translation is produced. If the systems have failed to learn a translation due to data sparseness, like for asinine, which should be translated as dumm, we see translations that are wrong, but could be plausible for (partial) loanwords (asine|nine Situation → Asinin-Situation).

The English—Russian examples show that the subword systems are capable of transliteration. However, transliteration errors do occur, either due to ambiguous transliterations, or because of non-consistent segmentations between source and target text which make it hard for the system to learn a transliteration mapping. Note that the BPE-60k system encodes Mirzayeva inconsistently for the two language pairs (Mirz|ayeva—Ìèðçàåâà Mir|za|eva). This example is stil|l translated correctly, but we observe spurious insertions and deletions of characters in the BPE-60k system. An example is the transliteration of rakfisk, where a n is inserted and a x is deleted. We trace this error back to translation pairs in the training data with inconsistent segmentations, such as (pi|rakfisk → xp|frr|n)
system sentence
source health research institutes
reference Gesundheitsforschungsinstitute
WDict ... Translation
Empirical Methods in Natural Language Processing
(EMNLP).

Table 4: English—German translation example. “|” marks subword boundaries.

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>reference</td>
</tr>
<tr>
<td>WDdict</td>
<td>Mirzayeva</td>
</tr>
<tr>
<td>C2-50k</td>
<td>Mirzayeva → UNK → Mirzaeva</td>
</tr>
<tr>
<td>BPE-60k</td>
<td>Mirzayeva → Mhpjašča (Mirzayeva)</td>
</tr>
<tr>
<td>BPE-J90k</td>
<td>Mirzayeva → Mhpjašča (Mirzayeva)</td>
</tr>
</tbody>
</table>

Table 5: English—Russian translation examples. “|” marks subword boundaries.

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>reference</td>
</tr>
<tr>
<td>WDdict</td>
<td>rafisk</td>
</tr>
<tr>
<td>C2-50k</td>
<td>rakfisk → UNK → rakfiska</td>
</tr>
<tr>
<td>BPE-60k</td>
<td>rakfisk → ṁpačých (rafisk)</td>
</tr>
<tr>
<td>BPE-J90k</td>
<td>rakfisk → ṁpačých (rafisk)</td>
</tr>
</tbody>
</table>

6 Conclusion

The main contribution of this paper is that we show that neural machine translation systems are capable of open-vocabulary translation by representing rare and unseen words as a sequence of subword units. This is both simpler and more effective than using a back-off translation model. We introduce a variant of byte pair encoding for word segmentation, which is capable of encoding open vocabularies with a compact symbol vocabulary of variable-length subword units. We show performance gains over the baseline with both BPE segmentation, and a simple character bigram segmentation.

Our analysis shows that not only out-of-vocabulary words, but also rare in-vocabulary words are translated poorly by our baseline NMT system, and that reducing the vocabulary size of subword models can actually improve performance. In this work, our choice of vocabulary size is somewhat arbitrary, and mainly motivated by comparison to prior work. One avenue of future research is to learn the optimal vocabulary size for a translation task, which we expect to depend on the language pair and amount of training data, automatically. We also believe there is further potential in bilingually informed segmentation algorithms to create more alignable subword units, although the segmentation algorithm cannot rely on the target text at runtime.

While the relative effectiveness will depend on language-specific factors such as vocabulary size, we believe that subword segmentations are suitable for most language pairs, eliminating the need for large NMT vocabularies or back-off models.

Acknowledgments

We thank Maja Popović for her implementation of CHRf, with which we verified our re-implementation. The research presented in this publication was conducted in cooperation with Samsung Electronics Polska sp. z o.o. - Samsung R&D Institute Poland. This project received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement 645452 (QT21).

References


Page 45 of 223


A.3 Neural Language Models with Morphology

This section provides details on our experiments with neural language models enriched with morphological information. The experiments were part of a master thesis at CUNI, on which this report is based.

A.3.1 Representing Morphology

In a neural language model, the probability distribution is estimated by an artificial neural network. In this work we proposed a neural network architecture that estimates a language model based on lemmas and morphological tags. We used morphological annotation in the Prague Dependency Treebank format. There is a manual for morphological annotation by Zeman et al., also available online.

<table>
<thead>
<tr>
<th>position</th>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PoS</td>
<td>Part of speech</td>
</tr>
<tr>
<td>2</td>
<td>SubPoS</td>
<td>Detailed part of speech</td>
</tr>
<tr>
<td>3</td>
<td>Gender</td>
<td>Gender</td>
</tr>
<tr>
<td>4</td>
<td>Number</td>
<td>Number</td>
</tr>
<tr>
<td>5</td>
<td>Case</td>
<td>Case</td>
</tr>
<tr>
<td>6</td>
<td>PossGender</td>
<td>Possessor’s gender</td>
</tr>
<tr>
<td>7</td>
<td>PossNumber</td>
<td>Possessor’s number</td>
</tr>
<tr>
<td>8</td>
<td>Person</td>
<td>Person</td>
</tr>
<tr>
<td>9</td>
<td>Tense</td>
<td>Tense</td>
</tr>
<tr>
<td>10</td>
<td>Grade</td>
<td>Degree of comparison</td>
</tr>
<tr>
<td>11</td>
<td>Negation</td>
<td>Negation</td>
</tr>
<tr>
<td>12</td>
<td>Voice</td>
<td>Voice</td>
</tr>
<tr>
<td>13</td>
<td>Reserve1</td>
<td>Reserve</td>
</tr>
<tr>
<td>14</td>
<td>Reserve2</td>
<td>Reserve</td>
</tr>
<tr>
<td>15</td>
<td>Var</td>
<td>Variant, style</td>
</tr>
</tbody>
</table>

Table 3: Positions in Czech morphological tags.

The morphology is annotated in form of positional tags. The lemma and the tag together should uniquely identify the word form. Each positional tag is a string of 15 characters. Every position encodes one morphological category (see Table 3 with one character (mostly upper case letters or numbers).

To encode the tags for the neural network, we converted them to binary vectors. For each possible value of all morphological categories (in the order in which they appear in the documentation), we reserve one position in the binary vector. The position is filled with 1 if the tag has the corresponding value and with 0 otherwise. The resulting vector has 139 positions, where up to 13 values are 1, the rest is 0. In other words, it is concatenation of a series of one-hot vectors, one for each morphological category in the tag (or a zero vector of the corresponding dimension if the category is not used in that tag).

We also tried shorter representation, where we decided to ignore some of the categories (notably category “SubPoS”, which takes up almost as many positions as the rest of the tag combined) and ended up with 60-dimensional binary tags. However, in our experiments, the full 139 position tags worked better.

The architecture of our experimental model (Figure 1) is similar to the usual NLM architecture. The model predicts word forms (so it is directly comparable to the baseline) but the embeddings are trained on lemmas. We add the information from binary-encoded morphological tags to the embedding layer.

Figure 1: Schematic representation of the proposed LM architecture. The input layer is split into two parts. Each $l_k$ denotes one-hot vector representation of the $k$-th lemma, $e_k$ is the embedding of the lemma and $t_k$ is the corresponding morphological tag, encoded in a binary vector. For simplicity, the second hidden layer is omitted in this schema.
### Table 4: Datasets and their sizes.

<table>
<thead>
<tr>
<th>corpus</th>
<th>part</th>
<th>sentences</th>
<th>tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>c-news</td>
<td>train</td>
<td>197 k</td>
<td>4218 k</td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>2 k</td>
<td>47 k</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>2 k</td>
<td>41 k</td>
</tr>
<tr>
<td>c-fiction</td>
<td>train</td>
<td>4248 k</td>
<td>56432 k</td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>43 k</td>
<td>571 k</td>
</tr>
<tr>
<td></td>
<td>test</td>
<td>43 k</td>
<td>573 k</td>
</tr>
</tbody>
</table>

### Table 5: Perplexity for final models.

<table>
<thead>
<tr>
<th>model</th>
<th>dev</th>
<th>σ</th>
<th>test</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>MorphLM</td>
<td>161.93</td>
<td>1.28</td>
<td>272.55</td>
<td>8.45</td>
</tr>
<tr>
<td>FormLM</td>
<td>211.91</td>
<td>2.47</td>
<td>354.97</td>
<td>21.56</td>
</tr>
</tbody>
</table>

#### A.3.2 Evaluation

All our datasets are taken from the CzEng 1.0 corpus. Table 4 lists the sizes and sources (from which part of CzEng they were taken) of our datasets.

For a baseline, we trained a neural language model on word forms (we will refer to this model as FormLM). We will refer to our model with morphological information as MorphLM. We compared perplexity of the models and their performance in SMT n-best lists re-ranking.

Based on series of experiments, we selected parameters for the FormLM model to be compared against the MorphLM architecture:

- **word embedding size**: 300,
- **first hidden layer size**: 500,
- **second hidden layer size**: 500,
- **n-gram order**: 7,
- **softmax estimation**: sampled softmax,
- **sampling noise**: 500,
- **maximum number of epochs**: 15.

We trained 5 models and selected the one with the lowest perplexity on the development set (246.81). The perplexity of this model on the test dataset was 273.26.

By the same method, we selected parameters for the MorphLM model to be compared against the FormLM architecture:

- **lemma embedding size**: 300,
- **first hidden layer size**: 500,
- **second hidden layer size**: 500,
- **n-gram order**: 7,
- **softmax estimation**: sampled softmax,
- **sampling noise**: 500,
- **morphological tags encoding**: full (139 bits),
We trained 5 models and selected the one with the lowest perplexity on the development set (193.66). The perplexity of this model on the test dataset was 206.46, significantly lower than test perplexity for FormLM (273.26). The comparison of average perplexities for the models is in Table 5, their learning curves are plotted in Figure 2.

We also evaluated the models in re-ranking n-best list from the Moses SMT decoder. For training the SMT systems, we used the c-news dataset. For testing the systems, we used the c-news test set and also the WMT2013 test set with multiple reference translations (see Table 6 reproduced from Bojar et al. [4]). We will refer to this dataset as the bigref dataset.

We produced a 500-best list for the development set using Moses with a 5-gram KenLM. The hypotheses were lemmatized and tagged with MorphoDiTa [5]. We added the score from our models as a new feature to the 500-best list and trained the reranking weights with k-best MIRA [10].

Because the last stage of Moses training and the training of the reranking weights are both random processes, we repeated each of them five times and processed the results with MultEval [11]. In the following tables, the baseline is just the translation of the test set with the same model that produced the 500-best list for the development set, FormLM and MorphLM are 100-best lists of the test set translations reranked by our models.

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Czech</th>
<th># Refs</th>
<th># Snts</th>
</tr>
</thead>
<tbody>
<tr>
<td>newstest2011</td>
<td>official + 3 more from German</td>
<td>4</td>
<td>3003</td>
<td></td>
</tr>
<tr>
<td>newstest2011</td>
<td>2 post-edits of a system</td>
<td>2</td>
<td>1997</td>
<td></td>
</tr>
<tr>
<td>newstest2009</td>
<td>official</td>
<td>1</td>
<td>2525</td>
<td></td>
</tr>
<tr>
<td>newstest2008</td>
<td>official</td>
<td>1</td>
<td>2051</td>
<td></td>
</tr>
<tr>
<td>newstest2007</td>
<td>official</td>
<td>1</td>
<td>2007</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>4</strong></td>
<td><strong>11583</strong></td>
</tr>
</tbody>
</table>

Table 6: The contents of the bigref dataset.

**maximum number of epochs**: 15.
Table 7: Reranking with final FormLM and MorphLM models. Measured on the c-news test set. FormLM100 and MorphLM100 are models with 100,000 words vocabulary. Metric scores for all systems: jBLEU V0.1.1 (an exact reimplementation of NIST’s mteval-v13.pl without tokenization); Meteor V1.4 cz on rank task with all default modules NOT ignoring punctuation; Translation Error Rate (TER) V0.8.0; Hypothesis length over reference length as a percent. P-values are relative to baseline and indicate whether a difference of this magnitude (between the baseline and the system on that line) is likely to be generated again by some random process (a randomized optimizer). Metric scores are averages over multiple runs. $s_{sel}$ indicates the variance due to test set selection and has nothing to do with optimizer instability.

We tested our final FormLM and MorphLM models on the c-news test set. We also tested two models trained with a larger (100,000) vocabulary.

A.3.3 Results

Comparing perplexity of the models, it is clear that MorphLM achieves lower perplexity than FormLM (see Table 5 for the results of the final experiment).

Our implementation also reports accuracy on the development set. The accuracy was growing during the training, achieving the maximum of 22% for FormLM and 24% for MorphLM. The top-10 accuracy (probability that the correct word was within the first ten highest-scoring possibilities) achieved the maximum of 46% for FormLM and 49% for MorphLM.

In the final setup, MorphLM training was on average around two times slower than FormLM training.

Reranking n-best lists improves translation for both FormLM and MorphLM, although not significantly. MorphLM did not perform better than FormLM in these tests. Models with larger vocabulary performed slightly better, but the difference is not significant.

The results are summarized in Table 7. We also tested our final models on the bigref dataset (Table 8) with multiple references. With multiple references, there is lower risk of the situation where a system actually achieves better translation, but it is not evaluated as such, because it
Quality Translation 21
D2.3: Final Report: Morphologically Rich Languages

<table>
<thead>
<tr>
<th>Metric</th>
<th>System</th>
<th>Avg</th>
<th>s_{sel}</th>
<th>s_{Test}</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU ↑</td>
<td>baseline</td>
<td>29.0</td>
<td>0.4</td>
<td>0.1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>FormLM</td>
<td>29.2</td>
<td>0.4</td>
<td>0.1</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>MorphLM</td>
<td>29.2</td>
<td>0.4</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>METEOR ↑</td>
<td>baseline</td>
<td>25.7</td>
<td>0.2</td>
<td>0.0</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>FormLM</td>
<td>25.9</td>
<td>0.2</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>MorphLM</td>
<td>25.9</td>
<td>0.2</td>
<td>0.0</td>
<td>0.00</td>
</tr>
<tr>
<td>TER ↓</td>
<td>baseline</td>
<td>61.3</td>
<td>0.3</td>
<td>0.2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>FormLM</td>
<td>61.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>MorphLM</td>
<td>61.3</td>
<td>0.3</td>
<td>0.1</td>
<td>0.46</td>
</tr>
<tr>
<td>Length</td>
<td>baseline</td>
<td>105.2</td>
<td>0.1</td>
<td>0.2</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>FormLM</td>
<td>105.6</td>
<td>0.1</td>
<td>0.3</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>MorphLM</td>
<td>105.6</td>
<td>0.1</td>
<td>0.2</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 8: Reranking with final FormLM and MorphLM models. Measured on the bigref test set. For description of the metrics, see Table 7.

deviated from the reference too much.

We also tested our models on a single MERT run, that produced much better translations, than the rest. The results are in Table 9. The effect of the reranking was stronger here and MorphLM performed slightly better than FormLM. We think that reranking works better on better baseline, because the LMs are only trained to distinguish good sentences from noise, not bad sentences from worse sentences. However, we were not able to replicate this result and the difference between FormLM and MorphLM probably is not significant here.

A.3.4 Conclusion

We have examined various architectures of neural language models, both in theory and empirically. We proposed a new NLM architecture, that works with explicit morphological information, to take advantage of annotation tools and annotated datasets.

We were able to train language models with a significantly lower perplexity with the proposed architecture. However, it did not bring significant improvements to statistical machine translation with n-best list rescoring, compared to baseline NLM architecture.

Our results suggest that there is a potential to improve machine translation by including morphological information into language models. Recent developments in machine translation seem to favor neural machine translation with various subword approaches. An NMT system that combines a subword encoding with explicit morphological information might be a more promising way of including morphological annotation into machine translation.
<table>
<thead>
<tr>
<th>LMs</th>
<th>dev-BLEU</th>
<th>test-BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MorphLM</td>
<td>26.62</td>
<td>24.86</td>
</tr>
<tr>
<td>FormLM</td>
<td>26.72</td>
<td>24.76</td>
</tr>
<tr>
<td>baseline</td>
<td>26.24</td>
<td>24.14</td>
</tr>
</tbody>
</table>

Table 9: Reranking results for our models on the single best baseline MERT run.
SubGram: Extending Skip-Gram Word Representation with Substrings

Tom Kocmi(B) and Ondřej Bojar
Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Charles University in Prague, Prague, Czech Republic
{kocmi,bojar}@ufal.mff.cuni.cz

Abstract. Skip-gram (word2vec) is a recent method for creating vector representations of words (‘distributed word representations’) using a neural network. The representation gained popularity in various areas of natural language processing, because it seems to capture syntactic and semantic information about words without any explicit supervision in this respect.

We propose SubGram, a refinement of the Skip-gram model to consider also the word structure during the training process, achieving large gains on the Skip-gram original test set.

Keywords: Distributed word representations · Unsupervised learning of morphological relations

1 Introduction

Vector representations of words learned using neural networks (NN) have proven helpful in many algorithms for image annotation [1] or [2], language modeling [3–5] or other natural language processing (NLP) tasks [6] or [7].

Traditionally, every input word of an NN is stored in the “one-hot” representation, where the vector has only one element set to one and the rest of the vector are zeros. The size of the vector equals to the size of the vocabulary. The NN is trained to perform some prediction, e.g. to predict surrounding words given a word of interest. Instead of using this prediction capacity in some task, the practice is to extract the output of NN’s hidden layer of each word (called distributed representation) and directly use this deterministic mapping vec(·) of word forms to the vectors of real numbers as the word representation.

The input one-hot representation of words has two weaknesses: the bloat of the size of the vector with more words in vocabulary and the inability to provide any explicit semantic or syntactic information to the NN.

The learned distributed representation of words relies on much shorter vectors (e.g. vocabularies containing millions words are represented in vectors of a few hundred elements) and semantic or syntactic information is often found to be implicitly present (“embedded”) in the vector space. For example, the Euclidean distance between two words in the vector space may be related to semantic or syntactic similarity between them.
1.1 Skip-Gram Model

The authors of [8] created a model called Skip-gram, in which linear vector operations allow to find related words with surprisingly good results. For instance vec(king) − vec(man) + vec(woman) gives a value close to vec(queen).

In this paper, we extend Skip-gram model with the internal word structure and show how it improves the performance on embedding morpho-syntactic information.

The Skip-gram model defined in [8] is trained to predict context words of the input word. Given a corpus $T$ of words $w$ and their context words $c \in C(w)$ (i.e. individual words $c$ appearing close the original word $w$), it considers the conditional probabilities $p(c|w)$. The training finds the parameters $\theta$ of $p(c|w; \theta)$ to maximize the corpus probability:

$$\arg \max_{\theta} \prod_{w \in T} \prod_{c \in C(w)} p(c|w; \theta)$$

The Skip-gram model is a classic NN, where activation functions are removed and hierarchical soft-max [9] is used instead of soft-max normalization. The input representation is one-hot so the activation function is not needed on hidden layer, there is nothing to be summed up. This way, the model is learned much faster than comparable non-linear NNs and lends itself to linear vector operations possibly useful for finding semantically or syntactically related words.

2 Related Work

In [10] was proposed to append part-of-speech (POS) tags to each word and train Skip-gram model on the new vocabulary. This avoided conflating, e.g. nouns and verbs, leading to a better performance, at the cost of (1) the reliance on POS tags and their accurate estimation and (2) the increased sparsity of the data due to the larger vocabulary.

The authors in [11] used character-level input to train language models using a complex setup of NNs of several types. Their model was able to assign meaning to out-of-vocabulary words based on the closest neighbor. One disadvantage of the model is its need to run the computation on a GPU for a long time.

The authors of [12] proposed an extension of Skip-gram model which uses character similarity of words to improve performance on syntactic and semantic tasks. They are using a set of similar words as additional features for the NN. Various similarity measures are tested: Levenshtein, longest common substring, morpheme and syllable similarity.

The authors of [13] added the information about word’s root, affixes, syllables, synonyms, antonyms and POS tags to continuous bag-of-words model (CBOW) proposed by [8] and showed how these types of knowledge lead to better word embeddings. The CBOW model is a simpler model with usually worse performance than Skip-gram.
3 SubGram

We propose a substring-oriented extension of Skip-gram model which induces vector embeddings from character-level structure of individual words. This approach gives the NN more information about the examined word with no drawbacks in data sparsity or reliance on explicit linguistic annotation.

We append the characters ` and $ to the word to indicate its beginning and end. In order to generate the vector of substrings, we take all character bigrams, trigrams etc. up to the length of the word. This way, even the word itself is represented as one of the substrings. For the NN, each input word is then represented as a binary vector indicating which substrings appear in the word.

The original Skip-gram model [8] uses one-hot representation of a word in vocabulary as the input vector. This representation makes training fast because no summation or normalization is needed. The weights $w_i$ of the input word $i$ can be directly used as the output of hidden layer $h$ (and as the distributed word representation): $h_j = w_{ij}$.

In our approach, we provide the network with a binary vector representing all substrings of the word. To compute the input of hidden layer we decided to use mean value as it is computationally simpler than sigmoid:

$$h_j = \frac{\sum_{i=1}^{|X|} x_i \ast w_{ij}}{|S|}$$  \hspace{1cm} (2)

where $|S|$ is the number of substrings of the word $x$.

4 Evaluation and Data Sets

We train our NN on words and their contexts extracted from the English wikipedia dump from May 2015. We have cleaned the data by replacing all numbers with 0 and removing special characters except those usually present in the English text like dots, brackets, apostrophes etc. For the final training data we have randomly selected only 2.5M segments (mostly sentences). It consist of 96M running words with the vocabulary size of 1.09M distinct word forms.

We consider only the 141K most frequent word forms to simplify the training. The remaining word forms fall out of vocabulary (OOV), so the original Skip-gram cannot provide them with any vector representation. Our SubGram relies on known substrings and always provides at least some approximation.

We test our model on the original test set [8]. The test set consists of 19544 “questions”, of which 8869 are called “semantic” and 10675 are called “syntactic” and further divided into 14 types, see Table 1. Each question contains two pairs of words $(x_1, x_2, y_1, y_2)$ and captures relations like “What is to ‘woman’ ($y_1$) as ‘king’ ($x_2$) is to ‘man’ ($x_1$)?”, together with the expected answer ‘queen’ ($y_2$).

The model is evaluated by finding the word whose representation is the nearest (cosine similarity) to the vector $vec(king) − vec(man) + vec(woman)$. If the nearest neighbor is $vec(queen)$, we consider the question answered correctly.
Table 1. Mikolov’s test set question types, the upper part are “semantic” questions, the lower part are “syntactic”.

<table>
<thead>
<tr>
<th>Question type</th>
<th>Sample pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital-countries</td>
<td>Athens – Greece</td>
</tr>
<tr>
<td>Capital-world</td>
<td>Abuja – Nigeria</td>
</tr>
<tr>
<td>Currency</td>
<td>Algeria – dinar</td>
</tr>
<tr>
<td>City-in-state</td>
<td>Houston – Texas</td>
</tr>
<tr>
<td>Family</td>
<td>boy - girl</td>
</tr>
<tr>
<td>Adjective-to-adverb</td>
<td>calm – calmly</td>
</tr>
<tr>
<td>Opposite</td>
<td>aware – unaware</td>
</tr>
<tr>
<td>Comparative</td>
<td>bad – worse</td>
</tr>
<tr>
<td>Superlative</td>
<td>bad – worst</td>
</tr>
<tr>
<td>Present-participle</td>
<td>code – coding</td>
</tr>
<tr>
<td>Nationality-adjective</td>
<td>Albania – Albanian</td>
</tr>
<tr>
<td>Past-tense</td>
<td>dancing – danced</td>
</tr>
<tr>
<td>Plural</td>
<td>banana – bananas</td>
</tr>
<tr>
<td>Plural-verbs</td>
<td>decrease – decreases</td>
</tr>
</tbody>
</table>

In this work, we use Mikolov’s test set which is used in many papers. After a closer examination we came to the conclusion, that it does not test what the broad terms “syntactic” and “semantic relations” suggest. “Semantics” is covered by questions of only 3 types: predict a city based on a country or state, currency name from the country and the feminine variant of nouns denoting family relations. The authors of [14] showed, that many other semantic relationships could be tested, e.g. walk-run, dog-puppy, bark-dog, cook-eat and others.

“Syntactic” questions cover a wider range of relations at the boundary of morphology and syntax. The problem is that all questions of a given type are constructed from just a few dozens of word pairs, comparing pairs with each other. Overall, there are 313 distinct pairs throughout the whole syntactic test set of 10675 questions, which means only around 35 different pairs per question set. Moreover, of the 313 pairs, 286 pairs are regularly formed (e.g. by adding the suffix ‘ly’ to change an adjective into the corresponding adverb). Though it has to be mentioned that original model could not use this kind of information.

We find such a small test set unreliable to answer the question whether the embedding captures semantic and syntactic properties of words.

4.1 Rule-Based Baseline Approach

Although the original test set has been used to compare results in several papers, no-one tried to process it with some baseline approach. Therefore, we created a very simple set of rules for comparison on the syntactic part of the test set. The rules cover only the most frequent grammatical phenomena.
186 T. Kocmi and O. Bojar

- adjective-to-adverb: Add ly at the end of the adjective.
- opposite: Add un at the beginning of positive form.
- comparative: If the adjective ends with y, replace it with yer. If it ends with e, add r. Otherwise add er at the end.
- superlative: If the adjective ends with y, replace it with yest. If it ends with e, add st. Otherwise add est at the end.
- present-participle: If the verb ends with e, replace it with ing, otherwise add ing at the end.
- nationality-adjective: Add n at the end, e.g. Russia → Russian.
- past-tense: Remove ing and add ed at the end of the verb.
- plural: Add s at the end of the word.
- plural-verbs: If the word ends with a vowel, add es at the end, else add s.

4.2 Our Test Set

We have decided to create more general test set which would consider more than 35 pairs per question set. Since we are interested in morphosyntactic relations, we extended only the questions of the “syntactic” type with exception of nationality adjectives which is already covered completely in original test set.

We constructed the pairs more or less manually, taking inspiration in the Czech side of the CzEng corpus [15], where explicit morphological annotation allows to identify various pairs of Czech words (different grades of adjectives, words and their negations, etc.). The word-aligned English words often shared the same properties. Another sources of pairs were acquired from various web-pages usually written for learners of English. For example for verb tense, we relied on a freely available list of English verbs and their morphological variations. We have included 100–1000 different pairs per question set. The questions were constructed from the pairs similarly as by Mikolov: generating all possible pairs of pairs. This leads to millions of questions, so we randomly selected 1000 instances per question set, to keep the test set in the same order of magnitude. Additionally, we decided to extend set of questions on opposites to cover not only opposites of adjectives but also of nouns and verbs.

In order to test our extension of Skip-gram on out-of-vocabulary words, we created an additional subset of our test set with questions where at least one of \(x_1, x_2\) and \(y_1\) is not among the known word forms. Note that the last word \(y_2\) must be in vocabulary in order to check if the output vector is correct.

5 Experiments and Results

We used a Python implementation of word2vec\(^1\) as the basis for our SubGram, which we have made freely available\(^2\).

\(^1\) http://radimrehurek.com/gensim

Gensim implements the model twice, in Python and an optimized version in C. For our prototype, we opted to modify the Python version, which unfortunately resulted in a code about 100 times slower and forced us to train the model only on the 96M word corpus as opposed to Mikolov’s 100,000M word2vec training data used in training of the released model.

\(^2\) https://github.com/tomkocmi/SubGram.
SubGram: Extending Skip-Gram Word Representation with Substrings

We limit the vocabulary, requiring each word form to appear at least 10 times in the corpus and each substring to appear at least 500 times in the corpus. This way, we get the 141K unique words mentioned above and 170K unique substrings (+141K words, as we downsample words separately).

Our word vectors have the size of 100. The size of the context window is 5.

The accuracy is computed as the number of correctly answered questions divided by the total number of questions in the set. Because the Skip-gram cannot answer questions containing OOV words, we also provide results with such questions excluded from the test set (scores in brackets).

Tables 2 and 3 report the results. The first column shows the rule-based approach. The column “Released Skip-gram” shows results of the model released by Mikolov and was trained on a 100 billion word corpus from Google News and generates 300 dimensional vector representation. The third column shows Skip-gram model trained on our training data, the same data as used for the training of the SubGram. Last column shows the results obtained from our SubGram model.

Comparing Skip-gram and SubGram on the original test set (Table 2), we see that our SubGram outperforms Skip-gram in several morpho-syntactic question sets but over all performs similarly (42.5 % vs. 42.3 %). On the other hand, it does not capture the tested semantic relations at all, getting a zero score on average.

Table 2. Results on original test set questions. The values in brackets are based on questions without any OOVs.

<table>
<thead>
<tr>
<th></th>
<th>Rule based</th>
<th>Released skip-gram</th>
<th>Our skip-gram</th>
<th>SubGram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capital-countries</td>
<td>0%</td>
<td>18.6% (24.7%)</td>
<td>71.9% (71.9%)</td>
<td>0%</td>
</tr>
<tr>
<td>Capital-world</td>
<td>0%</td>
<td>2.2% (15.0%)</td>
<td>53.6% (54.6%)</td>
<td>0%</td>
</tr>
<tr>
<td>Currency</td>
<td>0%</td>
<td>7% (12.2%)</td>
<td>3% (4.7%)</td>
<td>0.1% (0.2%)</td>
</tr>
<tr>
<td>City-in-state</td>
<td>0%</td>
<td>9.2% (14%)</td>
<td>40.5% (40.5%)</td>
<td>0.1% (0.1%)</td>
</tr>
<tr>
<td>Family</td>
<td>0%</td>
<td>84.6% (84.6%)</td>
<td>82.6% (82.6%)</td>
<td>5.9% (5.9%)</td>
</tr>
<tr>
<td>Overall semantic</td>
<td>0%</td>
<td>10.2% (24.8%)</td>
<td>47.7% (50%)</td>
<td>0%</td>
</tr>
<tr>
<td>Adjective-to-adverb</td>
<td>90.6%</td>
<td>28.5% (28.5%)</td>
<td>16.3% (16.3%)</td>
<td>37.6% (73.7%)</td>
</tr>
<tr>
<td>Opposite</td>
<td>65.5%</td>
<td>42.7% (42.7%)</td>
<td>9.4% (10.1%)</td>
<td>43.1% (46.3%)</td>
</tr>
<tr>
<td>Comparative</td>
<td>89.2%</td>
<td>90.8% (90.8%)</td>
<td>72.1% (72.1%)</td>
<td>46.5% (46.5%)</td>
</tr>
<tr>
<td>Superlative</td>
<td>88.2%</td>
<td>87.3% (87.3%)</td>
<td>24.4% (25.9%)</td>
<td>45.9% (48.8%)</td>
</tr>
<tr>
<td>Present-participle</td>
<td>87.9%</td>
<td>78.1% (78.1%)</td>
<td>44.2% (44.2%)</td>
<td>43.5% (43.5%)</td>
</tr>
<tr>
<td>Nationality-adjective</td>
<td>31.7%</td>
<td>13.3% (21.9%)</td>
<td>60.4% (60.4%)</td>
<td>21.8% (21.8%)</td>
</tr>
<tr>
<td>Past-tense</td>
<td>42.5%</td>
<td>66% (66%)</td>
<td>35.6% (35.6%)</td>
<td>15.8% (15.8%)</td>
</tr>
<tr>
<td>Plural</td>
<td>86.5%</td>
<td>89.9% (89.9%)</td>
<td>46.8% (46.8%)</td>
<td>44.7% (44.7%)</td>
</tr>
<tr>
<td>Plural-verbs</td>
<td>93.3%</td>
<td>67.9% (67.9%)</td>
<td>51.5% (51.5%)</td>
<td>74.3% (74.3%)</td>
</tr>
<tr>
<td>Overall syntactic</td>
<td>71.9%</td>
<td>62.5% (66.5%)</td>
<td>42.5% (43%)</td>
<td>42.3% (42.7%)</td>
</tr>
</tbody>
</table>

1 https://code.google.com/archive/p/word2vec/.
When comparing models on our test set (Table 3), we see that given the same training set, SubGram significantly outperforms Skip-gram model (22.4% vs. 9.7%). The performance of Skip-gram trained on the much larger dataset is higher (43.5%) and it would be interesting to see the SubGram model, if we could get access to such training data. Note however, that the Rule-based baseline is significantly better on both test sets.

The last column suggests that the performance of our model on OOV words is not very high, but it is still an improvement over flat zero of the Skip-gram model. The performance on OOVs is expected to be lower, since the model has no knowledge of exceptions and can only benefit from regularities in substrings.

6 Future Work

We are working on a better test set for word embeddings which would include many more relations over a larger vocabulary especially semantics relations. We want to extend the test set with Czech and perhaps other languages, to see what word embeddings can bring to languages morphologically richer than English.

As shown in the results, the rule based approach outperforms NN approach on this type of task, therefore we would like to create a hybrid system which could use rules and part-of-speech tags. We will also include morphological tags in the model as proposed in [10] but without making the data sparse.

Finally, we plan to reimplement SubGram to scale up to larger training data.

7 Conclusion

We described SubGram, an extension of the Skip-gram model that considers also substrings of input words. The learned embeddings then better capture almost all morpho-syntactic relations tested on test set which we extended from original described in [8]. This test set is released for the public use\(^4\).

\(^4\) [https://ufal.mff.cuni.cz/tom-kocmi/syntactic-questions.]
An useful feature of our model is the ability to generate vector embeddings even for unseen words. This could be exploited by NNs also in different tasks.

Acknowledgment. This work has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 645452 (QT21), the grant GAUK 8502/2016, and SVV project number 260 333.

This work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

References

When we translate from a highly inflected language into a less morphologically rich language, not all syntactic information encoded in the surface forms may be needed to produce an accurate translation. For example, verbs in French must agree with the noun in case and gender. When we translate these verbs into English, both the case and gender information may be safely discarded. In order to address this sparsity problem, KIT tried to cluster words that have the same translation probability distribution, leading to higher occurrence counts and therefore more reliable translation statistics. After clustering the words into groups that can be translated in the same or at least in a similar way, there are different possibilities to use them in the translation system. A naive strategy is to replace each word by its cluster representative, called hard decision stemming. However, this carries the risk of discarding vital information. Therefore techniques to integrate both, the surface forms as well as the word stems, into the translation system were investigated. In the combined input, the stemmed adjectives were added as translation alternatives to the preordering lattices. Since this poses problems for the application of more advanced translation models during decoding, a novel hidden combination technique was proposed.

### A.5.1 Combined Input

Mistakes made during hard decision stemming cannot be recovered. Soft integration techniques avoid this pitfall by deferring the decision on whether to use the stem or surface form of a word until decoding. The system is able to choose by combining both the surface form based (default) phrase table and the word stem based (stemmed) phrase table log-linearly. The weights of the phrase scores are then learned during optimization. In order to be able to apply both phrase tables at the same time, the input of the decoder needs to be modified. Our baseline system already uses preordering lattices, which encode different reordering possibilities of the source sentence. Every edge in the lattice containing an adjective is replaced by two edges: one containing the surface form and the other the word stem. This allows the decoder to choose which word form to use depending on the word and its context.

### A.5.2 Hidden Combination

While it is possible to modify our phrase table to use both surface forms and stems in the last strategy, other models in our log-linear system suffer from the different types of source input. For example, the bilingual language model is based on tokens of target words and their aligned source words. In training, either the stemmed corpus or the original one can be used, but during decoding a mixture of stems and surface forms occurs. For the unknown word forms...
the scores will not be accurate and the performance of our model will suffer. Similar problems occur when using other translation models such as neural network based translation models.

Therefore a novel strategy to integrate the word stems into the translation system was developed. Instead of stemming the input to fit the stemmed phrase table, the stemmed phrase table is modified so that it can be applied to the surface forms. The workflow is illustrated in Figure 3. All the stem mappings are extracted from the development and test data and compiled a stem lexicon. This maps the surface forms observed in the dev and test data to their corresponding stems. Then this lexicon is applied in reverse to our stemmed phrase table, in effect duplicating every entry containing a stemmed adjective with the inflected form replacing the stem. Afterwards this “unstemmed” phrase table is log-linearly combined with the default phrase table and used for translation.

This allows us to retain our generalization won by using word clusters to estimate phrase probabilities, and still use all models trained on the surface forms. Using the hidden combination strategy, stemming can easily be implemented into current state-of-the-art SMT systems without the need to change any of the advanced models beyond the phrase table. This makes our approach highly versatile and easy to implement for any number of system architectures and languages.

### A.5.3 Evaluation

The problem of a high number of OOV-words is especially problematic, when translating from a morphologically rich language on a task with only limited training data. We tested this approach on the task of translating German TED lectures into English. The systems were only trained on the TED corpus.

The results for the systems built only on the TED corpus are summarized in Table 10 for the small system and Table 11 for the extended system. A bold font indicates results that are significantly better than the baseline system with $p < 0.05$. The baseline systems reach a BLEU score on the test set of 30.25 and 31.33 respectively.

The small system could be improve slightly to 30.30 using only stemmed adjectives. Adding the stemmed forms as alternatives to the preordering lattice leads to an improvement of 0.2 BLEU points over the small baseline system. This strategy does not tap the full potential of our extended system, as there is still a mismatch between the combined input and the training data of the advanced models.

The hidden combination strategy rectifies this problem, which is reflected in the results. Using the hidden combination the best BLEU score could be achieved for both systems. We could improve it by almost 0.4 BLEU points over the small baseline system and 0.3 BLEU points on the system using extended features.

<table>
<thead>
<tr>
<th>System</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>28.91</td>
<td>30.25</td>
</tr>
<tr>
<td>Hard Decision</td>
<td>29.01</td>
<td>30.30 (+0.05)</td>
</tr>
<tr>
<td>Combined Input</td>
<td>29.13</td>
<td>30.47 (+0.22)</td>
</tr>
<tr>
<td>Hidden Combination</td>
<td>29.25</td>
<td>30.62 (+0.37)</td>
</tr>
</tbody>
</table>

Table 10: TED low-resource small systems results.

<table>
<thead>
<tr>
<th>System</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>29.73</td>
<td>31.33</td>
</tr>
<tr>
<td>Hard Decision</td>
<td>29.74</td>
<td>30.84 (-0.49)</td>
</tr>
<tr>
<td>Combined Input</td>
<td>29.97</td>
<td>31.22 (-0.11)</td>
</tr>
<tr>
<td>Hidden Combination</td>
<td>29.87</td>
<td>31.61 (+0.28)</td>
</tr>
</tbody>
</table>

Table 11: TED extended features systems results.
A.6 Morphology-aware Alignments

Morphology-Aware Alignments for Translation to and from a Synthetic Language

Franck Burlot, François Yvon

LIMSI, CNRS, Université Paris Saclay, 91 403 Orsay, France
firstname.lastname@limsi.fr

Abstract

Most statistical translation models rely on the unsupervised computation of word-based alignments, which both serve to identify elementary translation units and to uncover hidden translation derivations. It is widely acknowledged that such alignments can only be reliably established for languages that share a sufficiently close notion of a word. When this is not the case, the usual method is to pre-process the data so as to balance the number of tokens on both sides of the corpus. In this paper, we propose a factored alignment model specifically designed to handle alignments involving a synthetic language (using the case of the Czech:English language pair). We show that this model can greatly reduce the number of non-aligned words on the English side, yielding more compact translation models, with little impact on the translation quality in our testing conditions.

1. Introduction

Most statistical translation models rely on the unsupervised computation of word-based alignments, which both serve to identify elementary translation units, as in phrase-based [11] and hierarchical [2] Machine Translation (MT) and to uncover hidden translation derivations, as in n-gram-based MT [3]. The de-facto standard for computing such alignments is to use the IBM models [4], as implemented in efficient software packages such as GIZA++ [5, 6] or fast_align [7]. It is however widely acknowledged that such alignments can only be reliably established for languages that share a sufficiently close notion of a word. When this is not the case, the usual method is to pre-process the data so as to balance the number of tokens on both sides of the corpus. Assuming translation into English from a morphologically rich language, this process will decompose complex source forms into shorter segments, and/or neutralize morphological variations that are not overly marked (and thus not necessary for the translation process) in the morphologically simpler one: forms that only differ in their case marking can, for instance, be collapsed into one non-marked version for the purpose of translating into English. This situation also occurs, though in a more extreme form, when translating from a language without explicit word separators such as Chinese [8, 9].

This strategy has been successfully applied to many language pairs in the context of MT applications: [10] is a first attempt to cluster morphological variants when translating from German into English; while [11] focuses on splitting German compounds. Similar techniques have been proposed for other language pairs such as Czech [12], Arabic [13, 14], Spanish [15], Finnish [16], Turkish [17] to name a few early studies. Note that the benefits (in terms of translation quality) of such pre-processing can be limited, except for the translation of out-of-vocabulary words.

In this paper, we focus on a slightly different problem, which arises when aligning English with a synthetic language. In this situation, many English words, notably function words such as determiners, pronouns and prepositions, may have no direct equivalent on the source side, in cases where for example their function is expressed morphologically by bound morphemes. Such problems, and their detrimental consequences for MT, are more thoroughly discussed in § 2 taking the Czech:English language pair as the main source of examples. To mitigate this undesirable situation, we propose a factored alignment model specifically designed to handle alignments involving a synthetic language, (see § 3, where we introduce these new variants of IBM Model 2). In our experiments with MT from and into English (§ 4), we show that this model can greatly reduce the number of non-aligned words on the English side, yielding more compact translation models, with little impact on the translation quality in our testing conditions. We finally discuss related work (§ 5) and conclude with further prospects.

2. Alignments with a Synthetic Language

Czech is a morphologically rich language with complex nominal, adjectival and verbal inflection systems. For instance, compared to the English adjective, which is invariable, its Czech counterpart has many different forms, varying in case (7), number (2) and gender (3). Therefore, Czech words contain more information than in English, which is typical of a synthetic language. On the other hand, the same kind of information may be encoded in a separate word in English, a language that has analytical tendencies. For instance, the Czech nominal genitive marker plays a similar role to the English preposition of, as in the English preposition of, as in the engine of the car -> motor auto.

Therefore, when aligning those two languages, linking a Czech noun (or verb, or adjective) solely to its English counterpart is quite unsatisfactory, since the information encoded in the Czech word ending is then lost (see Figure 1);
Table 1: Unaligned preposition causing a mistake (Czech-English).

<table>
<thead>
<tr>
<th>Source</th>
<th>Output</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na seznamu jsou v první řadě plany na rozšíření spolupráci v oblasti jaderné energetiky.</td>
<td>On the list are the first in a series of plans for greater cooperation in the field of nuclear energy.</td>
<td>I go by car</td>
</tr>
<tr>
<td>High on the agenda are plans for greater nuclear co-operation.</td>
<td>jedu autem</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Unaligned preposition causing a mistake (Czech-English).

Figure 1: Lexical alignments missing the English pronoun and preposition that are encoded in the Czech endings.

and it might be desirable to also align neighboring function words on the English side. Missing these links indeed leads to mistakes in the output. In the Moses [18] baseline for Czech to English described in § 4, we often observed that an unaligned English preposition is associated to the wrong phrase, leading to a translation error, as illustrated in Table 1. In this example, the Czech v první řadě means literally in first-Locative rank-Locative and the phrases that were selected incorrectly include prepositions that were not aligned:

- v první - first in: this phrase pair leaves out the translation of the Czech preposition v and includes an English preposition that has no equivalent in the source, and might be erroneously aligned to v.
- řadě - a series of: the Czech locative case is not translated and the English preposition of is not present on the Czech side.

We observe that standard alignment toolkits tend to miss such links. Table 2 reports the ratio of English words that remained unaligned after we trained alignments in both directions with symmetrization, using fast_align. Among the 7% unaligned words, almost 50% are determiners, which was predictable, since Czech does not have articles. Prepositions account for 33.2% of the unaligned words, over 10 points more than what we observe when aligning French and English. A similar situation happens with Russian, where more than 20% of English prepositions have no alignment.

This suggests a difference between languages with synthetic tendencies such as Czech or Russian and more analytical ones such as English in the way they encode grammatical features such as case. When running asymmetric alignments from Czech to English, numbers are even worse, with 52.9% of the English prepositions remaining unaligned. We conclude that there is often no preposition on the Czech side to be linked to an English one. On the contrary, aligning French or Spanish to English means fewer unlinked prepositions and a higher rate of unaligned nouns. Hence, the problem of function word alignments is less obvious and the information we lose the most is lexical, rather than grammatical.

We argue that a more suitable alignment should extract phrases in which the English preposition is more systematically co-aligned with its head noun. This would make the extraction of phrases with a dangling, unaligned of less likely, and contribute to fixing certain case agreement errors.

Unaligned words are not only a problem in terms of the translation of prepositions. Since Czech is a pro-drop language, many English subject personal pronouns have no source to align to, leading to their omissions in many hypotheses translations when translating into English, such as in the clause with no subject found in one of the outputs of our baseline systems and will go into it. Aligning more English pronouns to Czech verbs should help to capture the necessity of jointly translating a verb into a pronoun and a verb in the target. In our English-to-Czech baseline (§ 4), we also often encounter situations where a negative Czech verb is translated into an affirmative form in English. Since Czech negation is encoded as a prefix (ne-, see Table 3), it is difficult to align it to English words such as not.\footnote{The adverb not makes up the majority of unaligned adverbs in Table 2.}

Note that the units we need to find alignments for on the Czech side always encode grammatical information: person, negation and case, which should align to English function words. This is the main motivation for our proposal to add morphological alignments on top of lexical ones.

3. Morphological Alignment Model

3.1. Aligning words with feature vectors

Our model aims to make word-to-word alignments more dense by linking morphological tags on the Czech side to English function words. We first perform a morphological analysis of Czech and obtain a vector-based representation for each token, containing the lemma and various morphological labels (see § 2). Our model thus assumes sentences taking the form of a vector $e$ of $I$ word forms on the English side and of a $K \times J$ matrix $f$ on the Czech side, where each row corresponds to various features of the word (such as lemma, person and case, as shown in Figure 2.a). By convention, we assume that the lemma is at index 1 in vector $f_j$.

Using these notations, our alignment model is a simple variant of IBM model 2 where (a) lemmas are aligned independently from one another, and (b) tag alignments are inde-
Table 2: Unaligned English words with symmetrized alignments across four language pairs using fast_align. POS: rate of unaligned occurrences of the POS over all unaligned words; unali.: rate of unaligned words over all occurrences of the POS.

<table>
<thead>
<tr>
<th>POS</th>
<th>Cs-En (sym)</th>
<th>Cs-En (asym)</th>
<th>Ru-En</th>
<th>Fr-En</th>
<th>Es-En</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PPOS num</td>
<td>PPOS unali.</td>
<td>PPOS num</td>
<td>PPOS unali.</td>
<td>PPOS num</td>
</tr>
<tr>
<td>Determiners</td>
<td>26.2%</td>
<td>65.2%</td>
<td>48.7%</td>
<td>30.1%</td>
<td>16.2%</td>
</tr>
<tr>
<td>Prepositions</td>
<td>28.6%</td>
<td>52.9%</td>
<td>33.2%</td>
<td>15.3%</td>
<td>19.1%</td>
</tr>
<tr>
<td>Auxiliaries</td>
<td>9.7%</td>
<td>37.6%</td>
<td>4.3%</td>
<td>4.4%</td>
<td>5.4%</td>
</tr>
<tr>
<td>Nouns</td>
<td>8.7%</td>
<td>8.8%</td>
<td>3.4%</td>
<td>0.9%</td>
<td>26.7%</td>
</tr>
<tr>
<td>Adverbs</td>
<td>4.9%</td>
<td>26.8%</td>
<td>1.9%</td>
<td>2.5%</td>
<td>3.6%</td>
</tr>
<tr>
<td>Pers. Pronouns</td>
<td>7.3%</td>
<td>65.5%</td>
<td>0.6%</td>
<td>1.2%</td>
<td>2.5%</td>
</tr>
<tr>
<td>Aligned words</td>
<td>72.0%</td>
<td>93.0%</td>
<td>81.6%</td>
<td>90.3%</td>
<td>96.3%</td>
</tr>
</tbody>
</table>

Table 3: Unaligned negation adverb causing a mistake (English-Czech).

<table>
<thead>
<tr>
<th>source</th>
<th>he is not at all aggressive</th>
</tr>
</thead>
<tbody>
<tr>
<td>output</td>
<td>je vůbec agresivní</td>
</tr>
<tr>
<td>ref.</td>
<td>není vůbec agresivní</td>
</tr>
</tbody>
</table>

Figure 2: Morphological alignments. (a) The source 1st person tag is aligned to the target pronoun I and the instrumental case tag to the preposition by (English-Czech). (b) Lemma and tag alignments are merged to provide links between word forms.

\[
p(f|e) = \sum_{a} \prod_{j=1}^{J} \left[ p(a_{1j}|e)p(f_{j1}|e_{a_{1j}}) \right] \times \prod_{k=2}^{K} p(a_{kj}|e)p(f_{jk}|e_{a_{kj}}) \]

This model thus allows us to integrate into the alignment probability the morphological properties of a lemma, which should for instance reinforce the alignment of a Czech noun with an English noun when the former is marked with a case that often matches a nearby preposition of the latter. Note that using the IBM model 2 is somewhat oversimplistic, as it assumes for instance that morphological markers of close words are unrelated, even though agreement rules enforce similar cases for words within the same noun phrase. A more realistic version, in which such dependencies would be modeled at least indirectly, would be to use a better distortion model to constrain the alignment of neighboring lemmas. Given the implementation choices described above, it was not necessary to develop this idea any further.

To complete the description, note that we assume that the alignment of the lemma \(a_{1j}\) only depends on \(j\), \(I\) and \(J\); and that the alignments of the morphological tags \((a_{1j})\) only depend on the difference \((a_{kj} - a_{1j})\). We further enforce \(p(a_{1j}|e) = 0\) outside of a fixed-size window centered on \(a_{1j}\) (3 words to the left side, one word to the right side). The model defined in Equation (1) lends itself well to estimation via EM. We however also performed experiments with more constrained implementations, as described below.

3.2. Implementation variants

In the experiments reported below, we contrast various implementations of this alignment model in the computation of the Czech-to-English alignments; note that we use a standard word-based IBM model for the other direction. A first condition (joint/ibm in Table 10) uses a faithful implementation of EM for the model of Equation (1), in which we initialize uniformly the translation and the distortion parameters.

A second condition uses the output of a first pass alignment to better constrain the alignments of lemmas. The first stage computes alignments between Czech lemmas and English words using standard word alignment pipelines: in our experiments, we used both asymmetric alignments computed with IBM model 2 and IBM model 4, or symmetrized alignments obtained by running these models in both directions. In any case, we keep these alignment links fixed during the second stage, in which we estimate the morphological alignment model and compute alignments links for tags.

A softer version of the second condition is to use the first pass alignments to initialize the translation model, which are then free to change in the course of the EM procedure. Finally note that we also enforce a void alignment for “null” morphological tags (e.g. the case marking for verbs, or the tense of nouns, see Figure 2b).

For all conditions, training involves multiples iterations.
of EM with models of increasing complexity for a fixed number of iterations. We first train the lemma-to-word alignments, before also considering the tags-to-word parameters. A final decoding computes the optimal alignment for morphological tags; at this stage, we only keep alignment links that match a non-aligned word on the English side, and use these to complete the baseline alignment, as shown in Figure 2.b. The rest of the training of the translation model (phrase extraction, etc.) remains unchanged.

4. Experimental Results

4.1. Data and Experimental Setup

We used two datasets to train our SMT systems:

- A small dataset consisting of about 790k parallel sentences taken from the Europarl [19] and News Commentary corpora distributed for the shared translation task of WMT 2015. The monolingual data is made up of one side of the parallel corpora and the News Crawl corpora (2014) and adds up to 29M sentences for English and 37M for Czech.

- A bigger dataset of about 15M parallel sentences, composed of the previous set and the Czeng 1.0 corpus [20]. We added to the monolingual data one side of the Czeng 1.0 corpus and the previous versions of the News Crawl corpora (2007-2013), and obtained a total of 52M Czech and 43M English sentences.

This data is tokenized and true-cased before starting the alignment. The morphological analysis on the Czech side is performed using MorphoDiTa [21]. After word alignment, all downstream training steps are carried out using the Moses toolkit [18]; this includes phrase extraction and scoring, lexical weighting and learning the lexicalized reordering models. 4-gram language models are trained with KenLM [22] and a PPL齐 for both languages. Tuning is performed using MERT [23] on the test set of the WMT 2014 translation task. For the sake of comparison, we also report results obtained with n-gram-based systems trained with Ncode [3, 24].

4.2. Alignment Setup

We used M morphological features to fill the Czech word vectors F in our experiments: case, person, time/mode, and whether a verb has a negative form - Czech representations have therefore M = 5 dimensions.

Regarding condition 1, where lexical alignments are learnt jointly with morphological links (for Czech-to-English), 4 strategies were tested:

- ibm: only forward (cs-en) alignments;
- joint: only forward (cs-en) alignments trained according to our model;
- ibm+morph: forward and backward alignments symmetrized with the grow-diag-final-and heuristics;
- joint/ibm: symmetrization is performed with joint-none and the backward (en-cs) alignments;

Regarding the training condition 2, we used fast_align (resp. Mqiza) to get initial IBM2 (resp. IBM4) alignments between Czech lemmas and English words. We added to the former 3 strategies to obtain different alignment types:

- ibm+morph/none: forward and morphological alignments;
- ibm+morph/ibm: a symmetrized version also involving backward en<cs> alignments;
- [ibm/ibm]+morph: morphological alignment is performed after symmetrization.

During decoding, the most likely morphological alignments are subject to three constraints in order to be accepted:

- The candidate English lemma should not be aligned;
- The morphological alignment probability should be higher than a threshold (0.05 in our experiments);
- The candidate English lemma should have a frequency higher than 1,000 occurrences (15,000 for the bigger data set) in the English part of the parallel corpus.

These heuristics help to improve the quality of alignment by reducing links with rare words that may have a high probability, given a specific tag. Since the words we target are mainly English function words (pronouns, prepositions, etc.), it seems reasonable to focus on a small set of high frequency tokens. Note finally that the same word alignments were used both to train the cs-en and the cs-en systems.

4.3. Results

Morphological alignments effectively address the problem of previously unaligned words by linking function words, as reflected in Table 4, even though ibm+morph/none also returns a few more alignments for nouns. This shows that some lexical alignments had also been wrongly performed, most of which are corrected by symmetrization in the ibm+morph/ibm variant. The first impact of morphological alignments is a reduction of the phrase table size: using fast_align, we lost almost 1.5M phrases when adding morphological alignments to the symmetrized baseline, meaning that over 6% of initial phrases have been discarded (see Table 5).4 Mqiza alignments show the clearest contrast, since the number of phrase pairs for ibm/ibm (44M) is reduced to less than 28M in ibm+morph/ibm.

4 Note that if the number of phrase pairs is lower, the average length of phrases stay the same in every system. For instance, ibm/ibm has 3.77 tokens per Czech phrase and 4.26 per English one, which is very similar to [ibm/ibm]+morph with respectively 3.79 and 4.25 tokens per phrase.
Table 4: Links added by morphological alignments (Czech-English) using fast_align. P OS unali.: rate of unaligned occurrences of the POS over all unaligned words; unali. POS: rate of unaligned words over all occurrences of the POS.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rate</td>
<td>rate</td>
<td>rate</td>
<td>rate</td>
<td>rate</td>
</tr>
<tr>
<td></td>
<td>positions</td>
<td>positions</td>
<td>positions</td>
<td>positions</td>
<td>positions</td>
</tr>
<tr>
<td></td>
<td>unali.</td>
<td>unali.</td>
<td>unali.</td>
<td>unali.</td>
<td>unali.</td>
</tr>
<tr>
<td>Determiners</td>
<td>26.2%</td>
<td>65.2%</td>
<td>32.6%</td>
<td>58.2%</td>
<td>48.7%</td>
</tr>
<tr>
<td></td>
<td>48.7%</td>
<td>30.1%</td>
<td>58.7%</td>
<td>28.3%</td>
<td>51.6%</td>
</tr>
<tr>
<td></td>
<td>24.3%</td>
<td>11.0%</td>
<td>33.1%</td>
<td>10.0%</td>
<td>11.0%</td>
</tr>
<tr>
<td>Prepositions</td>
<td>28.6%</td>
<td>32.9%</td>
<td>25.6%</td>
<td>34.0%</td>
<td>33.2%</td>
</tr>
<tr>
<td></td>
<td>33.2%</td>
<td>15.3%</td>
<td>24.4%</td>
<td>8.8%</td>
<td>31.3%</td>
</tr>
<tr>
<td></td>
<td>11.0%</td>
<td>11.0%</td>
<td>33.1%</td>
<td>11.0%</td>
<td>11.0%</td>
</tr>
<tr>
<td>Auxiliaries</td>
<td>9.7%</td>
<td>37.6%</td>
<td>7.0%</td>
<td>20.6%</td>
<td>4.3%</td>
</tr>
<tr>
<td></td>
<td>4.3%</td>
<td>4.4%</td>
<td>3.3%</td>
<td>2.7%</td>
<td>4.5%</td>
</tr>
<tr>
<td></td>
<td>3.5%</td>
<td>3.5%</td>
<td>3.5%</td>
<td>3.5%</td>
<td>3.5%</td>
</tr>
<tr>
<td>Nouns</td>
<td>4.9%</td>
<td>26.8%</td>
<td>5.0%</td>
<td>19.8%</td>
<td>1.9%</td>
</tr>
<tr>
<td></td>
<td>1.9%</td>
<td>2.5%</td>
<td>2.0%</td>
<td>2.2%</td>
<td>1.9%</td>
</tr>
<tr>
<td></td>
<td>2.0%</td>
<td>2.0%</td>
<td>2.0%</td>
<td>2.0%</td>
<td>2.0%</td>
</tr>
<tr>
<td>Adverbs</td>
<td>7.3%</td>
<td>65.5%</td>
<td>4.7%</td>
<td>25.7%</td>
<td>0.6%</td>
</tr>
<tr>
<td></td>
<td>0.6%</td>
<td>1.2%</td>
<td>0.7%</td>
<td>1.0%</td>
<td>0.7%</td>
</tr>
<tr>
<td></td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
</tr>
<tr>
<td>Pers. Pronouns</td>
<td>26.2%</td>
<td>65.2%</td>
<td>32.6%</td>
<td>58.2%</td>
<td>48.7%</td>
</tr>
<tr>
<td></td>
<td>48.7%</td>
<td>30.1%</td>
<td>58.7%</td>
<td>28.3%</td>
<td>51.6%</td>
</tr>
<tr>
<td></td>
<td>24.3%</td>
<td>11.0%</td>
<td>33.1%</td>
<td>11.0%</td>
<td>11.0%</td>
</tr>
<tr>
<td>Aligned words</td>
<td>72.0%</td>
<td>79.3%</td>
<td>93.0%</td>
<td>94.4%</td>
<td>94.6%</td>
</tr>
</tbody>
</table>

Table 5: Results in BLEU for Czech-English (smaller data condition).

<table>
<thead>
<tr>
<th>Alignment Setup</th>
<th>Ncode</th>
<th>Moses</th>
<th>Phrase Table Size</th>
<th>Moses</th>
<th>Phrase Table Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm//none</td>
<td>50.462,274</td>
<td>20.34</td>
<td>20.31</td>
<td>56.967,921</td>
<td></td>
</tr>
<tr>
<td>ibm+morph//none</td>
<td>35.364,892</td>
<td>20.08</td>
<td>20.14</td>
<td>45,549,682</td>
<td></td>
</tr>
<tr>
<td>ibm+morph//ibm</td>
<td>20,286,841</td>
<td>19.72</td>
<td>20.35</td>
<td>44,410,638</td>
<td></td>
</tr>
<tr>
<td>ibm//ibm</td>
<td>19,72</td>
<td>20.34</td>
<td>21,247,701</td>
<td>20.33</td>
<td>40,805,062</td>
</tr>
</tbody>
</table>

Table 6: Results in BLEU for English-Czech (for the small data condition). The size of the phrase tables is the same as in Table 5.

<table>
<thead>
<tr>
<th>Alignment Setup</th>
<th>Ncode</th>
<th>Moses</th>
<th>Moses</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm//none</td>
<td>13.94</td>
<td>14.24</td>
<td></td>
</tr>
<tr>
<td>ibm+morph//none</td>
<td>13.90</td>
<td>14.03</td>
<td></td>
</tr>
<tr>
<td>ibm+morph//ibm</td>
<td>13.92</td>
<td>13.91</td>
<td></td>
</tr>
<tr>
<td>ibm//ibm</td>
<td>14.02</td>
<td>14.09</td>
<td>14.45</td>
</tr>
</tbody>
</table>

We evaluated our systems using the test set of the WMT 2015 translation shared task. Even though the effect on the BLEU score is minor, we observe a slight improvement when translating into Czech with fast_align\(^5\), which interprets it as a sign of more relevant use of English prepositions in a morphology-aware system.

For the same translation direction, the number of subject personal pronouns is higher in [ibm//ibm]+morph (1,561) than in ibm//ibm (1,501), which suggests better constructions in the English output, such as in Table 9, where the Czech verb with no subject expressed is translated by a verb with its subject pronoun corresponding to the source word ending. Furthermore, handling negation during the alignment step also yields improvement when translating into English. Indeed, the word not has 206 occurrences in ibm//ibm and 234 in [ibm//ibm]+morph, suggesting that the latter system...

\(^5\)The descriptions of our outputs relate to the alignments performed using fast_align.
As described in [16], Turkish [17] to cite a few. Note that the opposite approach, consisting of “splicing” English words into artificially complex forms has also been considered (eg. in [31]).

Aligning English with “morphologically-complex” languages poses several challenges, depending on the exact differences between the source and target – it has, over the years, attracted a considerable amount of effort, which has only been briefly reviewed here. In fact, morphological complexity can have multiple consequences for alignment.

First, it is often assumed that the morphologically complex language has more word types, due for instance to a richer inflectional system: this is the case for French or Spanish, which have a much richer conjugation than English. This, in turn, yields sparser counts, and less reliable probability estimates for the alignment models (notwithstanding a high Out-of-Vocabulary (OOV) ratio at testing time). The simplest remedy is to normalize the target side, using lemmas or other kinds of abstraction instead of words for the purpose of the alignment [25, 26, 27]. Note that defining the optimal level of abstraction is not obvious and often requires a significant tuning effort. Going one step further, it may also be interesting to keep these abstract representations for translation, but this requires a non-trivial post-processing step to restore the correct inflection when translating into the morphologically rich language [28]. The alternative strategy, which translates word forms, is plagued with OOV issues which translates word forms, is plagued with OOV issues – as in the factored-models approach of [29, 30]. In our own alignment model, we borrow the idea to compute a first-pass alignment based primarily on lemmas, which seems to be more effective than using full forms. However, in our case, morphological information is not used to smooth alignment counts, but rather to take account of the function words in the English side.

The other well documented issue with morphologically rich languages is that word forms are more complex, meaning that they are made of several parts (morphemes for basic lexical units, lexemes for compounds). Depending on the language under consideration, identifying the orthographical and/or phonological counterparts of these elementary units can be fairly easy (in the case of purely agglutinative languages) or near impossible (in the case of fusional languages), with a large number of in-between situations. Many rule-based attempts at performing such decompositions as a pre-processing of the source side text have nonetheless been entertained: see [12], Arabic [13, 14], Spanish [15], Finnish [16], Turkish [17] to cite a few. Note that the opposite approach, consisting of “splicing” English words into artificially complex forms has also been considered (eg. in [31]).
As noted by several authors, decomposing word forms into morphemes goes against the main intuition of phrase-based SMT, which favors the translation of large units, and it also reduces the effectiveness of language models, as it decreases the size of the context. To mitigate these potentially negative effects, it is possible to simultaneously consider multiple decomposition schemes, which are then recombined using system combination techniques [32, 33, 34]. This however requires mechanisms to generate multiple morphological decompositions of the same text, using for instance the unsupervised segmentation models of [35, 36, 37]. As pointed out in [38], performing morphological segmentation of the source independently of the target is vastly suboptimal, and joint models for alignment and segmentations are probably more appropriate in a MT context eg. [38, 39].

Our main focus being a fusional language, we have not made any attempt to segment the source words into smaller morphemes, and have instead used a feature-based representation associating a lemma and morphological properties.

6. Conclusions

This paper has described a factored alignment model specifically designed to handle alignments involving a language with synthetic tendencies, such as Czech. We have shown that this model can greatly reduce the number of non-aligned words on the English side, yielding more compact translation models that contain more relevant phrases. Case is the morphological feature that produces most alignments, which turned out to give some improvement when translating into Czech. On the other hand, using time and mode did not bring the expected gain, although it did help to better translate verb inflections in Czech and constructions in English.

The reported improvement over the baseline systems is not confirmed by a straight BLEU improvement. However we showed that one-to-many alignments from Czech to English help to better take into account the specificities of each language. While the English output has more words than in the baseline system, such as negative adverbs, auxiliaries, pronouns (disregarding the fact that it has fewer prepositions), the Czech output is more concise, showing eg. fewer incorrect verbal constructions and more reliance on inflection, which leads to better agreement.

In future work, we intend to confirm these tendencies by (a) using an improved model of morphological alignments, with an improved modeling of the dependency between tags and lemmas, and (b) testing our model with other translation tasks involving a synthetic target language.

7. Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work has been partly funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

8. References

A.7 Output Re-inflection for Rich Morphology

Two-Step MT: Predicting Target Morphology

Franck Burlot, Elena Knyazeva, Thomas Lavergne, François Yvon
LIMSI, CNRS, Univ. Paris-Sud, Université Paris Saclay, 91 403 Orsay, France
firstname.lastname@limsi.fr

Abstract

This paper describes a two-step machine translation system that addresses the issue of translating into a morphologically rich language (English to Czech), by performing separately the translation and the generation of target morphology. The first step consists in translating from English into a normalized version of Czech, where some morphological information has been removed. The second step retrieves this information and re-inflects the normalized output, turning it into fully inflected Czech. We introduce different setups for the second step and evaluate the quality of their predictions over different MT systems trained on different amounts of parallel and monolingual data and report ways to adapt to different data sizes, which improves the translation in low-resource conditions, as well as when large training data is available.

1. Introduction

When translating into a morphologically rich language, statistical machine translation (SMT) systems generally perform poorly, generating several incorrect word forms that show errors in agreement within a noun phrase, that encode the wrong grammatical function of the word in the sentence, or that simply convey the wrong meaning from the source. Such errors come from an important assumption upon which current SMT systems rely: translation is based on a source-target mapping of one or several word forms that are memorized in the model disregarding any broader context.

This assumption can be problematic when both languages involved lack symmetry in their linguistic systems. A more analytical language such as English encodes grammatical information using distinct words (prepositions, negative particles, auxiliary verbs), while a language showing syntactical tendencies such as Czech encodes the same information in word inflection. Moreover, inflection encodes more explicitly the grammatical function of a word in the sentence, while such information is in English encoded in its position in the sentence (e.g., the object is on the right side of the verb).

At the word level, extracting one-to-one word mappings from the parallel data based on alignment links has two main consequences:

- One source word can translate into several target words, leaving difficult choices to be made by the system. This is typically the case of the English adjective

that is invariable and translates into potentially forty-two Czech word forms (varying according to ten cases, three genders and two numbers), not considering lexical ambiguities. When the information is extracted, the context accountable for an inflection is lost.

- Such a variety of word forms that may need to be generated on the target side face important sparsity issues. At test time, one may need to produce on output a word form that has not been seen in the training data, which is challenging and may require extra processing or enriched representations (e.g., factored models).

It may also occur that the word form is contained in the model, but its probability is not well estimated because of its low frequency in the training data. Mapping source to target phrases prevents enough generalization in such a situation.

This paper focuses on word representations in SMT for translation from English to Czech. In order to minimize sparsity on the target side, we proceed to a normalization of the target language by removing grammatical information, such as case, gender and number. This results in a representation of Czech that makes it more symmetric to English. We expect from such a representation an improvement in the translation quality, since the wide variety of choices the translation system has to make is minimized. In this configuration, the SMT system translates from English into a normalized version of Czech. A second translation step is thus necessary and consists in re-inflecting the previously obtained normalized language, in order to output fully inflected Czech. Re-inflection is therefore performed independently of the translation process. It can take advantage of the full context in the output sentence and is also less dependent on the training data, since it may generate word forms that have not been seen in the parallel corpus.

After a description of related work, we will present our system that is built on a translation system that translates from English into a normalized version of Czech (Section 3.1). This output is re-inflected in a second step to turn normalized Czech into a fully inflected language (Section 3.2). Having described our experimental setup (Section 4), we then show that improvement gets lower as the quantity of training data grows (Section 5). Finally, we show a way to better leverage high quantity of data (Section 6).
2. Related work
This paper fits into previous work on two-step machine translation addressing morphology as a post-processing step. Minkov [1] and Toutanova et al. [2] translate from English into Russian (and Arabic) stems, which are used to generate full paradigms, then disambiguated using a classifier. In a comparable way, Chabaneau et al. [3] augment the translation model with synthetic phrases obtained by re-inflecting target stems.

Bojar et al. [4, 5, 6] use two SMT systems: the first one translates from English into Czech lemmas decorated with source-side information and the second one performs a monotone translation into fully inflected Czech. Jawaid and Bojar [7] use in the first step a hierarchical system that outputs a lattice presenting different word orders. The second system then selects the word order that allows for the best morphological predictions.

Fraser et al. [8] represent German words as lemmas followed by a sequence of tags and introduce a linguistically motivated selection of these in order to translate from English. The second step consists in predicting the tags that have been previously removed, using a different CRF for each morphological attribute to predict. Finally, word forms are produced via a look-up in a morphological dictionary. El Kholy and Habash [9, 10] propose a similar approach for Arabic. Weller et al. [11] introduce verbal subcategorization frames enabling the CRFs to make better predictions, and Weller-Di Marco et al. [12] handle the prediction of both prepositions and morphological features by building synthetic phrase tables.

The present work is close to the original idea of Fraser et al. [8] and follows unsuccessful attempts to model target morphology. Marie et al. [13] proposed a similar normalization scheme for translating from English to Russian. Allauzen et al. [14] introduced a hidden CRF model for English into Russian and Romanian aimed at directly predicting the word form, after having generated the full paradigm of the word translated at the previous step.

3. Morphological re-inflection
Our initial assumption is that translation could be easier if the MT model was relieved from having to make hard decisions about morphology. Two-step MT is a way to process morphology apart from the translation process. The first step of the proposed scenario consists in translating from English into normalized Czech. For this purpose, the target side of the parallel and monolingual data have to be pre-processed.

3.1. Normalization of the Czech data
Popovic and Ney [15], Goldwater and McClosky [16] and Durgar El-Kahlout and Yvon [17] show the benefits of normalizing the morphologically rich language (here Czech or German) on the source side when translating into English. Such a normalization consists in grouping different word forms sharing the same lemma into a common class, by removing one or many attributes (e.g. gender, number, case) that are considered as redundant with respect to English. This pre-processing has the effect of reducing the source vocabulary, making both languages more symmetric, and has a positive impact on the translation quality.

When translating in the reverse direction, these ideas hold, but one needs in addition to make sure that the attribute that was removed at normalization step is recoverable from the monolingual context in the SMT output. Indeed, the models we propose for re-inflection do not have access to source side information (see Section 3.2). Therefore, whenever an attribute is redundant with respect to English but is needed for the prediction of other attributes in surrounding words, it needs to be kept.

In our pre-processing, a word is represented as a lemma and a tag sequence, which we obtained using Morphodita [18]. Normalizing such a word simply means removing one or many tags from the sequence. We propose a deterministic schema for each part of speech. The following attributes are preserved:

- **Nouns**: lemma, PoS, gender and number. Number is an attribute that is common to English, and gender is an intrinsic part of Czech nouns, meaning that it may serve to disambiguate two identical lemmas that have a different lexical meaning. Moreover, as head of a noun phrase, the word propagates gender to its dependents. Case is systematically removed and we consider that it should be predictable from the monolingual context.\(^1\)

- **Adjectives**: lemma, PoS, negation, degree of comparison. Since the adjective is invariable in English, we remove gender and number, but keep both negation, which has a lexical value, and the degree of comparison, which is also marked in English.

- **Numerals**: lemma, PoS. English numbers only have one form.

- **Pronouns**: lemma, POS, subPoS, person, gender, number, number[poss], gender[poss]. Only case is removed from pronouns. Gender and number of both possessor ([poss]) and possessed are hard to predict and are generally not well handled in SMT. We leave these attributes and are aware that their prediction would require a special attention that is beyond the scope of this paper [19]. Person is also kept and we expect it to be a useful predictor of nominative case when a pronoun agrees with a verb in the context.

\(^1\)Some contexts make case prediction hard and this attribute should probably sometimes be conveyed by the source, as in the normalized output jim rukama (I eat with my hands). If the case tag is left in the output, the classifier used for re-inflection may ignore the semantic aspect of the clause and consider the noun as a direct object, generating the semantically less likely sentence with accusative case jím ruce (I eat hands).
• Prepositions: word form, POS, case. Here, we keep the word form, since some prepositions have different forms depending on the right side context, e.g. se tehou (with you) vs se mnou (with me). The SMT system handles well this phenomenon. Case is kept, since some prepositions can be followed by different cases and we expect this attribute to propagate through the entire preposition phrase in the output. This choice implies that verb constructions are expected to be handled by the SMT system that is considered to be able to distinguish jít v + Accusative (go to) and bit v + Locative (be in).

• Verb: The lemma and the whole tag sequence are kept. Verbs are not normalized, and we follow the same principle as Fraser et al. [8] that this PoS be considered an anchorage point of the output. The full tag sequence should help distinguish the object from the subject with which it should agree in person, gender, and number.

• Adverb, interjection, conjunction, particle: Word forms are kept, since they are all invariable.

In this setup, only three attributes can be removed: gender, number and case. This constraint makes the tag prediction task easier, since only sequences of three tags need to be predicted (as opposed to sequences of fifteen tags according to the Morphodita tagset\(^2\)). Finally, it allows us to train one different classifier for each attribute (see next section).

3.2. Output re-inflection

The machine translation system outputs a text in a normalized language that needs to be re-inflected. At this step, we have lemmas with a fixed sequence of attributes, some of them having missing values (gender, number and/or case). The task is therefore similar to any sequence labeling problem where the goal is to predict the right value for each empty attribute. When the full tag sequence has been predicted, a dictionary is used to recover the word form corresponding to the attributes and for correcting the predictions made by the previous models.

To extract the features based on previous models, a full decoding of the training data by these models is necessary. To get unbiased predictions, a 10-fold cross-validation is done for the training of the first three models.

The three morphological attributes should be predicted only in words for which they have been removed during the normalization process. Gender, for example, has to be predicted for adjectives but not for nouns. The models are trained in a specific order: gender, number and case are successively trained, then the joint model is learnt. The same order is followed for decoding.

For instance, if nouns have been normalized by removing the case attribute, the morphological generator will output the forms corresponding to each of the seven Czech cases. We end up with a sentence full of ambiguities at different positions. This new sentence is represented as a lattice that is rescored with a 4-gram language model trained on fully inflected Czech sentences.

3.2.2. Cascade of Conditional Random Fields (CRF)

The first supervised model we considered is a CRF [20] that predicts three morphological attributes using the Wapiti toolkit [21]. A joint prediction of all these attributes allows us to better account for the dependencies between them, but such a model can be challenging to train due to the potentially high number of attribute combinations to consider.

A total of 180 different combinations of attributes are observed in our corpus, which are reachable for a CRF model but would require more training data than available to obtain reasonable performance. To overcome this problem, we train a cascade of CRF models, in which the first three models predict a single morphological attribute. That output is used to feed the final joint classifier. The final joint model is therefore only responsible for discovering the dependencies between the attributes and for correcting the predictions made by the previous models.

All four models are trained using 1- to 3-gram word features in an 11-word window as well as 1- to 4-gram features concerning the known morphological information in the same window. Additionally, 1- to 4-gram features on the output of each previous models are used. The models are trained in a specific order: gender, number and case are successively trained, then the joint model is learnt. The same order is followed for decoding.

In order to train the two supervised models we used data from the Universal Dependencies Treebank project\(^3\). We used the Czech and Czech-CAC corpora covering general domain and transcripts of spoken language for a total of 23M words, from which 170k where held out for development.

3.2.1. Language Model (LM)

Each word of the normalized Czech output is re-inflected using an n-gram language model. First, the normalized word is used to query the Morphodita word generator [18]. As a result, we obtain one or several inflected Czech word forms.

For instance, if nouns have been normalized by removing the case attribute, the morphological generator will output the forms corresponding to each of the seven Czech cases. We end up with a sentence full of ambiguities at different positions. This new sentence is represented as a lattice that is rescored with a 4-gram language model trained on fully inflected Czech sentences.

3.2.3. Greedy sequence labeller (Greedy)

As an alternative to the CRF cascade model, a greedy model for sequence labeling was used. The predictions of each attribute (gender, number, case) are performed separately, one after the other, using an SVM multi-class classifier from LBLINEAR library by Fan et al. [22]. During both the training and the decoding process, gender is predicted first, then
number, then case, in a left-to-right order for each attribute.
The feature set is the same as for the CRF model except that it has the possibility of using the 1- to 4-gram features on morphological information predicted for the same attribute.

Another difference with the CRF model is that training examples are extracted only where a prediction should be made. This reduces the number of training examples and helps the model to focus on learning the real task. As for any greedy model, the error propagation problem is crucial here. To deal with it, we apply the SEARN strategy of Daumé et al. [23]. Several iterations of training are performed to alleviate the impact of previously made errors.

More precisely, the search space is generated directly during the learning/decoding process. The states are source-side lemmas and morphological information (given by the MT system), as well as all previously predicted morphological tags. During the first learning iteration, these are extracted from the reference as though decoding has been performed with no mistakes so far.

During the following iterations, we gradually add past decoding errors: for the k-th iteration, the probability of using a previous prediction (possibly erroneous) in the future is equal to $\frac{k}{10}$, otherwise the reference tag is used ($1 \leq k \leq 10$). Thus for the last iteration the search space is as close as possible to the one of the decoder. The action set is composed of all possible combinations of morphological tags. Then, all couples (state, action) produced in this way are used to train the classifier.

3.2.4. Final disambiguation

The latter two systems predict tags that are used to query the Morphodita word generator. At this step, ambiguities are solved using a unigram model, which simply selects the form that has the highest frequency in the training data.\(^4\) We assume that the stylistic level present in the data can be captured in this simple way.

4. Experimental setup

The SMT systems introduced in the following sections are trained with Moses [24] and Ncode [25], and optimized with Mira. 4-gram language models are trained with removed singletons using KenLM [26].

For this task, we used the data provided at both WMT 2016\(^5\) and IWSLT 2016.\(^6\) All systems are optimized on a concatenation of English-to-Czech TED test sets 2010 and 2011, and tested on a concatenation of TED test sets 2012 and 2013. All Czech data is tokenized and truecased using scripts from the Moses toolkit. The English-side tokenization and truecasing relies on in-house text processing tools [27].

Our previous attempts at re-inflection for machine translation suggest that improvement is expected mostly when low amount of either parallel or monolingual data is available. This is the case for under-resourced languages but also to some extent for domain specific translation. For this reason we choose to test our systems on TED talks (transcribed talks) to create a situation similar to low resources, since less parallel data is available. On the other hand, a large quantity of monolingual out-of-domain data is at our disposal to study the effect of corpus size in this context.

We consider Czech as representative of morphologically rich languages, with a complex nominal and adjectival inflection. Many such languages are not provided with a lot of parallel data, such as Bulgarian or Ukrainian. The system described in this paper should improve the translation into these languages as well. Using Czech for our experiments allows us to actually explore the impact of data size on the re-inflection quality, which would not be possible with genuine low-resourced languages.

5. Impact of data size

In this section, we will explore the impact of re-inflection on translation quality in setups involving different amounts of parallel and monolingual data.

5.1. Parallel data

We first explore the parallel data size dimension as this is generally the main limitation in the training of SMT systems for low-resourced languages. We have trained translation systems with increasing amounts of parallel data starting with a small one containing only the first 10k sentences of the TED corpus. The next system corresponds to that same full corpus (117k sentences) which is next increased to 242k sentences by adding the QED corpus,\(^7\) then to 885k sentences after adding Europarl. The final larger system is obtained by appending the news-commentary corpus (1M sentences).

The results of these different systems are shown in Table 1. We observe that Ncode systems have significantly higher results for two-step setups. The CRF and Greedy models provide significant improvements over the baselines (direct English to Czech translation). While there is improvement for all corpus sizes, we notice that as the amount of parallel data grows, the effectiveness of the re-inflexion decreases. This is expected as more parallel data means that the baseline systems have knowledge of more word forms and better statistics on them. With enough parallel data, it is expected that a direct translation system will reach the same performance as the two-step system.

Conversely, the word form selection using language models always deteriorates the baselines. These models are

\(^4\)Our attempt to solve these ambiguities using a 4-gram language model did not give any improvement over the simple unigram model.
\(^5\)http://www.statmt.org/wmt16
\(^6\)http://workshop2016.iwslt.org
\(^7\)http://alt.qcri.org/resources/qedcorpus/
Table 1: BLEU scores for Moses and Ncode systems over direct translations (en2cs) and two-step translations (en2cx2cs). Language models are trained over the target side of the parallel data. As the amount of parallel data grows, the effect of re-inflection gets lower.

<table>
<thead>
<tr>
<th>Data</th>
<th>Moses en2cs</th>
<th>Moses CRF</th>
<th>Moses Greedy</th>
<th>Ncode en2cs</th>
<th>Ncode CRF</th>
<th>Ncode Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>10k</td>
<td>10.06</td>
<td>11.60 (+1.54)</td>
<td>11.64 (+1.58)</td>
<td>10.62</td>
<td>12.13 (+1.51)</td>
<td>12.28 (+1.56)</td>
</tr>
<tr>
<td>117k</td>
<td>15.70</td>
<td>16.70 (+1.00)</td>
<td>16.78 (+1.08)</td>
<td>15.77</td>
<td>17.17 (+1.40)</td>
<td>17.32 (+1.55)</td>
</tr>
<tr>
<td>242k</td>
<td>15.96</td>
<td>16.72 (+0.76)</td>
<td>16.90 (+0.94)</td>
<td>16.06</td>
<td>17.17 (+1.11)</td>
<td>17.32 (+1.26)</td>
</tr>
<tr>
<td>885k</td>
<td>16.75</td>
<td>17.74 (+0.99)</td>
<td>17.94 (+1.19)</td>
<td>16.94</td>
<td>18.04 (+1.10)</td>
<td>18.25 (+1.29)</td>
</tr>
<tr>
<td>1M</td>
<td>17.14</td>
<td>17.64 (+0.51)</td>
<td>17.85 (+0.75)</td>
<td>17.15</td>
<td>17.97 (+0.84)</td>
<td>18.13 (+0.98)</td>
</tr>
</tbody>
</table>

Table 2: BLEU scores for Moses and Ncode systems over direct translations (en2cs) and two-step translations (en2cx2cs). The parallel data used adds up to 885k sentences. As the amount of monolingual data grows, the effect of re-inflection gets lower.

<table>
<thead>
<tr>
<th>Data</th>
<th>Moses en2cs</th>
<th>Moses CRF</th>
<th>Moses Greedy</th>
<th>Ncode en2cs</th>
<th>Ncode CRF</th>
<th>Ncode Greedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>5M</td>
<td>18.01</td>
<td>18.73 (+0.72)</td>
<td>18.84 (+0.83)</td>
<td>17.91</td>
<td>18.69 (+0.78)</td>
<td>18.87 (+0.96)</td>
</tr>
<tr>
<td>10M</td>
<td>18.58</td>
<td>18.87 (+0.29)</td>
<td>19.05 (+0.47)</td>
<td>18.38</td>
<td>18.88 (+0.50)</td>
<td>19.11 (+0.72)</td>
</tr>
<tr>
<td>50M</td>
<td>18.97</td>
<td>19.02 (+0.05)</td>
<td>19.22 (+0.25)</td>
<td>18.96</td>
<td>19.26 (+0.30)</td>
<td>19.53 (+0.57)</td>
</tr>
<tr>
<td>90M</td>
<td>19.34</td>
<td>19.26 (+0.08)</td>
<td>19.51 (+0.17)</td>
<td>19.59</td>
<td>19.52 (+0.07)</td>
<td>19.79 (+0.20)</td>
</tr>
<tr>
<td>200M</td>
<td>20.71</td>
<td>19.75 (+0.96)</td>
<td>20.02 (+0.66)</td>
<td>21.13</td>
<td>20.62 (+0.51)</td>
<td>20.91 (+0.22)</td>
</tr>
</tbody>
</table>

In a low resource context, re-inflexion helps when few parallel data is available. On the other hand, monolingual data is easier and cheaper to obtain and can therefore be used in large amounts.

5.2. Monolingual data

In this section, we will observe how growing monolingual data impact the improvement given by our models. All the following systems use the set of bilingual text of the 885k sentences from the previous section.

On the monolingual side, we use increasing corpus size from 5M to 200M sentences. These corpora include the target side of the parallel data as well as news data, subtitles, and a filtered part of the common-crawl corpus. Results are shown in Table 2. We again notice that the improvement given by the classifiers is higher when the translation is trained with Ncode. For small monolingual data size, the system with re-inflexion improves over the baseline as in the previous section but, as the data grow, the improvement vanishes and even starts to be detrimental for the biggest system. As opposed to the bilingual study, there is enough monolingual data to reach the point where it is possible to build a language model big enough to capture the richness of the fully inflected language efficiently. Such a model is able to make better predictions than the CRF as it can capture 4-gram dependencies between the inflected words where the CRF can only capture 2-gram dependencies. The greedy model, which also capture 4-gram dependencies, shows better results than the CRF but is still outperformed by the baseline for the larger MT system, probably due to its limited amount of training data.

The language models for re-inflexion, that performed poorly on the small data setup, start to be efficient when enough monolingual data is available. With 50M sentences, it improves over the baseline and finally outperforms our best supervised system on the biggest data size. With such amounts of data, it is interesting to note that a two-step system, where the translation is done on normalized Czech with the LM used for re-inflexion, performs better than the baseline with the same LM as it can output words never seen in the parallel data. The 200M system with LM re-inflexion now generates 1138 types and 1485 tokens not seen in parallel data, which makes it more similar to the CRF re-inflexion that outputs 1148 types (1493 tokens), just a few more.

---

8 The CRF re-inflexion of the 117k system generates 1109 types (1503 tokens) that were not seen in the parallel data, while the LM re-inflexion generates only 817 such types (1173 tokens) that were treated by the model as OOVs.

9 This last corpus was filtered by applying the Moore-Lewis method with XooC [28].
Table 3: BLEU scores for direct translations (en2cs) and two-step translations (en2c x2cs), re-inflecting n-best hypothesis from Ncode with different data sets (# parallel sentences / # monolingual sentences).

<table>
<thead>
<tr>
<th>Model</th>
<th>10k/10k</th>
<th>117k/117k</th>
<th>242k/242k</th>
<th>885k/885k</th>
<th>1M/1M</th>
</tr>
</thead>
<tbody>
<tr>
<td>en2cs</td>
<td>15.77</td>
<td>16.06</td>
<td>16.94</td>
<td>17.15</td>
<td></td>
</tr>
<tr>
<td>LM</td>
<td>15.47 (+0.20)</td>
<td>15.81 (+0.33)</td>
<td>16.44 (+0.96)</td>
<td>16.72 (+0.40)</td>
<td></td>
</tr>
<tr>
<td>CRF</td>
<td>17.31 (+1.77)</td>
<td>17.17 (+1.11)</td>
<td>18.24 (+1.30)</td>
<td>18.23 (+0.08)</td>
<td></td>
</tr>
<tr>
<td>+ CRF k-best</td>
<td>17.22 (+1.45)</td>
<td>17.37 (+1.31)</td>
<td>18.55 (+1.61)</td>
<td>18.62 (+1.47)</td>
<td></td>
</tr>
<tr>
<td>Greedy</td>
<td>17.49 (+1.72)</td>
<td>17.65 (+1.59)</td>
<td>18.31 (+1.37)</td>
<td>18.55 (+1.40)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>885k/5M</th>
<th>885k/10M</th>
<th>885k/50M</th>
<th>885k/90M</th>
<th>885k/200M</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM</td>
<td>18.30 (-0.08)</td>
<td>19.20 (+0.24)</td>
<td>19.81 (+0.22)</td>
<td>21.29 (+0.16)</td>
<td></td>
</tr>
<tr>
<td>CRF</td>
<td>19.23 (+0.85)</td>
<td>19.50 (+0.54)</td>
<td>20.02 (+0.43)</td>
<td>21.07 (+0.06)</td>
<td></td>
</tr>
<tr>
<td>+ CRF k-best</td>
<td>19.35 (+0.97)</td>
<td>19.90 (+0.94)</td>
<td>20.24 (+0.65)</td>
<td>21.40 (+0.27)</td>
<td></td>
</tr>
<tr>
<td>Greedy</td>
<td>19.54 (+1.16)</td>
<td>19.84 (+0.88)</td>
<td>20.23 (+0.64)</td>
<td>21.35 (+0.22)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Scores for CRF re-inflection of 1-best and nk-best hypothesis over increasing parallel and monolingual data size.

6. Taking advantage of larger data

Two-step MT is not limited to improvements in low resource conditions. With the systems we have described so far, the prediction of morphology is done within the scope of a fixed set of words in a fixed order, since we only re-inflect the best hypothesis of the MT system. There actually are situations where we could make a better prediction for a word on the condition that this word itself or its position changes in the sentence. We allow such a variation in the output by considering the n-best hypothesis of the MT system (n = 300).

We introduce results obtained by re-inflecting the n-best hypothesis from the MT system (now Ncode only) in Table 3. The re-inflected n-best hypothesis are rescored using MIRA and a language model trained over the monolingual data used for the MT system, except now in naturally inflected Czech. We also have an additional setup where the CRF outputs its k-best predictions (k = 5), leading to the rescoring of nk-best translation hypothesis. Using for this purpose a character-based neural language model provides only slight improvements over an n-gram language model (see Burlot et al. [29]).

We see that both classifiers still show a significant improvement over large systems and start decreasing only after 200M. Figure 1 shows the scores of the CRF for the re-inflection of 1-best and nk-best hypothesis. While using up to 242k parallel sentences for the MT training, the re-inflection of Ncode n-best hypothesis shows no significant improvement in BLEU over the 1-best re-inflection. Furthermore, n-best re-inflection performs better as the amount of data grows. Indeed, with larger language models, the space of the n-best list provides more useful alternatives, of which the morphology prediction models can take advantage. An example of this is shown in Table 4, where the 1-best hypothesis provided an ungrammatical verb frame with a future tense constructed on the auxiliary verb být and a perfective verb, leading to a bad prediction of the dative form for the pronoun. Exploring the n-best hypothesis for re-inflection allowed the model to make the right prediction (accusative) according to a correct verbal frame (with the imperfective verb).

It seems that the deterioration given by the classifiers with bigger MT systems is mainly due to a reduction of the translation quality improvement in the first step. Figure 2 shows the improvement over the baseline obtained with the normalized output (the BLEU score is computed over the normalized reference translation). We understand this BLEU score as a simulation of an ideal situation where...
Table 4: Better morphological predictions with nk-best hypothesis (885k/90M system).

<table>
<thead>
<tr>
<th>Source</th>
<th>CRF 1-best</th>
<th>CRF nk-best</th>
</tr>
</thead>
<tbody>
<tr>
<td>I will bypass you</td>
<td>budu B objiť</td>
<td>budu ti obcházet</td>
</tr>
<tr>
<td>will you-Dative bypass-Perfective</td>
<td>will you-Accusative bypass-Imperfective</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Difference in BLEU score between baseline (cs) and both normalized (cx) and re-inflected outputs (cx2cs) with growing monolingual data.

We are able to predict the same inflections as the reference. Thus, this shows the maximum improvement we can theoretically achieve with the output re-inflation. Here also, as the amount of data grows, improvement decreases: the score of the normalized output goes from a potential improvement of over 4 BLEU points with small data, to barely 3 with larger data. We note that the actual improvement given by the systems is highly correlated to this maximum achievable BLEU score. In a larger data setup, the low improvement is therefore mainly due to the quality of the first step (translation), that is, to take less advantage of Czech normalization. Therefore, making the translation step easier by normalizing the target side helps less when very large data is available.

### 7. Conclusions

We have presented a complete study on the effect of different amounts of parallel and monolingual data on a translation system with re-inflation. The results suggest that re-inflation is more effective when corpora are a scarce resource as with under-resourced languages or domain specific translation. In our experiments we found that, in such a context, even when vast amounts of monolingual data are available, a two-steps MT is still the best choice if we switch from a supervised morphological prediction to an LM when needed.

We have also studied the impact of using the n-best lists from the MT system and showed that, when enough monolingual data is available for an effective rescoring, they improve the overall performance of the system making relevant the use of a re-inflation system in bigger configurations.

These results explain some previous unsuccessful attempts. Weller et al. [11] and Marie et al. [13] obtain small to no improvements on translation into French and German with similar setups. In such cases, a large amount of parallel and monolingual data were used, making classifier predictions useless. Fraser et al. [8] unsuccessfully used the n-best list of the MT system, but on a small system. On such scale, the LM used for the rescoring of the re-inflected n-best is too small to be efficient.

Future works will include the exploration of alternative sequence predictors like the greedy one, which can better capture long range dependencies as our experiments demonstrate, as well as ways to integrate knowledge of the source sentence to improve the predictions. We also plan to investigate automatic ways to perform the normalization instead of our manual selection of the attributes to keep.

### 8. Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work has been partly funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

### References


A.8 Rich Morphology normalisation

Learning Morphological Normalization for Translation from and into Morphologically Rich Languages

Franck Burlot, François Yvon
LIMSI, CNRS, Université Paris-Saclay, France

Abstract

When translating between a morphologically rich language (MRL) and English, word forms in the MRL often encode grammatical information that is irrelevant with respect to English, leading to data sparsity issues. This problem can be mitigated by removing from the MRL irrelevant information through normalization. Such preprocessing is usually performed in a deterministic fashion, using hand-crafted rules and yielding suboptimal representations. We introduce here a simple way to automatically compute an appropriate normalization of the MRL and show that it can improve machine translation in both directions.

1. Introduction

Translating from a morphologically rich language (MRL) like Czech or Russian into a more analytical language like English leads to several issues, due to important divergences in their respective linguistic systems. The MRLs considered in this study have synthetic tendencies, which means that they often encode grammatical information in the endings of words, notably case marks which signal the grammatical function of a word in the sentence. There is no such phenomena in English, where the function of a word is instead encoded in a specific word order or expressed in prepositions. This results in an obvious lack of symmetry between those two types of languages. For instance, while on the MRL side adjectives may vary in gender, number and case, their English translation is invariable. Such differences can impact machine translation (MT) quality in several ways:
• The increase of word forms in the MRL means that each form has a smaller occurrence count than its English counterpart(s), yielding poor probability estimates for infrequent words;
• An even more extreme case is the translation of word forms unseen in training. Even if other forms of the same lemma are known, the MT system cannot generalize and will produce an erroneous output.

A well-known way to mitigate this problem is to “simplify” the MRL by removing information that is deemed redundant with respect to English. This solution has been repeatedly used to translate into the MRL (e.g. in (Ney and Popovic, 2004; Durgar El-Kahlout and Yvon, 2010) for German, (Goldwater and McClosky, 2005) for Czech), and is adopted in recent systems competing at WMT (e.g. (Allauzen et al., 2016; Lo et al., 2016) for Russian), as well as in the reverse direction (Minkov et al., 2007; Toutanova et al., 2008; Fraser et al., 2012) with the additional complexity that the simplified MT output needs to be augmented with the missing information (“re-inflected” in the MT jargon). One downside of these procedures is that they are entirely dependent on the language pairs under study, and rely on hand-crafted rules that need to be adapted for each new language. It is also likely that rule-based normalization is suboptimal with respect to the task, as it does not take the peculiarities of the training data into account.

We introduce (Section 3) a new way to automatically perform such normalization, by clustering together MRL forms. Clustering is performed on a per lemma basis and groups together morphological variants that tend to translate into the same target word(s). We show in Section 4 that this normalization helps when translating into English. A second contribution is a new neural reinflection system, which is crucially able to also take advantage of source-side information, yielding significant improvements when translating into a MRL (Section 5).

2. Related Work

The normalization of the vocabulary on the MRL side mostly consists in removing word information that is deemed redundant with respect to English. Most of the time, normalization relies on expert knowledge specifying which MRL words can be merged without generating confusion in English, (see eg. (Ney and Popovic, 2004; Goldwater and McClosky, 2005; Durgar El-Kahlout and Yvon, 2010)). An alternative, which does not require user expertise is introduced by Talbot and Osborne (2006), who proposed to use model selection techniques to identify useful clusters in the MRL vocabulary. Even though we start from the same intuition (to cluster forms having similar translation distributions), our model is much simpler and more explicitly oriented toward morphological variation, which makes it also easier to invert.

1Our implementation is available at https://github.com/franckbrl/bilingual_morph_normalizer.
The same kind of solution is also useful when translating in the reverse direction; it additionally requires a two-step MT architecture addressing morphology as a post-processing step. Minkov et al. (2007) and Toutanova et al. (2008) translate from English into Russian and Arabic stems, which are used to generate full paradigms, then disambiguated using a classifier. Similarly, Chahuneau et al. (2013) augment the translation model with synthetic phrases obtained by re-inflecting target stems. Bojar (2007) cascade two Statistical MT systems: the first one translates from English into Czech lemmas decorated with source-side information and the second one performs a monotone translation into fully inflected Czech.

Fraser et al. (2012) represent German words as lemmas followed by a sequence of tags and introduce a linguistically motivated selection of these in order to translate from English. The second step consists in predicting the tags that have been previously removed, using a dedicated model for each morphological attribute. Finally, word forms are produced by looking-up in a morphological dictionary. El Kholy and Habash (2012a; 2012b) propose a similar approach for Arabic.

3. Source-side Clustering

3.1. Information Gain

Our goal is to cluster together MRL forms that translate into the same target word(s). We assume that each MRL form \( f \) is a combination of a lemma, a part of speech (PoS) and a sequence of morphological tags, and that a word aligned parallel corpus is available, from which lexical translation probabilities \( p(e|f) \) and unigram probabilities \( p(f) \) can be readily computed. We first consider the simple case where the corpus contains one single lemma for each PoS. We denote respectively \( f \) the set of word forms (or, equivalently, of positions in the paradigm) for this lemma, and \( E \) the complete English vocabulary. The conditional entropy (CE) of the translation model is:

\[
H(E|f) = \sum_{e \in E} p(e|f) H(E|f) = \sum_{e \in E} \frac{p(f)}{\log_2 |E_{\text{en}}|} \sum_{e \in E_{\text{en}}} -p(e|f) \log_2 p(e|f),
\]

where \( E_{\text{en}} \) is the set of words aligned with \( f \). The normalizer \( (\log_2 |E_{\text{en}}|) \) ensures that all the entropy values are comparable, no matter the number of aligned target words.

From an initial state where each form is a singleton cluster, and proceeding bottom-up, we repeatedly try to merge cluster pairs \( (f_1, f_2) \) so as to reduce the CE. We therefore compute the information gain (IG) of the merge operation:

\[
IG(f_1, f_2) = p(f_1) H(E|f_1) + p(f_2) H(E|f_2) - p(f') H(E|f')
\]

\(^2\)For instance, the Czech autem (by car) is represented as: auto + Noun + neutral + singular + instrumental.
where \( f' \) is the resulting aggregate. IG \( (\in [-1, +1]) \) measures the difference between the combined CEs of clusters \( f_1 \) and \( f_2 \) before and after merging in \( f' \). If the corresponding forms have similar translation distributions, the information gain is positive; conversely when their translations are different, it is negative and the merge leads to a loss of information. Note that the total entropy \( H(E_f) \) of the translation model can be recomputed incrementally after merging \((f_1, f_2)\) by:

\[
H(E_{f'}) \leftarrow H(E_f) - IG(f_1, f_2)
\]

Algorithm 1: A bottom-up clustering algorithm

1. \( C(p) \leftarrow \{1, \ldots, n_p\} \)
2. \( i, j \leftarrow \arg \max_{i, j \in C(p)} M_p(i', j') \)
3. repeat
4. Merge i and j in \( C(p) \)
5. for \( l \in V_{\text{lem}} \) do
6. Remove \( L_l(i, j) \), create \( L_l(i, j) \)
7. Compute \( p(i), p(E_{ij}) \) and \( H(E_{ij}) \)
8. Compute \( L_l(i, j) \) for \( k \in C(p) \)
9. \( M_p \leftarrow \sum_{i, j \in V_{\text{lem}}} L_l \)
10. \( i, j \leftarrow \arg \max_{i, j \in C(p)} M_p(i', j') \)
11. until \( M_p(i, j) < m \) or \( |C(p)| = 1 \)

The clustering procedure is described in Algorithm 1. It starts with \( n_p \) classes for each PoS and iteratively performs merge operations, as long as the cumulated information gain for the merge exceeds a minimum threshold \( m \). After each merge,
the statistics for the new cluster (unigram probability, translation probability and entropy) are recomputed for all lemmas and used to update the PoS-level IG matrix $M_p$. When the procedure halts, a clustering $C(p)$ is obtained for PoS $p$, which can then be applied to normalize the source data in various ways (see Section 4.3).

In practice, we obtained slightly better results and a much better runtime than the exact computation of algorithm 1 with an alternative update regime for the IG Matrix $M_p$, which dispenses with the costly update of all the matrices $L_i$ (lines 5–8). Once initialized, $M_p$ is treated like a similarity matrix and updated using a procedure reminiscent of the linkage clustering algorithm. The aggregated matrix cell for clusters $c_1$ and $c_2$ is thus computed as the average IG of all possible 2-way merging operations:

$$M_p(c_1, c_2) = \frac{\sum_{f_1 \in c_1, f_2 \in c_2} M(f_1, f_2)}{|c_1| \times |c_2|}. \quad (4)$$

4. Translating from and into a normalized MRL

We assess the normalization model on MT tasks for three language pairs in both directions: Czech-English, Russian-English and Czech-French; note that the latter involves two MRLs.

4.1. Experimental Setup

Tokenization of English and French uses in-house tools. We used the script from the Moses toolkit (Koehn et al., 2007) for Czech and TreeTagger (Schmid, 1994) for Russian. The MT models are trained using Moses with various datasets from WMT 2016\(^3\) (Table 1). 4-gram language models were trained with KenLM (Heafield, 2011) over the monolingual datasets. These systems are optimized with KB-MIRA (Cherry and Foster, 2012) using WMT Newstest-2015 and tested on Newstest-2016. The Czech-French systems were tuned on Newstest-2014 and tested on Newstest-2013.

<table>
<thead>
<tr>
<th>Setup</th>
<th>parall</th>
<th>mono</th>
<th>parall</th>
<th>mono</th>
<th>parall</th>
<th>mono</th>
<th>parall</th>
<th>mono</th>
<th>parall</th>
<th>mono</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>10k</td>
<td>120M</td>
<td>10k</td>
<td>8.4M</td>
<td>622k</td>
<td>12.3M</td>
<td>622k</td>
<td>9.6M</td>
<td>190k</td>
<td>150M</td>
</tr>
<tr>
<td>Larger</td>
<td>1M</td>
<td>150M</td>
<td>1M</td>
<td>34.4M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Largest</td>
<td>7M</td>
<td>250M</td>
<td>7M</td>
<td>54M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Datasets used to train the MT systems

The source-side normalization is performed independently for each dataset, using the training set of the MT system, except for the Larger and Largest Czech systems for which the parallel data of the Larger system was used. The lemmas and tags are

\(^3\)http://www.statmt.org/wmt16
obtained with Morphodita (Straková et al., 2014) for Czech and TreeTagger (Schmid, 1994; Sharoff et al., 2008) for Russian. Filtering the MRL lemmas when performing clustering yields better results and we have excluded lemmas appearing less than 100 times, as well as word forms occurring less than 10 times in the training set in order to mitigate the noise in the initial alignments. When clustering paradigm cells (see Section 3.2), we set the minimum IG value \( m = 0 \).

### 4.2. A qualitative assessment of normalized Czech

The clustering learned over the Small Czech-English data led to a drastic reduction of the source vocabulary. Starting with 158,914 distinct character strings, corresponding to 237,378 fully disambiguated word forms (represented as lemmas and morphological information), we ended up with a set of 90,170 normalized entries. The resulting clusters confirm some linguistic intuitions. First, nouns turned out to be distinguished only by their number, a property that is also marked for English nouns. We also observed a small number of singleton noun classes, mainly at the instrumental case which often corresponds to the English prepositions *by* and *with* (including the dual number for *rukama* → *[my] hands*), as well as the vocative case. All possessive pronouns were distinguished only by their person, as is also the case in English; adjectives were clustered separately according to their degree of comparison, verbs are clustered by time, the third person singular of the present tense being separated, since it is marked in English (*I cluster*, *he clusters*). We only noticed a small residual noise with negative verbs, sometimes clustered with affirmative ones. This might be due to alignment errors where an English negation particle is not linked to a Czech negative verb, a typical issue for this language pair (Rosa, 2013). Our model thus seems to be able to capture subtle linguistic phenomena that would require a large amount of rules if such normalization had to be performed manually.

### 4.3. MT experiments

The results for all Czech systems are in Table 2 and are reported based on different applications of the normalization model. Indeed, normalization can be used to train both the alignment (*ali cx*) or the full system (*cx2en*), yielding a total improvement of 1.36 BLEU in the Small conditions. Using it only for alignments or only for the MT system gives worse results, still outperforming the baseline (*cs2en*). This shows that both tasks take advantage of the source normalization. Another way to apply the clustering model is to exclude from normalization the 100 most frequent lemmas (100 freq), which gives the best result for this setup. For the other direction (*en2cs*), the Czech normalization was used to train the alignments and gives only a slight improvement over the baseline. Results for the translation into normalized Czech (*en2cx*) after a reinflection step are reported in Section 5.2. The same tendency holds for the Larger Czech-English system, even though the contrasts in BLEU scores are slightly less visible, due to the larger amount of training...
F. Burlot, F. Yvon
Learning Morphological Normalization (49–60)

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs2en (ali cs)</td>
<td>22.62 (±1.36) 1888</td>
<td>24.57 (±0.72) 1610</td>
</tr>
<tr>
<td>cs2en (ali cs)</td>
<td>22.19 (±0.93) 2152</td>
<td>24.14 (±0.29) 1832</td>
</tr>
<tr>
<td>cs2en (ali cs)</td>
<td>22.34 (±1.98) 1914</td>
<td>24.36 (±0.51) 1627</td>
</tr>
<tr>
<td>cs2en (100 freq)</td>
<td>22.32 (±1.56) 1933</td>
<td>24.85 (±1.00) 1614</td>
</tr>
</tbody>
</table>

Table 2. Czech-English systems

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs2en (m = 10^{-4})</td>
<td>24.44 (+0.99) 1604</td>
<td>24.05 (+0.20) 1761</td>
</tr>
<tr>
<td>cs2en (m = 10^{-1})</td>
<td>24.46 (+0.81) 1623</td>
<td></td>
</tr>
<tr>
<td>en2cs (ali cs)</td>
<td>15.54 (+0.33) 17.55 (+0.13) 19.23 (+0.09)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Czech-French systems

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>ru-en (ali ru)</td>
<td>19.76 2260</td>
<td>19.28 (+1.16) 2033</td>
</tr>
<tr>
<td>ru-en (ali ru)</td>
<td>20.92 (+1.16) 2033</td>
<td>20.53 (+0.77) 2048</td>
</tr>
<tr>
<td>ru-en (100 freq)</td>
<td>20.89 (+1.13) 2026</td>
<td></td>
</tr>
<tr>
<td>en-ru (ali ru)</td>
<td>16.59 (+0.36)</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Russian-English systems

The results for Russian-English follow the same tendency as Czech-English, except that keeping the word forms for the 100 most frequent lemmas did not improve over

55
the full normalization of the training set. Finally, we note in Table 3 that the Czech normalization towards French also helps to improve the translation, even though the target language is morphologically richer than English. The improvements are smaller, though, than when translating into English. We assume that this is due to a degree of normalization that is lower when the source shares certain properties with the target, such as adjective inflection, which leads our model to create more classes. Indeed, the model distinguishes nouns by their number, just like with English, but moreover creates separate clusters for each adjective gender. This reduced degree of normalization did not help the training of alignments when translating into Czech (fr2cs).

5. Morphological Reinflection

When translating into a MRL, using normalization to train just the alignments did not prove very helpful (Section 4.3). We now consider using it for the complete translation system. Translating from English into fully inflected Czech however requires a non-trivial post-processing step for reinflection. In this section, we introduce our solution to this problem and provide results for several English-Czech systems.

5.1. A Morphological Reinflection Model

We view the reinflection of the normalized MT output as predicting the fine-grained PoS tag for each output token. Knowing the normalized word and its PoS tag is sufficient to recover the fully inflected word form by dictionary lookup.

Figure 1. RNN architecture for target-side morphology prediction.

For this sequence labeling task, we used a bidirectional recurrent neural network (RNN) that considers both normalized Czech words as well as source-side English
tokens to make its predictions (see Figure 1). It computes a probability distribution
over the output PoS tags $y$ at time $t$, given both the Czech ($f$) and the English ($e$)
sentences, as well as the previous prediction: $p(y_t | f, e, y_{t-1})$.

For each word $f_t$ in the Czech sentence, we need to encode the English words that
generated $f_t$ during the translation process. As there can be an arbitrary number of
them (denoted $I_t$ below), we used a RNN layer4 where each state $S_t$ inputs a source to-
ken representation $a_t; I_t$ and the previous hidden state $S_{t-1}$. The last state (at time $I_t$) of
that layer is used to represent the sequence of aligned tokens: $S_{I_t} = A(S_{I_t-1}; a_{I_t})$.

Each normalized Czech word representation is decomposed into a lemma em-
bding $l_t$ and a cluster embedding $c_t$, which are represented in distinct continuous
spaces. These vectors are concatenated with the source representation $S_{I_t}$, defining
the input to the bidirectional RNN5 performing PoS tagging. A forward layer hidden
state $H_t$ at time $t$ is therefore computed as: $H_t = R(H_{t-1}; [S_{I_t}; l_t; c_t])$. Finally, both
forward and backward layers are concatenated with the representation of the preced-
ing PoS tag $y_{t-1}$ and the result is passed through a last feed-forward layer to which
a softmax is applied. All the model parameters, including embeddings, are trained
jointly in order to maximize the log-likelihood of the training data.

5.2. Experimental Results

The reinflection systems introduced in this section were trained with the parallel
English-Czech data used for the Small setup (News-Commentary). The fine-grained
PoS tags are the same as the ones used to train the normalization in Section 4 (Straková
et al., 2014).7 The word alignments used for the training and validation sets were ob-
tained with fast-align (Dyer et al., 2013). At test time, we use the alignments produced
by the MT decoder. Since the Czech side of the parallel data must be normalized prior
to training, the results below were obtained with two versions of the RNN model:
with the Small data normalization and with the Larger data one (see Section 4).

Each normalized Czech word is associated with a sequence of source English words
that we collect as follows: using word alignments, we take the English words that are
linked to the current position, as well as surrounding unaligned words. These un-
collected words often contain essential information: as shown in (Burlot and Yvon,
2015), many of them have a grammatical content that is helpful to predict the correct
inflection on target side. For instance, the English preposition $of$ is an important pre-

---

4Encoding the sequence of aligned tokens with a “bag of words” model, where we just sum the embed-
dings, performed worse in our experiments.

5The RNN layers for English and normalized Czech contain gated recurrent units (Cho et al., 2014).

6Representing the full left-side target context with an additional RNN did not bring any improvement.

7Our attempts to use the manually annotated data from the Universal Dependency Treebank project
(http://universaldependencies.org) to train a monolingual variant of our model turned out to give worse
results, supposedly because this data is not entirely in-domain.
dictor of the Czech genitive case. This type of grammatical information is the only one that matters for this task, since the lexical content of the Czech words is already computed by the MT system and can not be changed. In fact, replacing the English content words by their PoS and keeping only words in a list of stopwords proved to work better than keeping all the words. Decoding used a beam search of size 5, and the final lookup uses the Morphodita morphological generator.

We consider here three English-Czech MT systems with reinflection. The training data is the same as the Small, Larger and Largest systems described in Section 4, except that the Czech target side is now normalized. The reinflection model can also be used in different ways. One can use it to process the one-best hypothesis of the MT system, or the n-best hypotheses ($n = 300$ in our experiments). A third approach reinflects n-best lists and outputs k-best hypotheses from the reinflection model ($k = 5$ in our experiments). These are finally scored by a language model trained on the same data as the one used in the MT system – albeit with fully inflected words. This score is added to the ones given by the MT system. With nk-best reinflection, we also add the scores given by the reinflection model (log-probability of the predicted sequence). All these scores are finally interpolated using Mira optimization over Newstest-2015 set and produce a single best output sentence.

<table>
<thead>
<tr>
<th></th>
<th>Small System</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU ↑</td>
<td>BEER ↑</td>
<td>CTER</td>
<td>BLEU ↑</td>
<td>BEER ↑</td>
<td>CTER</td>
<td>BLEU ↑</td>
</tr>
<tr>
<td>en2cs (ali cs)</td>
<td>15.21</td>
<td>0.512</td>
<td>0.624</td>
<td>17.42</td>
<td>0.531</td>
<td>0.606</td>
<td>19.24</td>
</tr>
<tr>
<td>en2cs (ali cs)</td>
<td>15.54</td>
<td>0.516</td>
<td>0.627</td>
<td>17.55</td>
<td>0.532</td>
<td>0.607</td>
<td>19.23</td>
</tr>
<tr>
<td>en2cs (1-best)</td>
<td>16.07</td>
<td>0.520</td>
<td>0.626</td>
<td>18.00</td>
<td>0.535</td>
<td>0.599</td>
<td>19.19</td>
</tr>
<tr>
<td>en2cs (n-best)</td>
<td>16.37</td>
<td>0.521</td>
<td>0.621</td>
<td>17.41</td>
<td>0.529</td>
<td>0.591</td>
<td>19.48</td>
</tr>
<tr>
<td>en2cs (nk-best)</td>
<td>16.93</td>
<td>0.525</td>
<td>0.602</td>
<td>18.81</td>
<td>0.540</td>
<td>0.580</td>
<td>19.93</td>
</tr>
</tbody>
</table>

Table 5. BLEU scores for English-Czech

Results are in Table 5, where we provide, in addition to BLEU, scores computed by BEER (Stanojević and Sima’an, 2014) and CharacTER (Wang et al., 2016). These two metrics proved to be more adapted to MRLs by Bojar et al. (2016). We observe a slight improvement when reinflecting the 1-best hypothesis in the Small data conditions. With the Largest dataset, the reinflection has nearly no impact on the translation quality according to BLEU and BEER. Like for the reverse direction, the improvements of normalization get lower as the size of the dataset grows. We were nevertheless able to obtain a reasonable improvement of 0.81 BLEU points over the baseline in the Largest data conditions, which shows that even when a huge quantity of data is available, a specific handling of morphology on target side can still be useful.
6. Conclusion

We have introduced a simple language agnostic way to automatically infer the normalization of a morphologically rich language with respect to the target language that consists in clustering together words that share the same translation, and have shown that it improves machine translation in both directions. Future work will consist in testing our model on neural machine translation systems.

Acknowledgments

This work has been partly funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

Bibliography


Address for correspondence:
Franck Burlot
franck.burlot@limsi.fr
LIMSI-CNRS
Campus Universitaire Orsay, 91 403 Osay, France
A.9 Word Representations in Factored NMT

Word Representations in Factored Neural Machine Translation

Franck Burlot∗
LIMSI, CNRS, Université Paris Saclay

Mercedes García-Martínez∗
LIUM, University of Le Mans

Loïc Barrault
LIUM, University of Le Mans

Fethi Bougares
LIUM, University of Le Mans

François Yvon
LIMSI, CNRS, Université Paris Saclay

Abstract

Translation into a morphologically rich language requires a large output vocabulary to model various morphological phenomena, which is a challenge for neural machine translation architectures. To address this issue, the present paper investigates the impact of having two output factors with a system able to generate separately two distinct representations of the target words. Within this framework, we investigate several word representations that correspond to different distributions of morpho-syntactic information across both factors. We report experiments for translation from English into two morphologically rich languages, Czech and Latvian, and show the importance of explicitly modeling target morphology.

1 Introduction

Open vocabularies remain a challenge for Neural Machine Translation (NMT) (Cho et al., 2014; Bahdanau et al., 2015), both for linguistic and computational reasons. From a linguistic standpoint, morphological variation and lexical productivity cause word forms unseen in training to occur in source texts, which may also require to generate novel target word forms. Using very large input/output vocabularies partially mitigates these issues, yet may cause serious instability (when computing embeddings of rare or unseen words) and complexity issues (when dealing with large softmax layers).

Several proposals have been put forward to address these problems, which are particularly harmful when one language is a morphologically rich language (MRL), exhibiting larger token/type ratio than is observed for English. One strategy is to improve NMT’s internal procedures: for instance by using a structured output layer (Mnih and Hinton, 2008) or by altering the training or decoding criteria (Jean et al., 2015). An alternative approach is to work with representations designed to remove some variations via source-side or target-side normalization procedures; or more radically to consider character-based representations (Ling et al., 2015; Luong and Manning, 2016; Costa-jussà and Fonollosa, 2016), which are however much more costly to train, and make long distance dependencies even longer.

None has however been as successful as the recent proposal of Sennrich et al. (2016b) which seems to achieve a right balance between a limited vocabulary size and an ability to translate a fully open vocabulary. In a nutshell, this approach decomposes source and target tokens into smaller units of variable length (using what is now termed as a “Byte Pair Encoding” or BPE in short): this means that (a) all source tokens can be represented as a sequence of such units, which crucially are all seen in training; (b) all possible target words can also be generated; (c) the size of the output layer can be set to remain within tractable limits; (d) most frequent words are kept as BPE units, which preserves the locality of many dependencies.

In this work, we consider possible ways to extend this approach by also supplying target-side linguistic information in order to help the system generate correct target word forms. Our proposal relies on two distinct components (a) linguistically or data-driven normalization procedures manipulating various source and target word segmentations, as well as eg. multiple factors on the target side (see § 4), and (b) a neural architecture equipped with a dual output layer to predict the target in two simpler tasks generating the lexi-
These components are assessed separately and in conjunction using translation from English into two MRLs: Czech and Latvian. Our experiments show improvement over a strong (Denkowski and Neubig, 2017) BPE-to-BPE baseline, incorporating ensemble of models and backtranslated data (§5). Overall, they suggest that BPE representations, which loosely simulates concatenative morphological processes, is complementary to feature-based morphological representations.

2 Related Work

Translating from and into MRLs has recently attracted some attention from the research community, as these languages compound a number of difficulties for automatic translation, such as the need to analyze or generate word forms unseen in training, or to handle variation in word order.

To mitigate the unknown word problem, a first approach consists in translating into target stems (Minkov et al., 2007; Toutanova et al., 2008); the right form is then selected from the full paradigms in a second step using a classifier. Target words may also be represented as lemmas complemented with side information. Bojar (2007); Bojar and Kos (2010); Bojar et al. (2012) use such a representation for two statistical MT systems: the first one translates from English into Czech lemmas decorated with source-side information and the second one performs a monotone translation into fully inflected Czech.

Fraser et al. (2012) propose a target morphology normalization for German words represented as lemmas followed by a sequence of morphological tags and introduce a linguistically motivated selection of these when translating from English. The selection step consists in predicting the tags that have been removed during normalization, using a specific Conditional Random Field (CRF) model for each morphological attribute to predict. Finally, word forms are produced via look-up in a morphological dictionary. This approach is extended by Weller et al. (2013), who takes verbal subcategorization frames into account, thus enabling the CRFs to make better predictions. Note that Burlot et al. (2016) and El Kholy and Habash (2012b,a) propose related approaches respectively for translating into Czech and Arabic.

Factored word representations have also been considered in neural language models (Niehues et al., 2016; Alexandrescu and Kirchhoff, 2006; Wu et al., 2012), and more recently in a neural machine translation architecture as input features (Sennrich and Haddow, 2016) and in the output by separating the lemma and morphological factors (García-Martínez et al., 2016). One contribution of the current paper is the investigation of new variants of the latter architecture. There have been other attempts with dual training objectives in NMT. In (Chen et al., 2016), a guided alignment training using topic information of the sentence as a second objective helps the decoder to improve the translation. Multi-task and multilingual learning in NMT have also been considered in several papers (Luong et al., 2015; Dong et al., 2015; Firat et al., 2016), where training batches have to carefully balance tasks and language pairs. In contrast to these approaches, our factored NMT (FNMT) system produces several outputs simultaneously.

3 Model Architectures

The baseline NMT system used in this paper is an implementation of a standard NMT model with attention mechanism (Bahdanau et al., 2015). It consists of a sequence to sequence encoder-decoder of two recurrent neural networks (RNN), one used by the encoder and the other by the decoder. This architecture integrates a bidirectional RNN encoder (see bottom left part with green background of Figure 1). Each input sentence word $x_i$ ($i \in 1 \ldots N$ with $N$ the source sequence length) is encoded into an annotation $a_i$, by concatenating the hidden states of a forward and a backward RNN. Each annotation $a_1 \ldots a_N$ thus represents the whole sentence with a focus on the word(s) being processed. The decoder is based on a conditional gated recurrent unit (GRU) (Firat and Cho, 2016) made of two GRUs interleaved with the attention mechanism. The attention mechanism computes a context vector $C_i$ as a convex combination of annotation vectors, where the weights of each annotation are computed locally using a feed-forward network. The decoder RNN takes as input the embedding of the previous output word in the first GRU, the context vector $C_i$ in the second GRU and its hidden state. The softmax output layer is connected to the network through a non-linear layer which takes as input the embedding of the previous output word as well as the context vector and the output of the decoder from the second GRU (both adapted with a linear trans-
formation, respectively, $L_C$ and $L_R$). Finally, the output probabilities for each word in the target vocabulary are computed with a softmax. The word with the highest probability is the translation output at each time step. The encoder and the decoder are trained jointly to maximize the conditional probability of the reference translation.

The Factored NMT system of García-Martínez et al. (2016) is an extension of the standard NMT architecture that allows the system to generate several output symbols at the same time, as presented in Figure 1.

![Factored NMT system](image)

Figure 1: Factored NMT system.

The encoder and the attention mechanism of the Factored NMT are the same as the standard NMT model. However, the decoder has been modified to produce multiple outputs. The two outputs are constrained to have the same length. The decoder feedback is also modified to use information from the multiple output streams. The concatenation of the embeddings of the pair of generated symbols is used to feed the decoder’s cGRU at each timestep.

Two types of FNMT models have been used for this work. Their architecture differs after the generation of the decoder state. The first model contains a single hidden-to-output ($h2o$) layer which is used by the two separate softmax. This layer uses the context vector, the decoder’s hidden state and the concatenation of the embeddings of the previous generated tokens. The second model is one contribution of the current work. As shown in Figure 1), it contains two separated $h2o$ layers. They are similar to the $h2o$ layer in the first model except that instead of using the concatenation of the embeddings of the previously generated factors, each $h2o$ layer receives only the embedding of the factor it is generating. The two separated $h2o$ layers allow the system to have more weights specialized for each output.

4 Word Representations

This paper focuses on the question of word representations, which we understand not only in terms of word segmentation, but also as the quantity of morpho-syntactic information encoded in a word. We introduce three representations varying in the quantity of grammatical information they contain:

- **fully inflected words**: this is a baseline setup where all the lexical and grammatical information is encoded in a single factor.
- **normalized words**: only a well chosen subset of morphological features is kept in the first factor; the second factor corresponds to the Part of Speech (PoS).
- **lemmas**: the output splits the lexical content of the word (first factor: lemma) and its grammatical content (second factor: PoS).

These differences are illustrated in Table 1.

4.1 Normalizing Word Forms

Translating from English into a MRL is made difficult by linguistic divergences, as English lacks many of the morphological contrasts that exist in the MRL. Normalization is needed to reduce the morphological variability on the MRL side so as to limit the number of types in the target, and to mitigate sparsity issues. This strategy is used for instance by Burlot et al. (2016) who remove the case mark from Czech nouns, which is not predictable from their English counterpart(s).

Normalization is usually performed using hand-crafted rules and requires expert knowledge for each language pair. In this paper, normalized words are obtained with an automatic data-driven method introduced in (Burlot and Yvon, 2017b).

In a nutshell, this method performs a clustering of the MRL vocabulary by grouping together words that tend to share the same translation(s) in English. This translational similarity is based on the conditional entropy of lexical translation models estimated, for each MRL word form, using automatic word alignments. The clustering procedure merges two words whenever the resulting cluster does not increase the conditional entropy, which ensures a minimal loss of information during the whole process.

---

1The source code is available at github.com/franckbrl/bilingual_morph_normalizer
The actual normalization algorithm is delexicalized and operates at the level of PoS. Each word is represented as a lemma, a coarse PoS and a sequence of morphological tags (e.g. \texttt{kočka+Noun+Sing+Accusative}). Translational similarities are computed on such words and are combined to provide a PoS-level similarity between two tag sequences. Successive merge operations group into one cluster different such tag sequences. As a result of this procedure, we represent words as a lemma and a cluster identifier (ID) taking the form of a coarse PoS and an arbitrary integer, such as \texttt{kočka+Noun+7} in Table \ref{tab:clusters}. In this example, the cluster ID \texttt{Noun+7} stands for a set of fine-grained PoS, such as \{\texttt{Sing+Nominative}, \texttt{Sing+Accusative}, \ldots\}.

This representation introduces a direct correspondence between the first and the second factor in our architecture, since the former (the cluster ID) constrains the set of possible values of the latter (the fine-grained PoS), which is notably used in our constrained decoding procedure (§ 5.4).

### 4.2 Word Representation Setup

The example of Table \ref{tab:clusters} shows that words are also varying along a second dimension: in addition to considering unsegmented lexical units (be it fully inflected words, normalized words or lemmas), we also investigate the impact of a segmentation of these units using BPE \cite{sennrich16b}.

In this scenario, BPE segmentation is performed on fully inflected words and lemmas. For its application to normalized words, the cluster ID was considered as a minimal unit that cannot be segmented (just like any other character), in order to avoid segmentations like \texttt{kočka+No-un+7}. For these setups, the PoS information (second factor) is replicated for all subparts of a word.

We finally use an alternative representation with normalized words to which BPE segmentation is applied and cluster IDs are systematically split from the lemma. Whenever the FNMT system predicts a lemma in the first factor, it is forced to predict a null PoS in the second factor. On the other hand, when a split cluster ID is predicted, the second factor should output an actual PoS. This specific treatment of the second factor is expected to give the system a better ability to map a word to a compatible PoS, thus avoiding, for instance, the prediction of a verbal PoS for the Czech noun \texttt{kočka} (cat).

These different word representations imply a progressive reduction of the target vocabulary. We computed the vocabulary size of Czech on the parallel data used to train the systems (§ 5.1) over unsegmented words. We thus have 2.1M fully inflected words, 1.9M normalized words, 1.5M normalized words with split clusters (lemmas and clusters), and 1.4M lemmas.

### 5 Experiments

We introduce here the experimental setup for all the reported systems translating from English into Czech and Latvian.

#### 5.1 Data and Preprocessing

Our experimental setting follows the guidelines of the WMT’17\footnote{http://data.statmt.org/wmt17} news translation task. The pre-processing of English data relies on in-house tools \cite{dechelotte08}. All the Czech data were tokenized and truecased the Moses toolkit \cite{koehn07}. PoS-tagging was performed with Morphodita \cite{strakova14}. The pre-processing of Latvian was provided by Tilde.\footnote{www.tilde.com} Latvian PoS-tags were obtained with the LUMII Tagger \cite{paakens13}.

For English-to-Czech, the parallel data used consisted in nearly 20M sentences from a subset of WMT data relevant to the news domain: News-commentary, Europarl and specific categories of the Czeng corpus (news, paraweb, EU, fiction). Newstest-2015 was used for validation and the systems are tested on Newstest-2016 and 2017. The normalization of the Czech data was trained on the parallel data used to train the MT systems, except Czeng fiction and paraweb subcorpora, which amounts to over 10M sentences.

A part of these systems was also trained on synthetic parallel data \cite{sennrich16a} (see § 6). The Czech monolingual corpus News-2016 was backtranslated to English using the single best system provided by the University of Edinburgh from WMT’16.\footnote{www.statmt.org/wmt16} In order to prevent learning from being too biased towards the synthetic source of this set, we used initial bitext parallel data as well. We added five copies\footnote{Adding multiple copies of the same corpus into the training set can be seen as a coarse way to weight different corpora and favor in-domain bitext.} of News-commentary and
The English-to-Latvian systems used all the parallel data provided at WMT'17. The DCEP corpus was filtered with the Microsoft sentence aligner\(^6\) and using modified Moore-Lewis. We kept the best 1M sentences, which led to a total of almost 2M parallel sentences. The systems were validated on 2k sentences held out from the LETA corpus and we report results on Newsdev-2017 and newstest-2017. The normalization of Latvian data was trained on the same parallel sentences used to train the MT systems.

Training was carried out for a part of these systems on synthetic parallel data. We used a back-translation of the monolingual corpora news-2015 and 2016 provided by the University of Edinburgh (Moses system). To these corpora were added 10 copies of the LETA corpus, as well as 2 copies of Europarl and Rapid.

Bilingual BPE models for each language pair and system setup were learned on the bitext parallel data. 90k merge operations were performed to obtain the final vocabularies. For (F)NMT models, the vocabulary size of the second factors is only 1.5k for Czech and 376 for Latvian. The number of parameters in (F)NMT systems increases around 2.5% for Czech and 7% in Latvian.

### 5.2 System Setup

Only sentences with a maximum length of 50 were kept in the training data, except for the setup where cluster IDs were split in normalized words. In this case, we set the maximum length to 100. For the training of all models, we used NMTPY, a Python toolkit based on Theano (Caglayan et al., 2017) and available as free software\(^7\). We used the standard NMT system on fully inflected words and the FNMT architecture described in § 3 on all other word representations.

All systems (F)NMT systems have an embedding dimension of 512 and hidden states of dimension 1024 for both the encoder and the decoder. Dropout is enabled on source embeddings, context vector, as well as output layers. When training starts, all parameters are initialized with Xavier (Glorot and Bengio, 2010). In order to slightly speed up the training on the actual parallel data, the learning rate was set to 0.0004, patience to 30 with validation every 20k updates. On the synthetic data, we finally set the learning rate to 0.0001 and performed validation every 5k updates. These systems were tuned with Adam optimizer (Kingma and Ba, 2014) and have been training for approximately 1 month.

### 5.3 Reinflection

The factored systems predict at each time step a lexical unit and a PoS-tag, which requires a nontrivial additional step producing sentences in a fully inflected language. We refer to this process as reinflection.

Given a lexical unit and a PoS-tag, word forms are retrieved with a dictionary look-up. In the context of MRL, deterministic mappings from a lemma and a PoS to a form are very rare. Instead, the dictionary often contains several word forms corresponding to the same lexical unit and morphological analysis.

A first way to solve this ambiguity is to simply compute unigram frequencies of each word form, which was done over all the monolingual data available at WMT'17 for both Czech and Latvian. During a dictionary look-up, ambiguities can then be solved by taking the most frequent word form. The downside of this procedure is that it ignores important information given by the target monolingual context. For instance, the Czech preposition s (with) will have different forms according to the right-side context: s tebou (with you), but se mnou (with me). A solution is to let an inflected-

---


\(^7\)https://github.com/lium-lst/nmtpy
word-based system select the correct word form from the dictionary. To this end, k-best hypotheses from the dictionary are generated. Given a sentence containing lemmas and PoS, we perform a beam search going through each word and keeping at each step the k-best reinflection hypotheses according to the unigram model mentioned above.

For Czech reinflection, we used the Morphodita generator (Straková et al., 2014). Since we had no such tool for Latvian, all monolingual data available at WMT’17 were automatically tagged using the LU MII Tagger (Paikens et al., 2013) and we gathered the result in a look-up table. As one could expect, we obtained a large table (nearly 2.5M forms) in which we observed a lot of noise.

5.4 Constrained Decoding

The factored system described in § 3 outputs a lexical unit and a PoS-tag at each time step. A peculiarity of this system is that the predictions of both factors are independent. There is only a weak dependency due to the fact that both share the same decoder state and context vector. As a consequence, the best hypothesis for the first factor can well be incompatible with the best hypothesis for the second factor, and the risks of such mismatches only get worse when top-n hypotheses are considered, as in beam search.

Our constrained decoding procedure aims at enforcing a strong consistency between factors. Each word in the target vocabulary is first associated with a specific set of PoS-tags. The decoding procedure is modified as follows: for each candidate target word, we only retain the compatible PoS tags, and select the top-n hypotheses to be kept in the beam from this filtered list. This constraint ensures that the beam search does not evaluate incompatible pairs of factors. (e.g. the PoS Preposition and the word cat).

With a dictionary, creating such a mapping is trivial for full lemmas, but less obvious in the case of BPE units. Since the latter can be generated from different words having different grammatical classes, the size of the set of possible PoS can grow quickly. For normalized words, things are much easier and do not even require a dictionary, as the mapping between cluster IDs and compatible PoS is learnt during the normalization process (see § 4.1). Thus constrained decoding was only performed for (a) unsegmented lemmas, and (b) unsegmented and segmented normalized words.

6 Automatic Evaluation

Results are reported using the following automatic metrics: BLEU (Papineni et al., 2002), BEER (Stanojević and Sima’an, 2014) which tunes a large number of features to maximize the human ranking correlation at sentence level and Character TER (Wang et al., 2016), a character-level version of TER which has shown a high correlation with human rankings (Bojar et al., 2016). Each score on fully inflected word systems is averaged from two independent runs (for both single and ensembled models).

6.1 Experiments with Bitext

The results using the bitext provided at the WMT’17 the evaluation campaign are presented in Table 2 for English-to-Czech and in Table 3 for English-to-Latvian.

We can observe that using the constrained decoding consistently improves the results, except when using split clusters. In this last case, the system is forced to predict a PoS in the second factor whenever it has generated a cluster ID in the first factor. Since there is a reduced quantity of such cluster IDs, the model has no difficulty to learn the constraints by itself and therefore to map a cluster ID exclusively to a specific subset of PoS.

In the Latvian lemma setup, we observe that the improvement using constrained decoding is lower than for Czech (see Table 3), which is probably due to the quality of the noisy look-up table we have created for Latvian (see § 5.1). Note that we have no such dependency on the lexical resources at decoding time for the normalized word setups, where improvements are comparable across both language pairs.

The systems using BPE tokens significantly outperform word-level systems, which confirms the analysis of Sennrich et al. (2016b). The results show that BPE units are even more efficient when applied to normalized words, providing significant improvements over segmented inflected words of 1.79 and 1.85 BLEU points for Czech, and 0.78 and 1.06 for Latvian.

The lemma representation was tested with the two FNMT models presented in § 3, one model using a single hidden-to-output layer (single h2o layer) and the other model using two separated hidden-to-output layers (separated h2o layers).
<table>
<thead>
<tr>
<th>System Type</th>
<th>BLEU</th>
<th>BEER</th>
<th>CTER</th>
<th>BLEU</th>
<th>BEER</th>
<th>CTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>word-to-word</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fully inflected</td>
<td>15.74</td>
<td>47.29</td>
<td>74.79</td>
<td>12.76</td>
<td>44.81</td>
<td>78.90</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factorized Norm</th>
<th>BLEU</th>
<th>BEER</th>
<th>CTER</th>
<th>BLEU</th>
<th>BEER</th>
<th>CTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>sep. h2o layers</td>
<td>16.63</td>
<td>49.78</td>
<td>68.02</td>
<td>13.70</td>
<td>47.13</td>
<td>72.81</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>17.71</td>
<td>50.38</td>
<td>66.94</td>
<td>14.88</td>
<td>47.81</td>
<td>71.44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Factorized Lemmas</th>
<th>BLEU</th>
<th>BEER</th>
<th>CTER</th>
<th>BLEU</th>
<th>BEER</th>
<th>CTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>single h2o layer</td>
<td>16.73</td>
<td>50.50</td>
<td>65.51</td>
<td>14.09</td>
<td>48.15</td>
<td>69.85</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>17.42</td>
<td>50.94</td>
<td>64.95</td>
<td>14.93</td>
<td>48.76</td>
<td>69.26</td>
</tr>
</tbody>
</table>

Table 2: Scores for English-to-Czech systems trained on official bitext data

We observe mixed results, here: the system with the single h2o layer has slightly better results for the word-to-word systems, but the BPE-to-BPE factorized lemma system obtains better performance with the separated h2o layers architecture. For that reason, we decided to only use the separated h2o layers architecture for the next set of experiments involving synthetic data which is the aim of the next section.

6.1.1 Experiments with Selected Bitext and Synthetic Data

Table 4 and 5 show the results of using selected parts of bitext and synthetic parallel data (see section 5.1) for both language pairs. Each model trained with a selection of bitext and synthetic data was initialized with the parameters of its counterpart trained on bitext. The BPE vocabulary used was the same as in the model used for initialization, which led the systems to generate unknown words. In our experiments, we forced the decoder to avoid unknown token generation.

By using synthetic data, we are able to obtain a large improvement for all systems, which is in line with (Sennrich et al., 2016a). We notice that the contrasts present in the previous section between the various word representations are less clear now. The baseline system (first two rows) is the system which benefits the most from the additional data with +5.7 and +6.9 BLEU for Czech and Latvian. The performance of factorized systems has also increased, but to a lesser extent, leading to slightly worse results compared to the baseline system. This situation changes when the reinflected hypotheses are rescored. We are then able to surpass the baseline system with normalized words.

The two language pairs react differently to k-best hypotheses rescoring (+k-best rescored in the tables). For Czech, this has nearly no impact on translation quality according to the metrics, whereas it provides an important improvement in Latvian: +2.03 and +0.84 BLEU in the split cluster setup. Note that this specific setup gives the best score we could achieve on newsdev-2017, without n-best rescoring or model ensembling. We interpret this situation as a result of the difference in quality observed for the Czech and Latvian dictionaries used for re infliction. Indeed, since Morphodita contains exclusively useful Czech re-inflection candidates, a simple unigram model is sufficient to select the right word forms, making the generation of 10-best reinflection hypotheses useless. On the other hand, the hypotheses returned by the look-up table we have used to generate Latvian word forms were noisy and required a rescoring from an MT system based on fully inflected words. We obtained the best results for

---

9Our experiments with 50-best and 100-best reinflections did not show any improvement.
10We assume that the word form generation at this step requires information from the monolingual context only, and could be modeled with a simple target language model, although this needs to be confirmed empirically.
Table 3: Scores for English-to-Latvian systems trained on official bitext.

<table>
<thead>
<tr>
<th>Fully inflected w.</th>
<th>BLEU ↑</th>
<th>BEER ↑</th>
<th>CTER ↓</th>
<th>BLEU ↑</th>
<th>BEER ↑</th>
<th>CTER ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>words-to-words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fully inflected w.</td>
<td>15.15</td>
<td>48.18</td>
<td>76.97</td>
<td>10.61</td>
<td>43.44</td>
<td>85.67</td>
</tr>
<tr>
<td>factored norm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sep. h2o layers</td>
<td>14.10</td>
<td>50.56</td>
<td>69.49</td>
<td>10.42</td>
<td>45.94</td>
<td>78.83</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>15.57</td>
<td>50.78</td>
<td>69.65</td>
<td>11.38</td>
<td>46.28</td>
<td>78.95</td>
</tr>
<tr>
<td>factored lemmas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>single h2o layer</td>
<td>13.96</td>
<td>49.53</td>
<td>68.36</td>
<td>9.68</td>
<td>45.24</td>
<td>77.07</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>14.02</td>
<td>49.48</td>
<td>69.97</td>
<td>9.94</td>
<td>45.21</td>
<td>78.11</td>
</tr>
<tr>
<td>sep. h2o layers</td>
<td>13.92</td>
<td>49.93</td>
<td>68.45</td>
<td>9.71</td>
<td>45.10</td>
<td>77.51</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>14.38</td>
<td>49.74</td>
<td>69.65</td>
<td>10.07</td>
<td>45.26</td>
<td>78.08</td>
</tr>
</tbody>
</table>

Table 4: Scores for English-to-Czech systems (BPE-to-BPE) trained on selected bitext and synthetic parallel data.

<table>
<thead>
<tr>
<th>Fully inflected w.</th>
<th>BLEU ↑</th>
<th>BEER ↑</th>
<th>CTER ↓</th>
<th>BLEU ↑</th>
<th>BEER ↑</th>
<th>CTER ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>words-to-words</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fully inflected w.</td>
<td>23.94</td>
<td>57.30</td>
<td>52.77</td>
<td>20.00</td>
<td>54.35</td>
<td>58.40</td>
</tr>
<tr>
<td>+ ensemble</td>
<td>24.34</td>
<td>57.51</td>
<td>52.48</td>
<td>20.16</td>
<td>54.62</td>
<td>58.22</td>
</tr>
<tr>
<td>factored norm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sep. h2o layers</td>
<td>22.26</td>
<td>56.49</td>
<td>53.43</td>
<td>18.74</td>
<td>53.76</td>
<td>59.18</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>23.02</td>
<td>56.76</td>
<td>53.29</td>
<td>19.34</td>
<td>54.03</td>
<td>58.67</td>
</tr>
<tr>
<td>split clusters</td>
<td>23.37</td>
<td>57.44</td>
<td>52.66</td>
<td>19.77</td>
<td>54.58</td>
<td>58.44</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>23.39</td>
<td>57.43</td>
<td>52.71</td>
<td>19.76</td>
<td>54.59</td>
<td>58.51</td>
</tr>
<tr>
<td>+ k-best rescored</td>
<td>23.43</td>
<td>57.45</td>
<td>52.64</td>
<td>19.79</td>
<td>54.60</td>
<td>58.49</td>
</tr>
<tr>
<td>+ n-best rescored</td>
<td>24.19</td>
<td>57.88</td>
<td>52.19</td>
<td>20.56</td>
<td>54.99</td>
<td>57.96</td>
</tr>
<tr>
<td>+ ensemble</td>
<td>24.55</td>
<td>58.00</td>
<td>51.97</td>
<td>20.68</td>
<td>55.08</td>
<td>57.93</td>
</tr>
<tr>
<td>factored lemmas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sep. h2o layers</td>
<td>22.30</td>
<td>56.63</td>
<td>53.46</td>
<td>19.34</td>
<td>54.16</td>
<td>58.76</td>
</tr>
<tr>
<td>+ k-best rescored</td>
<td>22.35</td>
<td>56.60</td>
<td>53.49</td>
<td>19.36</td>
<td>54.17</td>
<td>58.71</td>
</tr>
<tr>
<td>+ n-best rescored</td>
<td>23.39</td>
<td>57.25</td>
<td>52.73</td>
<td>19.83</td>
<td>54.57</td>
<td>58.35</td>
</tr>
<tr>
<td>+ ensemble</td>
<td>24.05</td>
<td>57.59</td>
<td>52.27</td>
<td>20.22</td>
<td>54.80</td>
<td>57.89</td>
</tr>
</tbody>
</table>

this Latvian setup by generating the 100-best reinflection hypotheses, which provides less dependency on the quality of the dictionary and relies more on the knowledge learned by a word-form-aware system. Despite the fact that such a rescoring procedure is costly in terms of computational time, we observe that it can be a helpful solution when no resources of quality are available.

Czech n-best reinflection, as opposed to k-best, turned out to be efficient, bringing the lemma-based systems to the level of the baselines and even above for the normalized word setups. Whereas it does not improve with Latvian normalized words, we observe a positive impact on the lemma-based systems. We assume that rescoring the n-best list is a way to rely on an inflected-word-based system to make important decisions related to translation, as opposed to the much simpler monolingual process of reinflection mentioned above. Latvian split-cluster models seem to have nothing to learn from such systems.

Factored norm performs best among all the presented models, showing consistent BLEU improvements over the baselines of 0.23 and 0.56 for Czech, and 0.57 and 0.89 for Latvian. We finally notice that ensembling two models slightly reduces those contrasts, and lemma-based systems are the ones that benefit the most from model ensembling. Conclusions are not easy to draw, since across the different setups, the level of indepen-
Table 5: Scores for English-to-Latvian systems (BPEs-to-BPEs) trained on selected bitext and synthetic parallel data.

<table>
<thead>
<tr>
<th></th>
<th>Newsdev-2017</th>
<th>Newstest-2017</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU ↑</td>
<td>BEER ↑</td>
</tr>
<tr>
<td>fully inflected w.</td>
<td>22.05</td>
<td>57.34</td>
</tr>
<tr>
<td>+ ensemble</td>
<td>22.41</td>
<td>57.78</td>
</tr>
<tr>
<td>factored norm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sep. 2x2 layers</td>
<td>18.81</td>
<td>55.65</td>
</tr>
<tr>
<td>+ constrained dec.</td>
<td>20.05</td>
<td>56.14</td>
</tr>
<tr>
<td>split clusters</td>
<td>20.85</td>
<td>56.77</td>
</tr>
<tr>
<td>+ k-best rescored</td>
<td>22.89</td>
<td>57.88</td>
</tr>
<tr>
<td>+ n-best rescored</td>
<td>22.62</td>
<td>57.43</td>
</tr>
<tr>
<td>+ ensemble</td>
<td>22.69</td>
<td>57.61</td>
</tr>
<tr>
<td>factored lemmas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sep. h2o layers</td>
<td>18.93</td>
<td>56.01</td>
</tr>
<tr>
<td>+ k-best rescored</td>
<td>20.56</td>
<td>56.94</td>
</tr>
<tr>
<td>+ n-best rescored</td>
<td>21.59</td>
<td>57.62</td>
</tr>
<tr>
<td>+ ensemble</td>
<td>21.90</td>
<td>57.83</td>
</tr>
</tbody>
</table>

7 Qualitative Evaluation

7.1 Attention in Factored Systems

In a factored NMT setup, the attention mechanism distributes weights across all positions in the input sentence in order to make two predictions, one for each factor, which is an important difference from single-objective NMT. An illustration of the impact of this difference is shown in Figure 2 for the ensembles of two English-to-Czech models introduced in § 6.

In this sentence, the system based on fully inflected words (translation on the top) erroneously predicts the verbal present tense in nevyh’ybâ (does not avoid). We can see that the target subword unit nevy@@ is rather strongly linked to the source didn’t, which allowed the system to correctly predict negative polarity. On the other hand, the ending of the verb b’ is not linked by attention to this same source word, from which the morphological feature of past should have been conveyed. We observe in (a) that the lemma-based system attention aligns the target position to both the source auxil-
Figure 2: An example of attention weight distribution in FNMT (bottom) and fully inflected words (top) output systems aligned to the source sentence (middle) for English-to-Czech. (a) corresponds to the factored lemmas system and (b) factored norm system.

Table 6: Morphological prediction consistency (Entropy).

<table>
<thead>
<tr>
<th>target</th>
<th>nouns</th>
<th>adjectives</th>
<th>verbs</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>system</td>
<td>case</td>
<td>gender</td>
<td>number</td>
</tr>
<tr>
<td>Czech</td>
<td>fully inflected w. factored lemmas</td>
<td>.208</td>
<td>.295</td>
<td>.272</td>
</tr>
<tr>
<td>Latvian</td>
<td>fully inflected w. factored lemmas</td>
<td>.264</td>
<td>.640</td>
<td>.924</td>
</tr>
<tr>
<td></td>
<td>factored norm</td>
<td>.206</td>
<td>.278</td>
<td>.240</td>
</tr>
<tr>
<td></td>
<td>factored norm</td>
<td>.220</td>
<td>.580</td>
<td>.577</td>
</tr>
</tbody>
</table>

D2.3: Final Report: Morphologically Rich Languages

Acknowledgments

This work has been partly funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21) and the French National Research Agency (ANR) through the CHIST-ERA M2CR project, under the contract number ANR-15-CHR2-0006-01.
References


Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Hafeng Wang. 2015. Multi-task learning for multilingual translation. In ACL.


Evaluated the morphological competence of Machine Translation
Systems

Franck Burlot and François Yvon
LIMSI, CNRS, Université Paris-Saclay, France
firstname.lastname@limsi.fr

Abstract

While recent changes in Machine Translation state-of-the-art brought translation quality a step further, it is regularly acknowledged that the standard automatic metrics do not provide enough insights to fully measure the impact of neural models. This paper proposes a new type of evaluation focused specifically on the morphological competence of a system with respect to various grammatical phenomena. Our approach uses automatically generated pairs of source sentences, where each pair tests one morphological contrast. This methodology is used to compare several systems submitted at WMT’17 for English into Czech and Latvian.

1 Introduction

It is nowadays unanimously recognized that Machine Translation (MT) engines based on the neural encoder-decoder architecture with attention (Cho et al., 2014; Bahdanau et al., 2014) constitute the new state-of-the-art in statistical MT, at least for open-domain tasks (Sennrich et al., 2016a). The previous phrase-based (PBMT) architectures were complex (Koehn, 2010) and hard to diagnose, and Neural MT (NMT) systems, which dispense with any sort of symbolic representation of the learned knowledge, are probably worse in this respect. Furthermore, the steady progress of MT engines makes automatic metrics such as BLEU (Papineni et al., 2002) or METEOR (Banerjee and Lavie, 2005) less appropriate to evaluate and compare modern NMT systems. To better understand the strength and weaknesses of these new architectures, it is thus necessary to investigate new, more focused, evaluation procedures.

Error analysis protocols, as proposed eg. by Vilar et al. (2006); Popović and Ney (2011) for PBMT, are obvious candidates for such studies and have been used eg. in (Bentivogli et al., 2016). Recently, various new proposals have been put forward to better diagnose neural models, notably by Linzen et al. (2016); Sennrich (2017), who focus respectively on the syntactic competence of Neural Language Models (NLMs) or of NMT; and by Isabelle et al. (2017); Burchardt et al. (2017), who resuscitate an old tradition of designing test suites.

Inspired by these (and other) works (see § 4), we propose in this paper a new evaluation scheme aimed at specifically assessing the morphological competence of MT engines translating from English into a Morphologically Rich Language (MRL). Morphology poses two main types of problems in MT: (a) morphological variation in the source increases the occurrence of Out-of-Vocabulary (OOV) source tokens, the translation of which is difficult to combine; (b) morphological variation in the target forces the MT to generate forms that have not been seen in training. Morphological complexity is also often associated to more flexible word orderings, which is mostly a problem when translating from a MRL (Bisazza and Federico, 2016). Reducing these issues is a legitimate and important goal for many language pairs.

Our method for measuring the morphological competence of MT systems (detailed in § 2) is mainly based on the analysis of minimal pairs, each representing a contrast that is expressed syntactically in English and morphologically in the MRL. By comparing the automatic translations of these pairs, it is then possible to approximately assess whether a given MT system has succeeded in generating the correct word form, carrying the proper morphological marks. In § 3, we illustrate the potential of our evaluation protocol in a large-scale comparison of multiple MT engines having participated to the WMT’17 News Transla-
tion tasks for the pairs English-Czech and English-Latvian. We finally relate our protocol to conventional metrics (§4), and conclude in §5 by discussing possible extensions of this methodology, for instance to other (sets of) language pairs.

2 Evaluation Protocol

2.1 Morphological competence and its assessment

In traditional linguistics, morphology is “the branch of the grammar that deals with the internal structure of words” (Matthews, 1974, p. 9); the “structure of words” being further subdivided into inflections, derivations (word formation) and compounds. Languages exhibit a large variety of formal processes to express morphological/lexical relatedness of a set of word forms: alternations in suffix/prefix are the most common processes in Indo-European languages, where other language families recourse to circumfixation, reduplication, transfixation, or tonal alternations. They also greatly differ in the phenomena that are expressed through morphological alternations versus grammatical constructions.

Our evaluation protocol is designed to assess the robustness of MT in the presence of morphological variation in the source and target, looking how source alternations (possibly implying to translate source OOVs) are reproduced in the target (possibly implying to generate target OOVs).

The general principle is as follows: for each source test sentence (the base), we generate one (or several) variant(s) containing exactly one difference with the base, focusing on a specific target lexeme of the base; the variant differs on a feature that is expressed morphologically in the target, such as the person, number or tense of a verb; or the number or case of a noun or an adjective. This configuration is illustrated in Table 1, where the first pair is an example of the tense contrast and the second pair an instance of the polarity contrast.

We consider that a system behaves correctly with respect to a given contrast if the translation of the base and the variant reproduce the targeted contrast: for the first example in Table 1, we expect to see in the translation of (1.a) and (1.b) different word forms accounting for the difference of verb tense: the translation of the variant should have a past form and any other case is considered as an error. Other modifications between the two translations, such as the selection of different lemmas for both forms or any modification of the context, are considered irrelevant with respect to the specific morphological feature at study, and are therefore ignored. In the following sections, we detail and justify our strategy for generating contrastive pairs.

2.2 Sentence selection and morphological contrasts

We consider the set of contrasts listed in Table 2. We distinguish three subsets (denoted A, B, and C), which slightly differ in their generation and scoring procedures.

Our choice for selecting this particular set of tests was dictated by a mixture of linguistic and also more practical reasons. From a linguistic standpoint, we were looking to cover a large variety of morphological phenomena in the target language, in particular we wished to include test instances for all open domain word classes (noun, verbs, adjectives). Our first set of tests (set A) is akin to paradigm completion tasks, adopting here a rather loose sense of “paradigm” which also includes simple derivational phenomena such as the formation of comparative for adjectives and mostly checks whether the morphological feature inserted in the source sentence has been translated. Tests in the set B look at various agreement phenomena, while tests in set C are more focused on the consistency of morphological choices. These three categories of tests slightly differ in their generation and scoring procedures.

For each contrast in the A and B sets, sentence generation takes the following steps:2

1. collect a sufficiently large number of short sentences (length < 15) containing a source word of interest for at least one morphological variation;
2. generate a variant as prescribed by the contrast (see below);
3. compute an average language model (LM) score for the pair (base, variant);
4. remove the 33% worst pairs based on their LM score;
5. randomly select 500 pairs for inclusion into the final test.

\[\text{http://statmt.org/wmt17/}\]

--

\(^2\)Examples of test pairs are given as supplementary material in the appendix.
Table 1: Generating minimal contrastive pairs

<table>
<thead>
<tr>
<th>Name</th>
<th>Contrast</th>
<th>Target</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>number</td>
<td>noun</td>
<td>base contains a singular noun, variant contains the plural form</td>
</tr>
<tr>
<td>A-2</td>
<td>number</td>
<td>pronoun</td>
<td>base contains a singular pronoun, variant contains the plural form</td>
</tr>
<tr>
<td>A-3</td>
<td>gender</td>
<td>pronoun</td>
<td>base contains a masculine pronoun, variant contains the feminine form</td>
</tr>
<tr>
<td>A-4</td>
<td>tense:future</td>
<td>verb</td>
<td>base and variant only differ in the tense of the main verb - present in the base, future in the variant</td>
</tr>
<tr>
<td>A-5</td>
<td>tense:past</td>
<td>verb</td>
<td>base and variant only differ in the tense of the main verb - present in the base, past in the variant</td>
</tr>
<tr>
<td>A-6</td>
<td>comparative</td>
<td>adjective</td>
<td>base contains the bare adjective, variant the comparative form</td>
</tr>
<tr>
<td>A-7</td>
<td>polarity</td>
<td>verb</td>
<td>base and variant only differ in the polarity of the main verb - affirmative in the base, negative in the variant</td>
</tr>
<tr>
<td>B-1</td>
<td>complex NP</td>
<td>pronoun</td>
<td>base contains a pronoun, variant contains a complex NP of the form adj noun</td>
</tr>
<tr>
<td>B-2</td>
<td>coordinated noun</td>
<td>pronoun</td>
<td>base contains a pronoun, variant contains a coordinated NP of the form noun and noun</td>
</tr>
<tr>
<td>B-3</td>
<td>coordinated verbs</td>
<td>verbs</td>
<td>base contain a simple verb, variant contains a coordinated VP of the form verb and verb</td>
</tr>
<tr>
<td>B-4</td>
<td>prep-case</td>
<td>preposition</td>
<td>base and variant differ in one preposition which implies a different case in the target (eg. during vs. before, with vs. without)</td>
</tr>
<tr>
<td>C-1</td>
<td>hyponyms</td>
<td>adjective</td>
<td>base contains an adjective, (4) variants with hyponyms</td>
</tr>
<tr>
<td>C-2</td>
<td>hyponyms</td>
<td>noun</td>
<td>base contains a noun, (4) variants with hyponyms</td>
</tr>
<tr>
<td>C-3</td>
<td>hyponyms</td>
<td>verb</td>
<td>base contains a verb, (4) variants with hyponyms</td>
</tr>
</tbody>
</table>

Table 2: A set of morphological contrasts. See text for details.

For set A, the creation of the variant (step 2) consists in replacing a word according to the morphological phenomenon to evaluate (see examples Table 1). This word is selected in such a way that its modification does not require a modification of any other word in the sentence. For instance, a singular subject noun is not replaced by its plural form, since the verb agreeing with it would also need to be replaced accordingly. Indeed, more than one modification would go against our initial idea of generating minimal pairs reflecting exactly one single contrast.

For B-1 (complex NPs), we spot a personal pronoun that we changed into an NP consisting in an adjective and a noun. Both words are generated randomly with the only constraint that the noun should refer to a human subject and the adjective to a psychological state, yielding NPs such as “the happy linguist” or “the gloomy philosopher”. In order to ensure that the context corresponds to a human subject, we selected pronouns that unambiguously refer to humans, such as “him”, “her”, “we” (avoiding “them”). For B-2 (coordinated NPs) the pronoun in the base sentence is transformed into a complex NP consisting of two coordinated nouns. Note that adjectives associated to these nouns, as well as adverbs, have been randomly inserted in order to produce some variation in the constructions. The B-3 contrasts are produced in a similar fashion, targeting verbs instead of nouns, with an additional random generation of a discourse marker that should not interfere with the translation, yielding variants like “he said and, as a matter of fact, shouted”.3 Those inser-

3The coordinated verbs are in bold, the discourse marker is underlined.
tions were performed in order to increase the distance between the two verbs, making agreement between them harder. Finally, the B-4 contrasts are produced in the same way as for the A-set and simply consist in modifying a preposition.

The C-set variants select a noun, an adjective or a verb and replace it with a random hyponym, producing an arbitrary number of sentences. Sentence selection slightly differs from the description above: during step 2, we generate as many variants as possible. Each variant is then scored with a language model and only the top four variants are kept, leading to buckets of five sentences. Those buckets are finally filtered in the same way as for the A and B sets, removing the 33% worst buckets based on their LM score (step 3).

All the sentences were selected from the English News-2008 corpus provided at WMT. The choice of the news domain was dictated by our intention to evaluate systems submitted at WMT’17\(^5\) News Translation task. Sentences longer than 15 tokens were removed in order to ensure a better focus on a specific part of the sentence in the MT output. The modifications of English sentences were based on a morpho-syntactic analysis produced with the TreeTagger (Schmid, 1994) and using the Pymorphy morphological generator\(^5\) to change the inflection of a word. Hyponyms (synonyms and/or antonyms) were generated with WordNet (Miller, 1995). The 5-gram language model used for sentence selection was learned with KenLM (Heafield, 2011) on all English monolingual data available at WMT’15.

2.3 Scoring Procedures

Regarding the scoring procedure, we again distinguish three cases (examples are in Table 3).

- set A: we compare the translations of base and variant and search for the word(s) in variant that are not in base. If one of these words contains the morphological feature associated with the source sentence modification, we report a success. Accuracy of each morphological feature is averaged over all the samples. In this set, we thus evaluate morphological information that should be conveyed from the source sentence, which leads to an assessment on the grammatical adequacy of the output towards the source.

- set B: we compare the translations of base and variant and check that (a) a pronoun in the former is replaced by a NP in the latter (b) the adjective and the noun in the NP share the same gender, number and case. A distinct accuracy rate per feature can then be reported; note that the situation is different in the complex and coordinated tests, as in the latter case some agreement properties may differ in the base and variant (eg. the NP gender agreement depends on the noun gender that may be different from the pronoun gender in base). For the test triggered by prepositions (B-4), we check whether the first noun on the right of a preposition carries the required case mark. Moreover, since we have prepositions associated to nouns in both base and variant, we performed this test on both sentences. This evaluation set checks for agreement and provides an insight about the morphological fluency of the produced translations.

- set C: in this set of tests, we wish to assess the consistency of morphological features with respect to lexical variation in a fixed context; accordingly, we measure the success based on the average normalized entropy of morphological features in the set of target sentences. Such scores can be computed either globally or on a per feature basis. The entropy is null when all variants have the same morphological features, the highest possible consistency; conversely, the normalized entropy is 1 when the five sentences contain different morphological features. For each set C-1, C-2 and C-3, we report average scores over 500 samples. In this setup, we measure the degree of certainty to which a system predicts morphological features across small lexical variations.

Our scoring procedure needs access to morphological information in the target. For A and B sets, the translated sentences are passed through a morphological analysis, where several PoS can be associated with a word. This makes the evaluation less dependent on the tagger’s accuracy. Therefore, when checking whether a specific morphological feature appears in the output (eg. negation of a verb), we look for at least one PoS tag indicating negation, ignoring all the others.

For Czech, we used the Morphodita analyzer (Straková et al., 2014). We had no such resource.
for Latvian and therefore used the LU MII Tagger (Paikens et al., 2013) to parse all Latvian monolingual data available at WMT’17. We then extracted a dictionary consisting of words and associated PoS from the automatic parses. We finally performed a coarse cleaning of this dictionary by removing the PoS that were predicted less than 100 times for a specific word. To run the morphological analysis of Latvian, we parsed the translated sentences with the tagger, then augmented the tagger predictions with our dictionary, producing the desired ambiguous analysis of the Latvian outputs.

For the C-set, the translated sentence analyses are disambiguated: each word is mapped to a single PoS. This was required to compute the entropy. Indeed, we need to select only one morphological value for each base and variant sentence, given that the entropy is normalized according the total number of sentences in the bucket.

<table>
<thead>
<tr>
<th>Base&amp;Variant(s)</th>
<th>Output</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A-set</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I am hungry</td>
<td>mám hlad</td>
<td></td>
</tr>
<tr>
<td>I am not hungry</td>
<td>nemám hlad</td>
<td>negation found</td>
</tr>
<tr>
<td><strong>B-set</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I see him</td>
<td>vidím ho</td>
<td>noun and adjective both</td>
</tr>
<tr>
<td>I see a crazy researcher</td>
<td>vidím bláznivého výzkumníka</td>
<td>have accusative form</td>
</tr>
<tr>
<td><strong>C-set</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I agree with the president</td>
<td>souhlasím s prezidentem</td>
<td>all nouns bear the same PoS</td>
</tr>
<tr>
<td>I agree with the director</td>
<td>souhlasím s řediteltem</td>
<td></td>
</tr>
<tr>
<td>I agree with the minister</td>
<td>souhlasím s ministrem</td>
<td>instrumental case</td>
</tr>
<tr>
<td>I agree with the driver</td>
<td>souhlasím s řidičem</td>
<td></td>
</tr>
<tr>
<td>I agree with the painter</td>
<td>souhlasím s malířem</td>
<td>(Entropy = 0.0)</td>
</tr>
</tbody>
</table>

Table 3: Examples of sentences that pass the tests.

These systems are representative of different models across statistical MT history. Phrase-based systems are a former state of the art that word-based NMT struggled to improve. The new state of the art is an NMT setup with an open vocabulary provided by byte pair encoding (BPE) segmentation (Sennrich et al., 2016b). Finally, we have a set of systems that are optimized in order to improve target morphology. The automatic scores of the systems submitted at WMT’17 are in Table 4 where we report BLEU, BEER (Stanojevic et al., 2014) and CharacTER (Wang et al., 2016).9 We also computed a morphology accuracy for these systems. Using output-to-reference alignments produced by METEOR on lemmas, we computed an accuracy measure for the outputs.

9The test suite and the scripts used for evaluation can be downloaded at github.com/franckbrl/morpheval.

9Chimera (Bojar et al., 2013) consists in a phrase-based factored system (Moses), a deep-syntactic transfer-based system (TectoMT) and a rule-based post-processing system.

Word based NMT: NMT words is a system trained on WMT’17 parallel data with a target vocabulary of 80k tokens. It was not submitted at WMT’17 and is used for contrast.

BPE-based NMT: LIMSI NMT (Burlot et al., 2017) is based on NMTPy (Caglayan et al., 2017), UEDIN NMT (Sennrich et al., 2017a) on Nematus (Sennrich et al., 2017b) and UFAL NMT (Bojar et al., 2017b) on Neural Monkey (Helcl and Libovický, 2017).

NMT modeling target morphology: LIMSI FNMT (Burlot et al., 2017) and LIUM FNMT (García-Martínez et al., 2017) use a factored output predicting words and PoS, and UFAL NMT Chim. (Bojar et al., 2017b) uses Chimera (Bojar et al., 2013). All these models also use BPE segmentation.
Table 4: Scores of the English-to-Czech WMT’17 submissions on the official test set.

checked whether aligned words shared the same form. Our assumption is that two different forms associated to the same lemma correspond to two different inflections of the same lexeme, which allows us to locate positions that likely correspond to morphological errors.

Table 5 lists the results for the A-set tests, which evaluate the morphological adequacy of the output wrt. the source sentence. The last column provides the mean of all scores for one system. We can note that all BPE-based NMT systems have a much higher performance than the phrase-based systems.\footnote{The prediction quality of future tense by PBMT systems is however comparable to that of NMT systems. We assume that this is due to the possibility to generate an analytic form of this tense (auxiliary + infinitive) that is easier to form well than its synthetic form (morphological phenomenon).} We explain the poor performance of the word-based NMT system by the use of a too small closed vocabulary: over the 18,500 sentences of the test suite, 12,016 unknown words were produced by this system. However, when it comes to predicting the morphology of closed class words, this systems performs much better: the accuracy computed for pronoun gender and number is similar to the ones of best BPE-based systems. As opposed to nouns and verbs (open classes), the set of pronouns in Czech is quite small; having observed all their inflections, the word-based system is in a better position to convey the target form.

Despite important differences in automatic metric scores between UEDIN NMT system and LIMSI FNMT, we see that the latter always outperforms the former, except for the feminine pronoun prediction. The overall morphological accuracies (Table 4) show that UEDIN NMT provides more similar word forms with the reference translation, but these global scores fail to show the higher adequacy performance of LIMSI FNMT highlighted in the A-set.

The results of the B-set evaluation for Czech are in Table 6 and are an estimate of the morphological fluency of the output. We observe here again that morphological phenomena such as agreement are better modeled by sequence-to-sequence models using BPE segmentation than phrase-based or word-based NMT systems. The overall best performance of UEDIN and UFAL NMT has to be noted, since both outperform systems that explicitly model target morphology.

The results for the C-set for English-to-Czech are shown in Table 7. We now observe that factored systems are less sensitive to lexical variations and make more stable morphological predictions. The differences with the entropy values computed for the phrase-based systems are spectacular, especially for verbal morphology. We understand this poor performance for phrase-based systems as a consequence of the initial assumption those systems rely on: the concatenation of phrases to constitute an output sentence does not help to provide a single morphological prediction in slightly various contexts.

As an attempt to evaluate the error margin of our evaluation results, we have run a manual check of our evaluation measures. For this, we have taken all 500 sentence pairs reflecting past tense (A-set), as well as case (pronouns to nouns in B-set), and took translations from different systems randomly. We report on cases where the modification of the source created a “bad” (meaningless or ungrammatical) variant, as well as sample translations erroneously considered successful or unsuccessful.

For past tense, we observe a low quantity of false positive (1.6%) and false negative (0.4%). The ratio of bad sources is quite low as well (3%), and is mostly related to cases where a word was given the wrong analysis in the first place, such as a noun labeled by the PoS-tagger as a verb, which was then turned into a past form. For pronouns to nouns, there are nearly no bad source sentences (0.2%): the transformation of pronouns into noun phrases is quite easy and safe. While false positive labels are lower (0.2%), there is a higher amount of false positive (4.4%), which was mainly due to our word-based NMT system that generates many unknown words and presents important differences between base and variant: several adjectives and nouns, not corresponding to the ones we generated in the source sentence, could then be considered during the evaluation.

For English-to-Latvian, we have represented the same types of systems as for Czech, with an additional hybrid system. The scores and mor-
Phonological accuracies of the systems submitted at WMT’17 are in Table 8.

- Phrase-based systems: The Moses baseline was trained on WMT’17 data and TILDE PBMT was provided by TILDE11 and is described in (Peter et al., 2017). These systems did not take part in the official WMT’17 evaluation campaign.

- Word-based NMT: NMT words is a system trained on WMT’17 parallel data with a 80K target vocabulary. It was not submitted at WMT’17 and is used here as a contrast.

- BPE-based NMT: LIMSI NMT (Burlot et al., 2017) is based on NMTPY and UEDIN NMT (Sennrich et al., 2017a) on Nematus.

- NMT modeling target morphology: LIMSI FNMT (Burlot et al., 2017) and LIUM FNMT (García-Martínez et al., 2017) use a factored output predicting words and PoS.

- Hybrid system: TILDE hybrid is an ensemble of NMT models using a PBMT to process rare and unknown words. It was submitted at WMT’17 (Pinnis et al., 2017).

The results for the A-set evaluation are in Table 9. Compared to the previous Czech evaluation, there is a less clear difference between phrase-based and NMT systems based on BPE. Indeed, TILDE hybrid has the best mean performance and is only 5 points above our Moses baseline. A possible reason for that situation is the lower amount of parallel data available for English-Latvian, compared to English-Czech. We notice that there is no significant difference between the two NMT systems and LIMSI FNMT.

With this language pair, word-based NMT performs significantly worse than all other systems on all morphological features, which is confirmed by the fluency evaluation in Table 10. Here, the factored systems tend to have a better verbal fluency, whereas NMT systems perform better on nominal agreement: LIMSI FNMT has the best mean score, but is only 0.2 points above UEDIN NMT. The best system, TILDE hybrid, is now 21.1 points above the Moses baseline, which again seems to be the main reason for such high overall morphological accuracy in Table 8.

Table 11 confirms the higher performance of NMT and factored NMT systems, with a clear advantage for TILDE hybrid, which has the best accuracy in terms of fluency, like in the previous Table 10, which tends to show some correlation between both types of tests.

---

11http://www.tilde.com/mt
When it comes to morphological correction of the output, our evaluation clearly shows the superiority of BPE-based NMT systems over phrase-based ones. On the other hand, while we observed that factored models obtain a higher performance in terms of adequacy, NMT models are still very close to them in terms of fluency. Finally, factored models, as well as TILDE hybrid, clearly showed more confidence in their predictions through slight lexical variations.

4 Related work: evaluating morphology

Automatic metrics Despite their well-known flaws, "general purpose" automatic metrics such as BLEU (Papineni et al., 2002), TER (Snover et al., 2006) or METEOR (Banerjee and Lavie, 2005) remain the preferred way to measure progress in Machine Translation. Evaluation campaigns aimed at comparing systems have long abandoned these measures and resort to human judgments, such as ranking (Callison-Burch et al., 2007) or direct assessment (Bojar et al., 2016). To compensate for the inability of eg. BLEU to detect improvements targeting specific difficulties of MT, several problem-specific measures have been introduced over the years such as the LR-Score (Birch and Osborne, 2010) to measure the correctness of reordering decisions, MEANT (Lo and Wu, 2011) to measure the transfer of entailment relationships, or CharacTER (Wang et al., 2016) to better assess the success of translation into a MRL. Stanojević and Sima’an (2014)’s BEER is a nice example of a sophisticated metric, based on a trainable mixture of multiple metrics: for MRLs, the inclusion of character n-gram matches and of reordering scores proves critical to reach good correlation with human judgments. In comparison, the proposal of Wang et al. (2016) simply computes a TER-like score at the character level, thereby partially crediting a system for predicting the right lemma with the wrong morphology.

Error typologies Error analysis protocols, as proposed by Vilar et al. (2006); Popović and Ney (2011); Stymne (2011) for PBMT systems are obvious candidates for running diagnosis studies and have been used eg. by Bentivogli et al. (2016); Toral Ruiz and Sánchez-Cartagena (2017); Costa-jussà (2017); Klubička et al. (2017). These works differ in the language pairs and in the error typology considered. Bentivogli et al. (2016) only recognizes three main error types which are automatically recognized based on aligning the hypotheses and references — for instance a morphological error is detected when the word form is wrong, whereas the lemma is correct; this definition is also adopted in (Toral Ruiz and Sánchez-Cartagena, 2017), and decomposed at the level of morphological features in (Peter et al., 2016); (Klubička et al., 2017) use a more detailed ty-
The work by Linzen et al. (2016) specifically looks at the prediction of the correct agreement features in increasingly complex contexts generated by augmenting the distance between the head and its dependent and the number of intervening distractors. A language model is deemed correct if it scores the correct agreement higher than any wrong one. One intriguing finding of this study is the very good performance of RNNS, provided that they receive the right kind of feedback in training. A similar approach is adapted for MT by Sennrich (2017), who looks at a wider range of phenomena. Contrastive pairs as automatically produced as follows: given a correct (source, target) pair \( (f, e) \), introduce one error in \( e \) yielding an alternative couple \( (f, e') \). A system is deemed to perform correctly wrt. this contrastive pair if it scores \( p \) higher than \( p' \). This approach is fully automatic, looks at a wide range of contexts and phenomena and facilitates for the pair English-German. Test suites enable to directly evaluate and compare specific abilities of MT Engines, including morphological competences: again, both studies found that NMT is markedly better than PBMT when it comes to phenomena such as word agreement. The downside is the requirement to have expert linguists prepare the data as well as evaluate the success of the MT system, which is a rather expensive price to pay to get a diagnostic evaluation.

**Automatic test suites**

The work by Linzen et al. (2016) specifically looks at the prediction of the correct agreement features in increasingly complex contexts generated by augmenting the distance between the head and its dependent and the number of intervening distractors. A language model is deemed correct if it scores the correct agreement higher than any wrong one. One intriguing finding of this study is the very good performance of RNNS, provided that they receive the right kind of feedback in training. A similar approach is adapted for MT by Sennrich (2017), who looks at a wider range of phenomena. Contrastive pairs as automatically produced as follows: given a correct (source, target) pair \( (f, e) \), introduce one error in \( e \) yielding an alternative couple \( (f, e') \). A system is deemed to perform correctly wrt. this contrastive pair if it scores \( p \) higher than \( p' \). This approach is fully automatic, looks at a wide range of contexts and phenomena and facilitates for the pair English-German. Test suites enable to directly evaluate and compare specific abilities of MT Engines, including morphological competences: again, both studies found that NMT is markedly better than PBMT when it comes to phenomena such as word agreement. The downside is the requirement to have expert linguists prepare the data as well as evaluate the success of the MT system, which is a rather expensive price to pay to get a diagnostic evaluation.

**Test suites**

The work of Isabelle et al. (2017); Burchardt et al. (2017) resuscitates an old tradition of using carefully designed test suites King and Falkedal (1990); Lehmann et al. (1996) to explore the ability of NMT to handle specific classes of difficulties. Test suites typically include a small set of handcrafted sentences for each targeted type of difficulty. For instance, Isabelle et al. (2017) focuses on translating from English into French and is based on a set of 108 short sentences illustrating situations of morphosyntactic, lexico-syntactic and syntactical divergences between these two languages. Assessing a system’s ability to handle these difficulties requires a human judge to decide whether the automated translation has successfully “crossed” the bridge between languages. A similar methodology is used in the work of Burchardt et al. (2017), who use a test suite of approximately 800 segments covering a wide array of translation diffi-

---

12http://www.qt21.eu/mqm-definition

13Note that this is a local evaluation – a system can produce a bad overall translation, yet pass the test.
A typology of evaluation protocols  The variety of evaluation protocols found in the literature can be categorized along the following dimensions:

- *holistic vs analytic*: a holistic metric provides a global sentence- or document-level score, of which the morphological ability is only one part; an analytic metric focuses on specific difficulties;

- *coarse vs fine-grain*: a coarse (analytic) metric only provides global appreciation of morphological competence; while a fine-grain metric distinguishes various types of errors;

- *natural vs hand-crafted vs artificial*: for the sake of this study, this distinction relates to the design of the test sentences – were they invented for the purpose of the evaluation or found in a corpus, or even generated using automatic processing?

- *automatic vs human-judgment*: is scoring fully automatic or is a human judge involved?

- scores can be distance-based, such as a global comparison with a reference translation, or a Boolean value that denotes success or failure wrt. a local test, or based on a comparison of model scores;

Based on this analysis, the work reported here is analytic/fine-grain, uses artificial data, and computes automatic scores based on a local comparison with an expected value (mostly). This is the only one of that kind we are aware of.

5 Conclusion and Outlook

In this paper, we have presented a new protocol for evaluating the morphological competence of a Machine Translation system, with the aim to measure progresses in handling complex morphological phenomena in the source or the target language. We have presented preliminary experiments for two language pairs, which show that NMT systems with BPE outperform in many ways the phrase-based MT systems. Interestingly, they also reveal subtle differences among NMT systems and indicate specific areas where improvements are still needed. This work will be developed in three main directions:

- improve the generation and scoring algorithms: our procedure for generating sentences relies on automatic morphological analysis, which can be error prone, and on crude heuristics. While these two sources of noise likely have a small impact on the final results, which represent an average over a large number of sentences, we would like to better evaluate these effects, and, if needed, apply the necessary fixes;

- refine our analysis of automatic scores: the numbers reported in § 3 are averages over multiple sentences, and could be subjected to more analyses such as looking more precisely at OOVs, or taking frequency effects in considerations. This would allow to assess a system’s ability to generate the right form for frequent vs rare vs unseen lemmas or morphological features. Frequency is also often correlated with regularity, and we also would like to assess morphological competence along those lines. Likewise, analyzing performance in agreement tests with respect to the distance between two coordinated nouns or verbs might also be revealing.

- increase the set of tests: we have focused on translating English into two MRLs having similar properties. Future work includes the generation of additional inflectional contrasts (introducing for instance mood or aspect, which are morphologically marked in many languages) as well as derivational contrasts (such as diminutives for nouns, or antonyms for adjectives). Again, this implies to improve our scoring and generation algorithms, and to adapt them to new languages.

Acknowledgements

The authors thank the participants to the WMT’17 News Translation task who kindly translated our test sets into Latvian and Czech. This work has been partly funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).
References


A.11 Phrase Table Filtering for English-Latvian Translation System

The phrase tables built during the training process of a phrase-based statistical machine translation system (PBSMT) are big and tend to contain a large portion of low quality data, which influences the quality of translation. In this section we explore the possibilities of how to filter out the noisy data during the training process. The aim of this study is to improve the quality of an MT system’s output for the morphologically rich Latvian language. We explored four approaches for phrase table filtering - we have filtered the phrase table directly, we have modified a symmetrized word alignment file that is then used for phrase pair extraction and creation of a phrase table, we have changed the algorithm of the phrase pair extraction module, and we have edited the source to target and target to source alignment files prior to the symmetrization.

A.11.1 Filtering a Phrase Table

By examining the phrase table we noticed several phrase patterns which seem to be incorrect or could unreasonably enlarge the size of the phrase table:

- Every token of the source phrase is a single character, e.g., _ a _ a _ a _
- The source phrase starts with a partial word (clitic), e.g., ' ve or s the;
- The source phrase starts but does not end with a double quote, e.g., " and;
- The source phrase ends but does not start with an apostrophe, e.g., we ‘;
- A single symbol is translated as a word or vice versa;
- The number of tokens in a source phrase is much bigger than in a target phrase or vice versa;
- The probability of translation is very small;
- The lexical probability is very small;
- The probability of direct translation is much smaller that the probability of inverse translation or vice versa.

Based on these observations we defined different rules for discarding a phrase pair from the phrase table. The BLEU score of the baseline system is 28.14, the size of the phrase table is 17.5 GB. Our experimental systems show BLEU scores ranging from 27.41 to 27.72 and the size of the phrase tables varies from 14.4 GB to 16.7 GB. We can conclude that the filtering of the phrase table allows decreasing the size of the phrase table notably, in the same time it does not affect the translation quality significantly.

A.11.2 Modifying a Symmetrized Word Alignment File

The experiments with the phrase table allowed us to decrease the size of the phrase table, but did not result in increase of BLEU score, therefore further experiments were performed on a symmetrized word alignment file.

1. Although word order in Latvian is rather free, in general, word order of Latvian and English sentences is similar – subject-verb-object (SVO). This allows us to assume that there is a strong correlation between the word position in source and target sentences. Although there are some differences in word ordering, for example, genitive noun phrase of type ‘noun1 of noun2’ in English is translated as a ‘noun2(in genitive) noun1’ in Latvian, the changes in word order are not big. If a sentence contains several instances of the same functional word or punctuation mark, sometimes it is aligned to the wrong instance of this word on a target side, causing bad alignment of the whole sentence. We calculate the Pearson correlation coefficient for the alignment points of every sentence pair. By exploring the values of the correlation coefficient, we can detect the sentences with potentially bad alignment. Such sentences are not used for the phrase extraction.

2. Every five consecutive alignment points we observe if the first two and the last two are in linear order and if the middle one differs by more than four positions, we remove it.
<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>Size of the phrase table (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>28.14</td>
<td>17.5</td>
</tr>
<tr>
<td>Experiment 1</td>
<td>28.32</td>
<td>17.2</td>
</tr>
<tr>
<td>Experiments 1 and 2</td>
<td>28.22</td>
<td>17.3</td>
</tr>
<tr>
<td>Experiments 1 and 3</td>
<td>28.23</td>
<td><strong>15.7</strong></td>
</tr>
<tr>
<td>Experiments 1 and 4</td>
<td>27.81</td>
<td>17.6</td>
</tr>
<tr>
<td>Experiments 1 and 5</td>
<td><strong>28.62</strong></td>
<td><strong>15.7</strong></td>
</tr>
</tbody>
</table>

Table 12: BLEU score and phrase table size of the baseline system and systems with the modified word alignment file (the best BLEU score is in bold).

3. We do not extract the phrases from sentences with five or more consecutive unaligned words in order to avoid overgeneration of phrases that differ only by a single word on one side and are the same on the other side.

4. Usually sentences end with a punctuation mark that is correctly aligned. However, if one or several words before the last token are not aligned, there is an overproduction of incorrect phrase pairs, such as one phrase as a single punctuation symbol with a corresponding phrase as a punctuation symbol and some unaligned words. We remove the last alignment point if the previous source or target word is not aligned.

5. We add out-of-range alignment points for unaligned words that are between five or more consecutive unaligned words. Those words will be ignored during the phrase extraction step. Several unaligned points in a row may contain information that is not in the other language, and translations produced by a system trained on such phrases may be inadequate.

Table 12 shows evaluation results for systems trained using the modified alignment file.

A.11.3 Modifying a Phrase Extraction Module

Some issues could not be addressed just by changing the symmetrized word alignment file. Therefore, we made the following changes in the extraction module and evaluated their impact on MT output:

1. Sentences with low correlated alignment points were not discarded completely. The fragments with a good correlation were kept (three or more tokens in a linear order in the source and in the target phrase).

2. Ignore a single word phrases translated as a single character or vice versa (except the single symbol token I).

3. If a source token is unaligned but one token before it is aligned with two adjacent target tokens then move the alignment point to make a linear alignment (the first source token to the first target token, the second source token to the second target token).

4. Ignore phrases with five or more unaligned tokens.

5. If a single word phrase is preposition, article, conjunction, adverb, pronoun or auxiliary verb (a, an, are, in, of, the, to, at, away, back, before, below, but, by, for, he, his, if, now, on, or, she, then, their, then, therefore, under, very, when, with) then only a single predefined translation is accepted.

Table 13 shows evaluation results for systems where different extraction modules were used in training.
## A.11.4 Modifying Alignment Files prior to Symmetrization

By exploring the source to target and target to source alignment files, we have noticed some alignment issues which we try to solve before symmetrization:

- The clitics ‘s, ‘t, ‘re, ‘m do not have adjacent alignment points with the words/tokens to which they are connected. As a result, they are not extracted together with a related word/token. By adding missing points we unite ‘s and ‘t with a previous word, but ‘re and ‘m is united with a next word, if it is a present participle.
- Sometimes for common phrases, the alignment results is a two-word source and target phrase pair, while separate words are not included in alignment. We change the alignment points in a way that both – the two-word phrase pair and each word separately – are extracted.
- We review and fix alignment for the phrases expressing negation.
- If sentences have several commas, then a single comma on a source side could be aligned to the several commas on a target side or vice versa. We remove the extra alignment points to solve this problem.
- If sentences contain several quotes, then there could be a similar problem as with a several comma alignment. We remove the extra alignment points to solve this problem.
- The lists of the words can be separated by the conjunctions or commas. If there are several such item separators there could be problems with an alignment. We make the list alignment consistent by shifting points or removing extra points.
- In English genitive noun phrases with a structure ‘x of y’ are very common. These correspond to the Latvian noun phrase with the structure ‘y (in genitive) x’. We change alignment to tie together all the words of such phrases.
- A functional word could have a wrong alignment to a content word. We have defined the lists of functional words for the source and for the target language. If the functional word is aligned to non-functional word and it does not have an adjacent both-sided alignment point, the alignment for this functional word is removed.

Table 14 shows evaluation results for the systems where different alignment issues were addressed.

## A.12 Source Discriminative Word Lexicon

The correct word form in a morphologically-rich language often has long-range dependencies. In order to better disambiguate the translations, KIT propose modeling the translation as a prediction task. The prediction is motivated by the discriminative word lexicon (DWL). While the DWL operates on the target side and learns to predict for each target word whether it should occur in a given target sentence, the source discriminative word lexicon (SDWL) operates on the source side. For every source word a classifier is trained to predict its translation in the
<table>
<thead>
<tr>
<th>System/issue</th>
<th>BLEU (test corpus)</th>
<th>BLEU (dev. corpus)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>28.09</td>
<td>22.63</td>
</tr>
<tr>
<td>1. Clitics</td>
<td>28.52</td>
<td>22.75</td>
</tr>
<tr>
<td>2. Common phrases</td>
<td>28.36</td>
<td>23.02</td>
</tr>
<tr>
<td>3. Negative expressions</td>
<td>28.55</td>
<td>22.75</td>
</tr>
<tr>
<td>4. Several commas</td>
<td>28.47</td>
<td>22.33</td>
</tr>
<tr>
<td>5. Several quotes</td>
<td>28.20</td>
<td>22.42</td>
</tr>
<tr>
<td>6. The lists of the words</td>
<td>28.47</td>
<td>23.02</td>
</tr>
<tr>
<td>7. ‘x of y’ type phrases</td>
<td>28.28</td>
<td>22.64</td>
</tr>
<tr>
<td>8. Functional/content word alignment</td>
<td>28.11</td>
<td>22.73</td>
</tr>
<tr>
<td>9. Cumulative system</td>
<td>28.50</td>
<td>22.86</td>
</tr>
</tbody>
</table>

Table 14: BLEU scores for the systems with fixed alignment issues (the scores exceeding the baseline are in bold).

given sentence. A multi-class classification task is performed by identifying for every source word the 20 most frequent translations, which are provided by the word alignment generated using GIZA++. All words in the target language that occur less frequently than the 20 most frequent words are assigned to one class, called other. Alignments to the NULL word on the target side are treated in the same way as if NULL were a word. The source vocabulary is limited to the words occurring in the test data and train up to 20 classifiers for each source word. In reality, most words have much fewer alternative translation options than 20. The SDWL uses binary maximum entropy classifiers trained using the one-against-all scheme. That means a maximum entropy model is used to estimate \( p(e|f,c(f)) \), where \( e \) is the target word that should be predicted given source word \( f \) and its context/dependency features \( c(f) \). During training, the maximum entropy models for the individual classes for each source word are trained based on the given set of features extracted from the source sentence and the correct class of each training example. For the prediction, the test data is first separated into words. For each word the features are extracted from the source sentence it stems from. Then all binary maximum entropy models for the multiple classes are applied and each of them produces a prediction. The final prediction corresponds to the class with the highest prediction probability.

A.12.1 Structural Features

The training examples and test data for the classifiers are represented by a set of features and the class each example belongs to. Different types of features representing the structure of a sentence to varying degrees were evaluated.

Bag-of-Words A straight forward way to represent the source sentence for this classification task is to use the bag-of-words approach. This is the least structural informative feature which does not provide any knowledge about the sentence beyond the mere existence of the words in it.

Context The context feature adds structural information about the local context of the modeled source word in the sentence. In addition to the context words themselves, their position is encoded in the feature such that the same word occurring at a different position (relative to the source word in question) would result in a different feature. Up to six context words were included, three on each side of the source word. Hence, this feature type provides structural information by means of sequential order within a limited context.

Dependency Relations The feature contributing the most information about the sentence structure is based on the relations between the source sentence words in a dependency tree. In order to obtain the dependency relations, a dependency tree is extracted from a constituency
Well it obviously is not.

| bag-of-words | not is it obviously well |
| Features: context | -1_well +1_obviously +2_is |
| dependency | dep_parent_nsubj_is |

Example A.1: Representation of the source word *it* by the different features.

A.12.2 Word Representation

In this work, two methods to represent the words in the features were compared: word IDs and word vectors.

**Word IDs** When representing words by word IDs, the source vocabulary size $V_{source}$ is used as the dimension of the feature space, a word’s ID in the vocabulary as a feature. The feature is set to 1 if it is used in the example. All other features are set to 0. For accommodating the context features (context), the size of the features space is extended such that $V_{context} = c \times V_{source}$ where $c$ equals the size of the context. Each position of a word in the context hence has its own range in the feature space, and words in different context positions can be distinguished accordingly. The features representing dependency relations (dep) are included in a similar fashion. Again, a new feature space is defined as $V_{dep} = d \times V_{source}$ where $d$ equals the amount of all dependency relations, where parent and child relations are counted separately. The feature types can be combined by simply concatenating the individual feature spaces. That means when all three types of features are used the size of the feature space amounts to $V_{source} + V_{context} + V_{dep}$.

It is obvious that with this strategy for design the feature space grows relatively big, possibly leading to data sparseness problems. In order to reduce dimensions, the representation via word vectors is used as a more appropriate measure.

**Word Vectors** The word vectors for word representation are generated using word2vec with the number of dimensions set to 100. That means each word is represented by a 100-dimensional vector. However, it is not straightforward how multiple words should be expressed in this representation, so that the representation by word vectors is not applied for the bag-of-words features, but only for the context and dependency features. In case of the vector representation of the context features (contextVec), each position in the context words receives its own range in the feature space. Hence, the size of the feature space equals to $V_{contextVec} = c \times dim$, where $c$ is the context size and $dim$ the dimension of the vector representation. This reduces the dimension significantly compared to $V_{context}$ used in the word ID-based representation. The feature space for dependency relations using word vectors (depVec) equals to $V_{depVec} = d \times dim$ with $d$ being the inventory of dependency relations. Compared to $V_{dep}$, again a huge reduction can be achieved. In addition to the depVec feature, further variants of the dependency feature are compared as followings.

**parentDepVec**

For this feature, only the dependency relation to the parent word is represented in vector representation.
**parentWordVec**

This feature consists of the vector representation of the parent word and an additional binary feature that is 1 if the parent word is the root of the dependency tree.

**parentWordVec+DepRel**

In addition to the **parentWordVec** feature, the dependency relation to the parent word is encoded as a vector.

As for the word-based features, word vector features can be combined by concatenation of feature spaces.
Paying Attention to Multi-Word Expressions in Neural Machine Translation

Matīss Rikters matiss@lielakeda.lv
Faculty of Computing, University of Latvia

Ondřej Bojar bojar@ufal.mff.cuni.cz
Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Abstract
Processing of multi-word expressions (MWEs) is a known problem for any natural language processing task. Even neural machine translation (NMT) struggles to overcome it. This paper presents results of experiments on investigating NMT attention allocation to the MWEs and improving automated translation of sentences that contain MWEs in English → Latvian and English → Czech NMT systems. Two improvement strategies were explored—(1) bilingual pairs of automatically extracted MWE candidates were added to the parallel corpus used to train the NMT system, and (2) full sentences containing the automatically extracted MWE candidates were added to the parallel corpus. Both approaches allowed to increase automated evaluation results. The best result—0.99 BLEU point increase—has been reached with the first approach, while with the second approach minimal improvements achieved. We also provide open-source software and tools used for MWE extraction and alignment inspection.

1 Introduction
It is well known that neural machine translation (NMT) has defined the new state of the art in the last few years (Sennrich et al., 2016a; Wu et al., 2016), but the many specific aspects of NMT outputs are not yet explored. One of which is translation of multi-word units or multi-word expressions (MWEs). MWEs are defined by Baldwin and Kim (2010) as “lexical items that: (a) can be decomposed into multiple lexemes; and (b) display lexical, syntactic, semantic, pragmatic and/or statistical idiomaticity”. MWEs have been a challenge for statistical machine translation (SMT). Even if standard phrase-based models can copy MWEs verbatim, they suffer in grammaticality. NMT, on the other hand, may struggle in memorizing and reproducing MWEs, because it represents the whole sentence in a high-dimensional vector, which can lose the specific meanings of the MWEs even in the more fine-grained attention model (Bahdanau et al., 2015), because MWEs may not appear frequently enough in the training data.

The goal of this research is to examine how MWEs are treated by NMT systems, compare that with related work in SMT, and find ways to improve MWE translation in NMT. We aimed to compare how NMT pays attention to MWEs during translation, using a test set particularly targeted at handling of MWEs, and if that can be improved by populating the training data for the NMT systems with parallel corpora of MWEs.

The objective was to obtain a comparison of how NMT with regular training data and NMT with synthetic MWE data pays attention to MWEs during the translation process as well
as to improve the final NMT output. To achieve this objective, it needed to be broken down into smaller sub-objectives:

- Train baseline NMT systems,
- Extract parallel MWE corpora from the training data,
- Train the NMT systems with synthetic MWE data, and
- Inspect alignments produced by the NMT.

The structure of this paper is as follows: Section 2 summarizes related work in translating MWEs with SMT and NMT. Section 3 describes the architecture of the baseline system and outlines the process of extracting parallel MWE corpora from the training data. Section 4 provides the experiment setup and results. Finally, conclusions and aims for further directions of work are summarized in Section 5.

2 Related Work

There have been several experiments with incorporating separate processing of MWEs in rule-based (Dekone et al., 2008) and statistical machine translation tasks (Bouamor et al., 2012; Skadina, 2016). However, there is little literature about similar integrations in NMT workflows so far.

Skadina (2016) performed a series of experiments on extracting MWE candidates and integrating them in SMT. The author experimented with several different methods for both the extraction of MWEs and integration of the extracted MWEs into the MT system. In terms of automatic MT evaluation, this allowed to achieve an increase of 0.5 BLEU points (Papineni et al., 2002) for an English→Latvian SMT system.

Tang et al. (2016) introduce an NMT approach that uses a stored phrase memory in symbolic form. The main difference from traditional NMT is tagging candidate phrases in the representation of the source sentence and forcing the decoder to generate multiple words all at once for the target phrase. Although they do mention MWEs, no identification or extraction of MWEs is performed and the phrases they mainly focus on are dates, names, numbers, locations, and organizations, that are collected from multiple dictionaries. For Chinese→English they report a 3.45 BLEU point increase over baseline NMT.

Cohn et al. (2016) describe an extension of the traditional attentional NMT model with the inclusion of structural biases from word-based alignment models, such as positional bias, Markov conditioning, fertility and agreement over translation directions. They perform experiments translating between English, Romanian, Estonian, Russian and Chinese and analyze the attention matrices of the output translations produced by running experiments using the different biases. Specific experiments targeting MWEs are not performed, but they do point out that using fertility, especially global fertility, can be useful for dealing with multi-word expressions. They report a statistically significant improvement of BLEU scores in almost all involved language pairs.

Chen et al. (2016) use a similar approach as we do. Their “bootstrapping” automatically extracts smaller parts of training segment pairs and adds them to the training data for NMT. The main difference is that they rely on automatic word alignment and punctuation in the sentence to identify matching sub-segments.

3 Data Preparation and Systems Used

To measure changes introduced by adding synthetic MWE data to the training corpora, first, a baseline NMT system was trained for each language pair. The experiments were conducted on English→Czech and English→Latvian translation directions.
3.1 Baseline NMT System
To be able to compare the results with other MT systems, training and development corpora were used from the WMT shared tasks: data from the News Translation Task\(^1\) for English→Latvian and data from the Neural MT Training Task\(^2\) (Bojar et al., 2017) for English→Czech. The English→Czech data consists of about 49 million parallel sentence pairs and the English→Latvian of about 4.5 million. The development corpora consist of 2003 sentences for English→Latvian and 6000 for English→Czech.

Neural Monkey (Helcl and Libovický, 2017), an open-source tool for sequence learning, was used to train the baseline NMT systems. Using the configuration provided by the WMT Neural MT Training Task organizers, the baseline reached 11.29 BLEU points for English→Latvian after having seen 23 million sentences in about 5 days and 13.71 BLEU points for English→Czech after having seen 18 million sentences in about 7 days.

3.2 Extraction of Parallel MWEs
To extract MWEs, the corpora were first tagged with morphological taggers: UDPipe (Ramisch, 2012) for English and Czech, LV Tagger (Paikens et al., 2013) for Latvian. After that, the tagged corpora were processed with the Multi-word Expressions toolkit (Ramisch, 2012), and finally aligned with the MPAAligner (Pinnis, 2013), intermittently pre-processing and post-processing with a set of custom tools. To extract MWEs from the corpora with the MWE Toolkit, patterns were required for each of the involved languages. Patterns from Skadina (2016) were used for Latvian (210 patterns) and English (57 patterns) languages and patterns from Majchráková et al. (2012) and Pecina (2008) for Czech (23 patterns).

This workflow allowed to extract a parallel corpus of about 400,000 multi-word expressions for English→Czech and about 60,000 for English→Latvian. For an extension of this experiment, all sentences containing these MWEs were also extracted from the training corpus, serving as a separate parallel corpus.

4 Experiments
We experiment with two forms of the presentation of MWEs to the NMT system: (1) we add only the parallel MWEs themselves, each pair forming a new “sentence pair” in the parallel corpus, and (2) we use full sentences containing the MWEs. We denote the approaches “MWE phrases” and “MWE sents.” in the following.

4.1 Training Corpus Layout
In both cases, we use the same corpus training corpus layout: we mix the baseline parallel corpus with synthetic data so that MWEs get more exposure to the neural network in training.
Figure 1 and Figure 2 illustrate how the training data was divided into portions. The block $1xMWE$ corresponds to the full set of extracted MWEs (400K for En→Cs, 60K for En→Lv) and $2xMWE$ corresponds to two copies of the set (800K for En→Cs, 120K for En→Lv). For En→Lv the full corpus was used. For En→Cs we used only the first 15M sentences to be able to train multiple epochs on the available hardware. The MWEs get repeated five times in both language pairs. By doing this, the En→Cs data set was reduced from 49M to 17M and the En→Lv data set increased to 4.8M parallel sentences for one epoch of training.

While the experiments were running, early stopping of the training was executed and snapshots of the models for evaluation were taken in stages where the models already were starting to converge. For En→Lv this was after the networks had been trained on 25M sentences (i.e. 5.2 epochs of the mixed corpus), for En→Cs 27M sentences (i.e. 1.6 epochs).

Neural Monkey does not shuffle the training corpus between epochs. This is not a problem if the corpus is properly shuffled and the number of epochs is not very large compared to the size of the epochs. We shuffled only the baseline corpus and the interleaved it with (shuffled) sections for MWEs. This worked well when MWEs were provided in full sentences, but not with MWEs presented as expressions. In the latter case, the NMT started to produce only very short output, losing very much of its performance. We, therefore, shuffle the whole composed corpus for the “MWE phrases” runs, effectively discarding the interleaved composition of the training data.

4.2 Results

Table 1 shows the results for both approaches and both language pairs. Due to hardware constraints, we were not able to try out both approaches on both language pairs.

We evaluate all setups with BLEU (Papineni et al., 2002) on the full development set (distinct from the training set), as shown in the column “Dev”, and on a subset of 611 (En→Lv) and 112 (En→Cs) sentences containing the identified MWEs (column “MWE”).

![Table 1: Experiment results.](image)

![Figure 3: Automatic evaluation progression of En→Cs experiments on validation data. Orange – baseline; blue — baseline with added MWEs.](image)

![Figure 4: Automatic evaluation progression of En→Lv experiments on validation data. Orange – baseline; purple — baseline with added MWE sentences.](image)
Quality Translation 21
D2.3: Final Report: Morphologically Rich Languages

Figure 5: Differences in translation between baseline and improved NMT system. Improving n-grams are highlighted in green and worsening n-grams — in red.

The improvement is more apparent when evaluated on the dedicated devset of sentences containing multi-word expressions. The improvement for Latvian is even 0.99 BLEU, but arguably, the baseline performance of our system is not very high. Also, more runs should be carried out for a full confidence, but this was unfortunately out of our limits on computing resources.

4.3 Manual Inspection

To find out whether changes in the results are due to the synthetic MWE corpora added, a subset of output sentences from the ones containing MWEs were selected for closer examination. For this task, we used the iBLEU (Madnani, 2011) tool.

In Figure 5, an improvement in the modified NMT translation is visible due to the treatment of the compound nominal “city bus” as a single expression. It seems that the baseline system translates “city” into “městské” and “bus” into “autobuse” individually, resulting in the wrong form of “city” in Czech (a noun used instead of an adjective). On the other hand, the improved NMT translates “city” into “městském” just like the target human translation. Attention alignments will be examined in the following section.

Figure 6 shows an example where the improved NMT scores higher in BLEU points and translates the MWE closer to the human, but loses a part of it in the process. While translating the noun phrase “electronic wall map” the improved system generates a closer match to the human translation “elektronické mapě”, it does not translate the word “wall” that was translated into “stěny” by the baseline system. Upon closer inspection, we discovered that this error was caused by the MWE extractor and aligner because the identified English phrase “electronic wall map” was aligned to an identified Czech phrase “elektronické mapě” and the whole phrase...
It should be noted that this is not the first time that Facebook has been actively involved in determining what network users see in their news feeds.

Baseline: Jaatzimė, ka šis nav pirmajā reizē, kad Facebook ir aktīvi iesaistīta, nosakot to, ko ķīlā izmanto viņu zīmu pārraides.

Improved NMT: Ir jaatzimē, ka šīs ir pirmā reize, kad Facebook ir aktīvi iesaistījusies, nosakot to, ko ķīlā lietojā dara viņu zīmu formātā.

Reference: Jāteic, ka šī nav pirmā reize, kad Facebook aktīvi iesaistās, nosakot, ko ķīlā lietojā redz savās jaunumu plūsmās.

Figure 7: Differences in translation between baseline and improved NMT system. Improving n-grams are highlighted in green and worsening n-grams — in red.

Figure 9: Fragment of soft alignments of the example sentence from the baseline NMT system.

Figure 10: Fragment of soft alignments of the example sentence from the improved NMT system.

“nāstēne elektroniecke mapē” was not identified by the MWE extractor at all.

Figure 7 illustrates translations of an example sentence by the En→Lv NMT systems. The MWE, in this case, is “network users” that is translated as “tīkla lietojā” by the modified system and completely mistranslated by the baseline.

4.4 Alignment Inspection

For inspecting the NMT attention alignments, we developed a tool (Rikters et al., 2017) that takes data produced by Neural Monkey—a 3D array (tensor) filled with the alignment probabilities together with source and target subword units (Sennrich et al., 2016b) or byte pair encodings (BPEs)—as input and produces a soft alignment matrix (Figure 8) of the subword units that highlights all units, that get attention when translating a specific subword unit. The tool includes a web version that was adapted from Nematus (Sennrich et al., 2017) utilities and slightly modified. It allows to output the soft alignments in a different perspective, as connections between BPEs as visible in Figure 9 and Figure 10.
Source: Just like in a city bus or a tram.
Baseline: Jako ve městském autobus nebo tramvaji.
Improved NMT: Jen jako v městském autobus nebo tramvaji.
Reference: Stejně jako v městském autobus či tramvaji.

Figure 11: Soft alignment example visualizations from translating an English sentence into Czech from the baseline (top, hypothesis 1) and improved (bottom, hypothesis 2) NMT systems.

Figure 8: Example of a soft alignment matrix.
In these examples, the attention state of the previously mentioned MWE from En→Lv translations (“network users”) is visible. The alignment inspection tool allows to see that the baseline NMT in Figure 9 has multiple faded alignment lines for both words “network” and “users”, which outlines that the neural network is unsure and looking all around for traces to the correct translation. However, in Figure 10, it is visible that both these words have strong alignment lines to the words “tīkla lietotāji”, that were also identified by the MWE Toolkit as an MWE candidate.

Figure 11 shows one of the previously mentioned En→Cs translation examples. Here it is clear that in the baseline alignment no attention goes to the word “mēstē” or the subword units “autobus@” and “se” when translating “city”. In the modified version, on the other hand, some attention from “city” goes into all closely related subword units: “mēstē@”, “skēm”, “autobus@”, and “se”. It is also visible that in this example, the translation of “bus” gets attention from not only “autobus@” and “se” but also the ending subword unit of “city”, i.e. the token “skēm”.

5 Conclusion

In this paper, we described the first experiments with handling multi-word expressions in neural machine translation systems. Details on identifying and extracting MWEs from parallel corpora, as well as aligning them and building corpora of parallel MWEs were provided. We explored two methods of integrating MWEs in training data for NMT and examined the output translations of the trained NMT systems with custom built tools for alignment inspection.

In addition to the methods described in this paper, we also released open-source scripts for a complete workflow of identifying, extracting and integrating MWEs into the NMT training and translation workflow.

While the experiments did not show outstanding improvements on the general development data set, an increase of 0.99 BLEU was observed when using an MWE specific test data set. Manual inspection of the output translations confirmed that translations of specific MWEs were improving after populating the training data with synthetic MWE data.

As the next steps, we plan (1) to analyze the obtained results of our experiments in more detail through the help of a larger scale manual human evaluation of the NMT output and (2) to continue experiments to find best ways how to treat different categories of MWEs, i.e. idioms.

Acknowledgement

This study was supported in parts by the grants H2020-ICT-2014-1-645442 (QT21), the ICT COST Action IC1207 ParseME: Parsing and multi-word expressions. Towards linguistic precision and computational efficiency in natural language processing, and Charles University Research Programme “Progres” Q18 – Social Sciences: From Multidisciplinarity to Interdisciplinarity.

References


Abstract

Translating into morphologically rich languages is difficult. Although the coverage of lemmas may be reasonable, many morphological variants cannot be learned from the training data. We present a statistical translation system that is able to produce these inflected word forms. Different from most previous work, we do not separate morphological prediction from lexical choice into two consecutive steps. Our approach is novel in that it is integrated in decoding and takes advantage of context information from both the source language and the target language sides.

1 Introduction

Morphologically rich languages exhibit a large amount of inflected word surface forms for most lemmas, which poses difficulties to current statistical machine translation (SMT) technology. SMT systems, such as phrase-based translation (PBT) engines (Koehn et al., 2003), are trained on parallel corpora and can learn the vocabulary that is observed in the data. After training, the decoder can output words which have been seen on the target side of the corpus, but no unseen words.

Sparsity of morphological variants leads to many linguistically valid morphological word forms remaining unseen in practical scenarios. This is a substantial issue under low-resource conditions, but the problem persists even with larger amounts of parallel training data. When translating into the morphologically rich language, the system fails at producing the unseen morphological variants, leading to major translation errors.

Consider the Czech example in Table 1. A small parallel corpus of 50K English-Czech sentences contains only a single variant of the morphological forms of the Czech lemma “čěška” (plural of English: “kneecap”), out of seven syntactically valid cases. The situation improves as we add in more training data (500K/5M/50M), but we can generally not expect the SMT system to learn all variants of each known lemma. In Czech, the number of possible variants is even larger for other word categories such as verbs or adjectives. Adjectives, for instance, have different suffixes depending on case, number, and gender of the governing noun.

In this paper, we propose an extension to phrase-based SMT that allows the decoder to produce any morphological variant of all known lemmas. We design techniques for generating and scoring unseen morphological variants fully integrated into phrase-based search, with the decoder being able to choose freely amongst all possible morphological variants. Empirically, we observe considerable gains in translation quality especially under medium- to low-resource conditions.

2 Related Work

Translation into morphologically rich languages is often tackled through “two-step”, i.e., separate modules for morphological prediction and generation (Toutanova et al., 2008; Bojar and Kos, 2010;
Fraser et al., 2012; Burlot et al., 2016). An important problem is that lexical choice (of the lemma) is carried out in a separate step from morphological prediction.

Factored machine translation with separate translation and generation models represents a different approach, operating with a single-step search. However, too many options in decoding cause a blow-up of the search space; and useful information is dropped when modeling source_word—target_lemma and target_lemma—target_word separately.

Word forms not seen in parallel data are sometimes still available in monolingual data. Back-translation (Bojar and Tarnopolsky, 2011) takes advantage of this. The monolingual target language data is lemmatized, automatically translated to the source language, and the translations are aligned with the original, inflected target corpus to produce supplementary training data. Disadvantages are both the computational expense and that the back-translated text may contain errors.

Previous work on synthetic phrases by Chahunneau et al. (2013) is most similar to our work. They commit to generation of a single candidate inflection of a lemma prior to decoding, chosen only based on a hierarchical rule and source-side information, a significant limitation. We instead consider all morphological variants, and we are able to use dynamically-generated target-side context in choosing the correct variant, which is critical for capturing phenomena such as target-side verb-subject agreement, or the agreement between a preposition marking case and the case on the noun it marks.

3 Generating Unseen Morphological Variants

We investigate an approach based on synthesized morphological variants. A morphological generation tool is utilized to synthesize all valid morphological forms from target-side lemmas. The phrase table is then augmented with additional entries to provide complete coverage.

We process single target-word entries from the baseline phrase table and feed the lemmatized target word into the morphological generation tool. If its output contains morphological forms that are not known as translations of the source side of the phrase, we add these morphological variants as new translation options. We consider two settings:

<table>
<thead>
<tr>
<th>feature type</th>
<th>configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>source indicator</td>
<td>l, l, t, l+p, t, r+p</td>
</tr>
<tr>
<td>source internal source context</td>
<td>l (3,3), t (5,5)</td>
</tr>
<tr>
<td>target indicator</td>
<td>l, t</td>
</tr>
<tr>
<td>target internal target context</td>
<td>l, t (2), t (2)</td>
</tr>
</tbody>
</table>

Table 2: Feature templates for the discriminative classifier: l (lemma), t (morphosyntactic tag), r (syntactic role), p (lemma of dependency parent). Numbers in parentheses indicate context size.

(1) word, where morphological word forms are generated from phrase table entries of length 1 on both source and target side, and (2) mtu (for “minimal translation unit”), where the phrase source side can have arbitrary length.

Morphological generation for Czech, for instance, can be performed with the MorphoDiTa toolkit (Straková et al., 2014), which we will use in our experiments. MorphoDiTa knows a dictionary of most Czech lemmas and can generate all their morphological variants (Hajièi, 2004).

When not restricted, the morphological generator also produces forms which do not match in number, tense, degree of comparison, or even negation. This may be undesirable and we therefore define a tag template. The tag template prevents the generation of some forms of the given Czech lemma. The template only allows freedom in the following morphological categories: gender, case, person, possessor’s number, and possessor’s gender. All other attributes must match the original Czech word form. The morphosyntax of the English source is not used to impose further constraints. We will mark this configuration with an asterisk (•) in our experiments.

4 Scoring Unseen Morphological Variants

Assigning dependable model scores to synthesized morphological forms is a primary challenge. During decoding, the artificially added phrase table entries compete with baseline phrases that had been directly extracted from the parallel training data. The correct choice has to be determined in search based on model scores.

A phrase-based model with linguistically motivated factors (Koehn and Hoang, 2007) enables us to achieve better generalization capabilities when translating into a morphologically rich language.
Table 3: English—Czech experimental results using 50K training sentence pairs.

<table>
<thead>
<tr>
<th>System</th>
<th>2014 BLEU</th>
<th>2015 BLEU</th>
<th>2016 BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline 50K</td>
<td>12.4</td>
<td>10.8</td>
<td>11.8</td>
</tr>
<tr>
<td>+ morph-vw-50K</td>
<td>12.2</td>
<td>10.6</td>
<td>11.8</td>
</tr>
<tr>
<td>+ synthetic (word)</td>
<td>13.4</td>
<td>11.3</td>
<td>12.5</td>
</tr>
<tr>
<td>+ morph-vw-50K</td>
<td>13.4</td>
<td>11.4</td>
<td>12.2</td>
</tr>
<tr>
<td>+ synthetic (word+)</td>
<td>13.3</td>
<td>11.3</td>
<td>12.5</td>
</tr>
<tr>
<td>+ morph-vw-50K</td>
<td>13.3</td>
<td>11.3</td>
<td>12.7</td>
</tr>
<tr>
<td>+ synthetic (mtu)</td>
<td>13.5</td>
<td>11.5</td>
<td>12.7</td>
</tr>
<tr>
<td>+ morph-vw-50K</td>
<td>13.4</td>
<td>11.4</td>
<td>12.7</td>
</tr>
<tr>
<td>+ synthetic (mtu+)</td>
<td>13.4</td>
<td>11.3</td>
<td>12.9</td>
</tr>
<tr>
<td>+ morph-vw-50K</td>
<td>13.5</td>
<td>11.5</td>
<td>13.1</td>
</tr>
</tbody>
</table>

Table 4: English—Czech experimental results using 500K training sentence pairs.

<table>
<thead>
<tr>
<th>System</th>
<th>2014 BLEU</th>
<th>2015 BLEU</th>
<th>2016 BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline 500K</td>
<td>17.7</td>
<td>14.4</td>
<td>16.1</td>
</tr>
<tr>
<td>+ morph-vw-500K</td>
<td>17.6</td>
<td>14.4</td>
<td>16.5</td>
</tr>
<tr>
<td>+ synthetic (word)</td>
<td>18.1</td>
<td>14.7</td>
<td>16.4</td>
</tr>
<tr>
<td>+ morph-vw-500K</td>
<td>18.4</td>
<td>15.2</td>
<td>17.3</td>
</tr>
<tr>
<td>+ synthetic (word+)</td>
<td>18.0</td>
<td>14.8</td>
<td>16.6</td>
</tr>
<tr>
<td>+ morph-vw-500K</td>
<td>18.2</td>
<td>14.9</td>
<td>17.0</td>
</tr>
<tr>
<td>+ synthetic (mtu)</td>
<td>18.1</td>
<td>14.8</td>
<td>16.6</td>
</tr>
<tr>
<td>+ morph-vw-500K</td>
<td>18.5</td>
<td>15.3</td>
<td>17.3</td>
</tr>
<tr>
<td>+ synthetic (mtu+)</td>
<td>18.3</td>
<td>15.0</td>
<td>16.9</td>
</tr>
<tr>
<td>+ morph-vw-500K</td>
<td>18.6</td>
<td>15.4</td>
<td>17.4</td>
</tr>
</tbody>
</table>

Table 5: English—Czech experimental results using 5M training sentence pairs.

<table>
<thead>
<tr>
<th>System</th>
<th>2014 BLEU</th>
<th>2015 BLEU</th>
<th>2016 BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline 5M</td>
<td>20.8</td>
<td>16.8</td>
<td>18.9</td>
</tr>
<tr>
<td>+ morph-vw-5M</td>
<td>20.9</td>
<td>16.8</td>
<td>19.0</td>
</tr>
<tr>
<td>+ synthetic (word)</td>
<td>20.9</td>
<td>17.0</td>
<td>19.0</td>
</tr>
<tr>
<td>+ morph-vw-5M</td>
<td>21.1</td>
<td>17.0</td>
<td>19.0</td>
</tr>
<tr>
<td>+ synthetic (word+)</td>
<td>20.7</td>
<td>16.8</td>
<td>19.0</td>
</tr>
<tr>
<td>+ morph-vw-5M</td>
<td>20.4</td>
<td>16.4</td>
<td>18.7</td>
</tr>
<tr>
<td>+ synthetic (mtu)</td>
<td>20.6</td>
<td>17.2</td>
<td>19.0</td>
</tr>
<tr>
<td>+ morph-vw-5M</td>
<td>21.0</td>
<td>16.9</td>
<td>19.0</td>
</tr>
<tr>
<td>+ synthetic (mtu+)</td>
<td>20.8</td>
<td>17.1</td>
<td>19.1</td>
</tr>
<tr>
<td>+ morph-vw-5M</td>
<td>20.9</td>
<td>16.8</td>
<td>19.0</td>
</tr>
</tbody>
</table>

Table 6: English—Czech experimental results using 50M training sentence pairs.

<table>
<thead>
<tr>
<th>System</th>
<th>2014 BLEU</th>
<th>2015 BLEU</th>
<th>2016 BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline 50M</td>
<td>22.3</td>
<td>18.1</td>
<td>20.5</td>
</tr>
<tr>
<td>+ morph-vw-50M</td>
<td>22.7</td>
<td>18.2</td>
<td>20.7</td>
</tr>
<tr>
<td>+ synthetic (word)</td>
<td>22.3</td>
<td>18.2</td>
<td>20.5</td>
</tr>
<tr>
<td>+ morph-vw-50M</td>
<td>22.3</td>
<td>18.1</td>
<td>20.4</td>
</tr>
<tr>
<td>+ synthetic (word+)</td>
<td>22.2</td>
<td>18.1</td>
<td>20.6</td>
</tr>
<tr>
<td>+ morph-vw-50M</td>
<td>22.5</td>
<td>18.1</td>
<td>20.6</td>
</tr>
<tr>
<td>+ synthetic (mtu)</td>
<td>22.3</td>
<td>18.1</td>
<td>20.5</td>
</tr>
<tr>
<td>+ morph-vw-50M</td>
<td>22.7</td>
<td>18.3</td>
<td>20.8</td>
</tr>
<tr>
<td>+ synthetic (mtu+)</td>
<td>22.3</td>
<td>17.9</td>
<td>20.3</td>
</tr>
<tr>
<td>+ morph-vw-50M</td>
<td>22.4</td>
<td>18.1</td>
<td>20.5</td>
</tr>
</tbody>
</table>

In our baseline systems, we already draw on lemmas and morphosyntactic tags as factors on the target side, in addition to word surface forms. The additional target-side factors allow us to integrate features that independently model word sense (in terms of the lemma) and morphological attributes (in terms of the morphosyntactic tag). All our translation engines (cf. Section 5) incorporate n-gram LMs over lemmas and over morphosyntactic tags, and an operation sequence model (OSM) (Durrani et al., 2013) with lemmas on the target side. These models counteract sparsity, and where models over surface forms fail for unseen variants, they still assign scores which are based on reliable probability estimates.

When enhancing a system with synthesized phrase table entries, we add further features. Since the usual phrase translation and lexical translation log-probabilities over surface forms cannot be estimated for unseen morphological variants, but all new variants are generated from existing lemmas, we utilize the corresponding log-probabilities over target lemmas. Those can be extracted from the parallel training data and added to the synthesized entries. For baseline phrase table entries, we retain their four baseline phrase translation and lexical translation features, meaning that features over target lemmas score synthesized entries and features over surface forms score baseline entries. The features have separate weights in the model combination. Furthermore, a binary indicator distinguishes baseline phrases from synthesized phrases.

The final key to our approach is using a discriminative classifier (morph-vw, Vowpal Wabbit for Morphology) which can take context from both the source side and the target side into account, as in (Tamchyna et al., 2016). We design feature templates for the classifier that generalize to unseen morphological variants, as listed in Table 2. “Indicator” features are concatenations of words inside...
5 Empirical Evaluation

For an empirical evaluation of our technique, we build baseline phrase-based SMT engines using Moses (Koehn et al., 2007). We then enrich these baselines with linguistically motivated morphological variants that are unseen in the parallel training data, and we augment the model with the discriminative classifier to guide morphological selection during decoding. Different flavors of synthetic morphological variants are compared, each either combined with the discriminative classifier or standalone.

We choose English—Czech as a task that is representative for machine translation from a morphologically underspecified language into a morphologically rich language.

5.1 Experimental Setup

We train a phrase-based translation system with three factors on the target side of the translation model (but no separate generation model). The target factors are the word surface form, lemma, and a morphosyntactic tag. We use the Czech positional tagset (Hajič and Hladká, 1998) which fully describes the word’s morphological attributes. On the source side we use only surface forms, except for the discriminative classifier, which includes the features as shown in Table 2.

We employ corpora that have been provided for the English—Czech News translation shared task at WMT16 (Bojar et al., 2016b), including the CzEng parallel corpus (Bojar et al., 2016a). Word alignments are created using fast_align (Dyer et al., 2013) and symmetrized. We extract phrases up to a maximum length of 7. The phrase table is pre-pruned by applying a minimum score threshold of 0.0001 on the source-to-target phrase translation probability, and the decoder loads a maximum of 100 best translation options per distinct source side. We use cube pruning in decoding. Pop limit and stack limit for cube pruning are set to 1000 for tuning and to 5000 for testing. The distortion limit is 6. Weights are tuned on newstest2013 with k-best MIRA (Cherry and Foster, 2012) over 200-best lists for 25 iterations. Translation quality is measured in BLEU (Papineni et al., 2002) on three different test sets, newstest2014, newstest2015, and newstest2016.3

Our training data amounts to around 50 million bilingual sentences overall, but we conduct sets of experiments with systems trained using different fractions of this data (50K, 500K, 5M, 50M). Whereas English—Czech has good coverage in terms of training corpora, we simulate low- and medium-resource conditions for the purpose of drawing more general conclusions. Irrespective of this, we utilize the same large LMs in all setups, assuming that proper amounts of target language monolingual data can often be gathered, even when parallel data is scarce. All other models (including the morph-vw) are trained using only the fraction of data as chosen for the respective set of experiments, and synthesized phrase table entries with generated morphological variants are produced individually for each baseline phrase table.

3We evaluate case-sensitive with mteval-v13a.pl -c, comparing post-processed hypotheses against the raw reference.
input: now, six in 10 Republicans have a favorable view of Donald Trump.
baseline: ted', šest v 10 republikáni mají pˇríznivý výhled Donald Trump.
+ synthetic (mtu) + morph-vw: ted', šest do deseti republikáni má pˇríznivý názor na Donalda Trumpu.

Figure 2: Example outputs of 500K system variants. Each translation has a corresponding gloss in italics. Errors are marked in bold. Synthetic phrase translations are underlined.

5.2 Experimental Results and Analysis

Translation results are reported in Tables 3 to 6. Our method is effective at improving BLEU especially in the low- and medium-resource settings, but shows only slight gains in the 5M and 50M scenarios. Overall, mtu leads to better results than word. When we also add translations to phrases with multiple input words, we give the system more leeway in phrasal segmentation and our synthetic phrases can perhaps be applied more easily.

In the 50K and 500K settings, we obtain considerable improvements even without using the discriminative model. This suggests that our scoring scheme based on lemmas is indeed effective for the synthetic phrase pairs. Additionally, model features such as the OSM with target-side lemmas as well as the LMs over lemmas and over morphosyntactic tags seem to cope with the synthetic word forms reasonably well. However, when we do use the classifier, we obtain a small but consistent further improvement.

Figure 1 visualizes the BLEU scores achieved under the four training resource conditions with the baseline system and with the system extended via synthesized morphological word forms (in the mtu variant) plus the discriminative classifier, respectively.

In order to better understand why the improvements fall off as we increase training data size, we measure target-side out-of-vocabulary (OOV) rates of the various settings. Our aim is to quantify the potential improvement that our method can bring. Table 7 shows the statistics: at 50K, the baseline OOV rate is nearly 17% and our technique successfully reduces it to less than 10%. The relative reduction of the OOV rate is quite steady as training data size increases.

Figure 2 illustrates the effect of our technique in a medium-size setting (500K). The baseline system is forced to use the incorrect nominative case due to the lack of required surface forms. Our method provides these inflections (“republikánu”, “Trumpu”) and produces a mostly grammatical translation (but is still unable to correctly translate the preposition “in”).

6 Conclusion

We have studied the important problem of modeling all morphological variants of our SMT system’s vocabulary. We showed that we can augment our system’s vocabulary with the missing variants and that we can effectively score these variants using a discriminative lexicon utilizing both source and target context. We have shown that this leads to substantial BLEU score improvements, particularly on small to medium resource translation tasks. Given the limited training data available for translation to many morphologically rich languages, our approach is widely applicable.
Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements № 644402 (HinL) and № 645452 (QT21), from the European Research Council (ERC) under grant agreement № 640550, and from the DFG grant Models of Morphosyntax for Statistical Machine Translation (Phase Two). This work has been using language resources and tools developed and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

References


Abstract

Linguistic resources such as part-of-speech (POS) tags have been extensively used in statistical machine translation (SMT) frameworks and have yielded better performances. However, usage of such linguistic annotations in neural machine translation (NMT) systems has been left under-explored.

In this work, we show that multi-task learning is a successful and easy approach to introduce an additional knowledge into an end-to-end neural attentional model. By jointly training several natural language processing (NLP) tasks in one system, we are able to leverage common information and improve the performance of the individual task.

We analyze the impact of three design decisions in multi-task learning: the tasks used in training, the training schedule, and the degree of parameter sharing across the tasks, which is defined by the network architecture. The experiments are conducted for a German to English translation task. As additional linguistic resources, we exploit POS information and named-entities (NE). Experiments show that the translation quality can be improved by up to 1.5 BLEU points under the low-resource condition. The performance of the POS tagger is also improved using the multi-task learning scheme.

1 Introduction

Recently, there has been a dramatic change in the state-of-the-art techniques for machine translation (MT). In a traditional method, often the best performance is achieved by using a complicated combination of several statistical models, which are individually trained. For example, POS information was shown to be very helpful to model word re-ordering between languages, as shown in Niehues and Kolss (2009). While the recent development of end-to-end trained neural models (Bahdanau et al., 2014) showed significant gains over traditional approaches, they are often trained only on the parallel data in an end-to-end fashion. In most cases, therefore, they do not facilitate other knowledge sources.

When parallel data is sparse, exploiting other knowledge sources can be crucial for performance. Two techniques to integrate the additional resources are well studied. In one technique, we train a tool on the additional resources (e.g. POS tagger) and then annotate the parallel data using this tool. This technique has been applied extensively in SMT systems (e.g. Niehues and Kolss (2009)) as well as in some NMT systems (e.g. Sennrich and Haddow (2016)). The second technique would be to use the annotated data directly to train the model.

The goal of this work is to integrate the additional linguistic resources directly into neural models, in order to achieve better performance. To do so, we build a multi-task model and train several NLP tasks jointly.

We use an attention-based sequence-to-sequence model for all tasks. Experiments show that we are able to improve the performance on the German to English machine translation task measured in BLEU, BEER and CharactER. Furthermore, we analyze three important decisions when designing multi-task models. First, we investigated the influence of secondary tasks. Also, we analyze the influence of training schedule, e.g. whether we need to adjust it in order to get the best performance on the target task. And finally,
we evaluated the amount of parameter sharing enforced by different model architectures.

The main contributions of this paper are (1) that we show multi-task learning is possible within attention-based sequence-to-sequence models, which are state-of-the-art in machine translation and (2) that we analyze the influence of three main design decisions.

2 Related Work

Motivated by the success of using features learned from linguistic resources in various NLP tasks, there have been several approaches including external information into neural network-based systems.

The POS-based information has been integrated for language models in Wu et al. (2012); Niehues et al. (2016). In the neural machine translation, using additional word factors like POS-tags has shown to be beneficial (Sennrich and Haddow, 2016).

The initial approach for multi-task learning for neural networks was presented in Collobert et al. (2011). The authors used convolutional and feed forward networks for several tasks such as semantic parsing and POS tagging. This idea was extended to sequence to sequence models in Luong et al. (2015).

A special case of multi-task learning for attention based models has been explored. In multilingual machine translation, for example, the tasks are still machine translation tasks but they need to consider different language pairs. In this case, a system with an individual encoder and decoder (Firat et al., 2016b) as well as a system with a shared encoder-decoder (Ha et al., 2016; Johnson et al., 2016) has been proposed.

2.1 Attention Models

Recently, state-of-the-art performance in machine translation was significantly improved by using neural machine translation. In this approach, a recurrent neural network (RNN)-based encoder-decoder architecture is used to transform the source sentence into the target sentence.

In the encoder, an RNN is used to encode the source sentence into a fixed size of continuous space, referred to as word embeddings, is learned. An RNN model will learn the source sentence representation over these word embeddings. In a second step, the decoder is initialized by the representation of the source sentence and is then generating the target sequence one word after the other using the last generated word as input for the RNN. In order to get the output probability at each target position, a softmax layer that get the hidden state of the RNN as input is used (Sutskever et al., 2014).

The main drawback of this approach is that the whole source sentence has to be stored in a fixed-size context vector. To overcome this problem, Bahdanau et al. (2014) introduced the soft attention mechanism. Instead of only considering the last state of the encoder RNN, they use a weighted sum of all hidden states. Using these weights, the model is able to put attention on different parts of the source sentence depending on the current status of the decoder RNN. In addition, they extended the encoder RNN to a bi-directional one to be able to get information from the whole sentence at every position of the encoder RNN. A detailed description of the NMT framework can be found in Bahdanau et al. (2014).

3 Multi-task Learning

In a traditional NLP pipeline, a named entity recognition or machine translation system employ POS information by using the POS tags as additional features. For example, the system will learn that the probability of a word being a named entity is higher if the word is marked as a noun. First, a POS tagger is used to annotate the input data. Combining the statistical models used for POS tagging and named entity recognition might not be straightforward.

Recent advances in deep learning approaches, e.g. CNN or RNN-based models (Labeau and Löser K., 2015), made it straightforward to use very similar techniques throughout different NLP tasks. Therefore, there are new methods to combine the tasks. Instead of using the output of a model as input for another one, for example, we can build one model for all tasks. The model is then automatically able to learn to share as much information across the tasks as necessary.

For building a model that can learn three NLP tasks, we use the attention-based encoder-decoder model, which is a standard in state-of-the-art ma-
Machine translation systems. The two non-MT tasks can also be modeled by converting them into a translation problem. Instead of translating the source words into the target language, we translate the words into labels, either POS-tags or NE-labels.

In this work, we study several crucial design aspects when applying attention-based encoder-decoder model for a multi-task learning scenario. First, we consider different architectures of the network in order to assess how much parameter sharing is useful between the tasks. In general, sharing more information across the tasks is preferred. However, if the tasks differ from each other greatly, it might be helpful to restrict the degree of sharing. In addition, the training schedule of each task has to be addressed. While all three tasks are sharing. In Section 3.3 we address this issue.

3.1 Architecture

The general attentional encoder-decoder model consists of three main parts: the encoder \(E\), the attention model \(A\) and the decoder \(D\). Figure 1 gives an overview of this layout.

Our baseline considers the scenario where we have separate models for each task. Therefore, all three parts (encoder, attention model, and decoder) stand separately for each task. We will have nine components \(E_{MT}, E_{POS}, E_{NE}, A_{MT}, A_{POS}, A_{NE}, D_{MT}, D_{POS}, D_{NE}\) in total.

The one main design decision for a multi-task learning architecture is the degree of sharing across the tasks. Motivated by architectures proposed for multi-lingual machine translation (Dong et al., 2015; Firat et al., 2016a; Ha et al., 2016), we analyze the impact of different degrees of sharing in the output quality. When sharing more parameters between the tasks, the models are able to learn more from the training data of other tasks. If the tasks are very distant, on the other hand, it might be harmful to share the parameters.

**Shared encoder (shrd Enc)** One promising way is to share components that handle the same type of data. Since all our tasks share English as input here is the encoder.

In this architecture, we therefore use one encoder for all tasks. This is the minimal degree of sharing we consider in our experiments. A common encoder \(E_{ALL}\) is used for all tasks, but separate attention models \(A_{MT}, A_{POS}, A_{NE}\) and decoders \(D_{MT}, D_{POS}, D_{NE}\) are used.

**Shared attention (shrd Att)** The next component is the attention model which connects the encoder and decoder. While the output should be different for the addressed tasks, the type of input is the same. Therefore, it might be helpful to share more information between the models.

In a second architecture, we also share the attention model in addition to the encoder. So in this setup, we have one encoder \(E_{ALL}, one attention model A_{ALL}\) and three decoder \(D_{MT}, D_{POS}, D_{NE}\).

**Shared decoder (shrd Dec)** Finally, we explore whether it is possible to share all information across the tasks and let the model learn how to represent the different tasks. Thus, in this scheme, we aim to share the decoder partially. The only thing that is not shared is the final softmax layer.

In this architecture, the decoder RNN has to model the generation of target words as well as that of labels. Therefore, we have only one encoder \(E_{ALL}, one attention model A_{ALL}\) and one decoder \(D_{ALL}\). In the decoder, however, we have separated output layers for each task.

Figure 1 depicts which layers are shared depending on the architecture.

3.2 Training Schedule

In this section, we discuss the influence of the training schedule on the quality of the model.

Throughout our experiments we used a mini-batch size of 512 tokens. The weight updates were determined using the Adam algorithm.

The training has to be adapted to the multi-task scenario. The main decision is how to present the training examples to the training algorithm. We only consider one task in each mini-batch. Although the model structure is the same for all tasks, the models for the individual tasks have different weights. Therefore, parallelization on the GPU would be less efficient when using different tasks within one batch. In order to train our model on all tasks in parallel, we randomly shuffle the mini-batches from all tasks. This is our default training schedule. One issue in the multi-task scenario is that the data size might vary. In this case, the model will mainly concentrate on the task with the most data and not achieve the best performance on each task.
This challenge is strongly related with the problem of domain adaptation in machine translation, where a large out-of-domain data is available but only a small amount of in-domain data. For this scenario, first training on all data and then fine-tuning on the in-domain data was very successful (Lavergne et al., 2011; Cho et al., 2016). Therefore, we adapt this approach to the multi-task scenario. In this case, we first trained the model on all tasks and then continued training only on the main task. We will refer to this training schedule as adapted.

3.3 Target Length

While all tasks are modeled as a translation problem in this work, the nature of each task is largely different. One main difference between the translation task and the other two tasks is the length of the target sequence. While it is unknown in the translation task, it is known and fixed for the other two cases. During training this does not matter as the target sequence is given. For testing the system, however, this issue is crucial to address.

In our initial experiment, it was shown that the POS tagger was able to learn the correct target length in most of the cases. For some sentences, however, the estimated target length was not correct. Therefore, the prior knowledge of sequence length is used during decoding so that label sequences are generated with the correct target length. It is worth to mention that the desired length of the labels is not exactly the length of the input to the model itself. Our model uses inputs with subwords units generated by byte-pair encoding (Sennrich et al., 2016).

4 Experimental Setup

We conduct experiments using the multi-task approach on three different tasks: machine translation from German to English, German fine-grained POS tagging and German NE tagging. As briefly mentioned in Section 1, multi-task approach can be helpful when data is sparse. In order to simulate this, we deploy only German to English TED data for the translation task.

4.1 Data

For the translation task, we used 4M tokens of the WIT corpus (Cettolo et al., 2012) for German to English as training data. We used dev2010 for validation and tst2013 and tst2014 for testing, provided by the IWSLT. We only used training examples shorter than 60 words per sentence.

The POS tagger was trained on 720K tokens the Tiger Corpus (Brants et al., 2004). This corpus contains German newspaper text. Consequently, it is out-of-domain data for the machine translation task. The development and the test data are also from this corpus. The POS tag set consists of 54 tags and the fine-grained POS tags with morphological annotations has 774 labels.

Finally, we trained the German named-entity
We preprocess the parallel data by tokenizing and true-casing. In addition, we trained a byte-pair encoding (Sennrich et al., 2016) with 40K subwords on the source and target side of the TED corpus jointly. We then applied the subwords to all German and English corpora.

4.2 System Architecture

For all our experiments, we use an attentional encoder-decoder model. The baseline systems use this architecture as well. The encoder uses word embeddings of size 256 and a bidirectional LSTM (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) with 256 hidden layers for each direction. For the attention, we use a multi-layer perceptron with 512 hidden units and tanh activation function. The decoder uses conditional GRU units with 512 hidden units. The models are all trained with Adam, where we restarted the algorithm twice and early stopping is applied using log-likelihood of the concatenated validation sets from the considered tasks. For the adapted schedule, Adam is started once again when training only on the target task. The model is implemented in lamtram (Neubig, 2015).

4.3 Evaluation

The machine translation output is evaluated with BLEU (Papineni et al., 2002), BEER (Stanojevic and Sima’an, 2014) and CharacTER (Wang et al., 2016). For the POS tags, we report error rates on the small label set as well as on the large label set.

5 Results

In this section, we present the results from our experiments and analysis.

5.1 Initial experiments on the architecture

The results of the initial experiments on the machine translation tasks are shown in Table 1. The table displays the performance on the validation set and on both test sets. For all experiments, we first show the BLEU score, then the BEER score and finally the characTER.

First, we show the results of the baseline neural MT system trained on the parallel data (single task). As mentioned in the beginning, we simulated a low-resource condition in these experiments by only using the data from TED, which are roughly 185K sentences.

We evaluated models that are trained both on the translation and POS tagging task. Although the POS data is out-of-domain and significantly smaller than the parallel training data for the translation task (ca. 20% of the size), we see improvements for all three architectures consistently in three metrics. The BLEU scores is improved by more than 1 point and the characTER is reduced by more than 1.5 points. The BEER metric score is improved by more than a half point on both sets.

In more detailed look at this task, we see that the model sharing the most (shrd Dec) performs better than the baseline, but worse than the other two. Therefore, we can conclude that it is helpful to separate the tasks when the components work on different types of data. Whether it is helpful to share the attention layer (shrd Att) or not (shrd Enc) is not clear from this experiment. Therefore, we concentrate on these two architectures in the following experiments.

5.2 Impact of design decisions

Following the initial experiment, we address the following three design questions:

- What kind of influence does the secondary task have?
- How do the different architectures perform?
- Do we need to adapt the training schedule?

In order to clarify the impact of the three hyperparameters (the architectures, the tasks and the training) we performed experiments based on possible combinations. We used two most promising architectures, shrd Enc and shrd Att as discussed in Section 5.1. We use three task combinations, POS+MT, NE+MT and NE+POS+MT. Two training strategies are applied with and without adaptation as described in Section 3.2. These 12 systems are evaluated on the two test sets using three different metrics. Consequently, in total we have 72 measurements for the 12 systems.

Since a first view on the results did not clearly reveal a best performing system, we conducted a more detailed analysis by averaging the results.
Table 1: Results of multi-task learning architectures on the machine translation task (BLEU/BEER/characTER)

<table>
<thead>
<tr>
<th>Task(s)</th>
<th>Arch.</th>
<th>Valid dev 2010</th>
<th>Test tst2013</th>
<th>Test tst2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT</td>
<td></td>
<td>29.91/62.16/51.06</td>
<td>30.85/62.27/51.16</td>
<td>26.12/58.73/55.17</td>
</tr>
<tr>
<td>POS + MT</td>
<td>shrd Enc</td>
<td>30.62/62.77/48.35</td>
<td>31.97/62.72/49.69</td>
<td>27.08/58.99/54.50</td>
</tr>
<tr>
<td></td>
<td>shrd Att</td>
<td>30.51/62.27/49.09</td>
<td>31.76/62.68/49.59</td>
<td>26.86/58.84/53.88</td>
</tr>
<tr>
<td></td>
<td>shrd Dec</td>
<td>30.36/62.34/49.28</td>
<td>31.26/62.31/50.35</td>
<td>26.52/58.48/54.00</td>
</tr>
<tr>
<td>Adapted NE + POS + MT</td>
<td>shrd Enc</td>
<td>30.70/62.96/48.60</td>
<td>32.30/63.25/49.22</td>
<td>27.78/59.74/53.49</td>
</tr>
</tbody>
</table>

We observed that the system with the default schedule performs better in 10 out of 18 cases. One reason for this can be that the default training schedule may not perform as well any more when only a few parameters are observed in every batch. In this case, continuing and concentrating on one task seems to be very important.

In addition, we evaluate the correlation between the tasks involved and the training schedule. The results are shown in the same table. The adapted training schedule has no effect when training on named entities and machine translation. The effect when training on POS tagging and MT is also relatively small. When training the three tasks together, however, the system with an adapted schedule performs always better than the system with the default one. The average BLEU is improved by 0.7. The BEER score and characTER are also improved by 0.5 and 1.2 points.

Inspired by the results, we build the adapted shrd Enc model trained on all three tasks, as shown in Table 1. This model improved the performance by 1.5 BLEU points over the baseline system. Also the BEER score is improved by 1 and the characTER score reduced by 1.8 to 2 points.

5.3 POS Tagging Performance

In addition to the results on the task of translation, we also evaluated the performance on the task of POS tagging. The results are shown in Table 3.

For the validation and test data, we show the error rate on the small tag sets as well as the error rate on the morpho-syntactic tag set. In the table, we always first show the results for the small test set.

The baseline system trained only on the Tiger corpus achieves an error rate of 5.49, for the POS tags in the validation set. For the morpho-syntactic tag of the validation set, it achieves 11.36. The
Table 2: Impact of the training schedule in the machine translation task (BLEU/BEER/character)

<table>
<thead>
<tr>
<th>Systems</th>
<th>Default Schedule</th>
<th>Adapted Schedule</th>
<th>Adapted better</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>29.48/60.89/52.05</td>
<td>29.89/61.08/51.64</td>
<td>25/36</td>
</tr>
<tr>
<td>shrd Enc</td>
<td>29.34/60.85/52.31</td>
<td>30.00/61.25/51.50</td>
<td>17/18</td>
</tr>
<tr>
<td>shrd Att</td>
<td>29.62/61.93/51.78</td>
<td>29.78/60.93/51.79</td>
<td>8/18</td>
</tr>
<tr>
<td>POS + MT</td>
<td>29.41/60.81/51.92</td>
<td>29.78/61.00/51.90</td>
<td>8/12</td>
</tr>
<tr>
<td>NE + MT</td>
<td>29.60/61.00/51.76</td>
<td>29.79/60.96/51.77</td>
<td>5/12</td>
</tr>
<tr>
<td>NE + POS + MT</td>
<td>29.42/60.87/52.46</td>
<td>30.09/61.46/51.25</td>
<td>12/12</td>
</tr>
</tbody>
</table>

Table 3: Results of different multi-task architectures on the POS task

<table>
<thead>
<tr>
<th>Task(s)</th>
<th>Model</th>
<th>Default schedule</th>
<th>Adaptation schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Valid</td>
<td>Test</td>
</tr>
<tr>
<td>POS</td>
<td></td>
<td>5.49/11.36</td>
<td>10.13/17.27</td>
</tr>
<tr>
<td></td>
<td>shrd Att</td>
<td>3.86/9.55</td>
<td>6.98/14.17</td>
</tr>
<tr>
<td></td>
<td>shrd Dec</td>
<td>3.57/9.28</td>
<td>7.40/14.62</td>
</tr>
<tr>
<td>NE + POS + MT</td>
<td>shrd Enc</td>
<td>3.42/9.00</td>
<td>5.86/12.87</td>
</tr>
<tr>
<td></td>
<td>shrd Att</td>
<td>3.08/8.45</td>
<td>6.23/13.28</td>
</tr>
</tbody>
</table>

Performance on the test data is 10.13 and 17.27 for both tag sets. In all systems we used one system to generate the both tag sets. The small tags were evaluated by removing the morpho-syntactic information from the output.

It is clear that all models outperform the baseline. It seems to be very helpful for the POS task to jointly train the model along with the translation task. The MT data is significantly larger than the POS data, which is beneficial for this task.

A more detailed look shows that model adaptation is beneficial for a good performance. In all cases the performance is improved by adapting the model to the POS task. Therefore, when the data of the main task is small compared to the overall training data, adapting on the main task is even more important.

Furthermore, we see improvements when using a third task in all cases. Facilitating this combination of tasks is also helpful for POS tagging.

As we observed in the MT task, the impact and differences brought from each architecture are not huge. The architectures considered in this work perform similar. Even the system sharing all components achieves a comparable performance on this task.

The best performing model, however, is the shrd Enc model, trained on all three tasks and adapted to the task. This model achieved an error of 5.06 on the small tag set. Compared to the baseline performance of 10.13, we can see that the error rate is halved. On the fine-grained tag set, we see an improvement from 17.27 to 11.62, which is a more than 30% reduction in error rate.

5.4 Analysis and Examples

In order to show the influence of the other tasks, we show translation examples in Table 4. For the examples we use the multi-task system trained on all three tasks with the shrd Enc architecture.

A common problem of many neural MT systems is that they do not translate parts of the source sentence, or that parts of the source sentence are translated twice. The baseline system suffers from this, as shown in the first two examples. The translation of the multi-task system is improved compared to the baseline in several aspects. In the first example, the baseline system is not translating the German compound Geburtsfehler into birth defect correctly, but into birth. Although the multi-task system does not generate the translation that exactly matches the reference the translation is understandable. In the second example, the phrase of 10 is not repeated. One explanation for this could be that the additional information from the POS data leads to a better encoding of the structure of the source sentence.

The influence of the named-entity training examples on the translation quality is clearer. In several cases, the model is able to handle named enti-
Table 4: Translation examples

<table>
<thead>
<tr>
<th>German</th>
<th>Reference</th>
<th>Baseline</th>
<th>Multi-task</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Benjamin Franklin&quot; from Walter Isaacson</td>
<td>&quot;John Adams&quot; by David McCullough</td>
<td>&quot;Benjamin Franklin&quot; from Walter Franklin</td>
<td>&quot;John Adams&quot; from David McCullough</td>
</tr>
<tr>
<td>das bedeutet, dass 8 von 10 Entscheidungen...</td>
<td>that means that eight out of 10 of the decisions...</td>
<td>that means that eight of 10 of 10 choices...</td>
<td>that means that eight of 10 decisions...</td>
</tr>
<tr>
<td>sie ist kein Geburtsfehler.</td>
<td>it’s not a birth defect.</td>
<td>she’s not born.</td>
<td>it’s not a birth error.</td>
</tr>
<tr>
<td>darum habe ich infantile Zerebralparese, ...</td>
<td>as a result, I have cerebral palsy,</td>
<td>that’s why I have the infantile,</td>
<td>I have infantile cerebral palsy,</td>
</tr>
</tbody>
</table>
| Prousts Freunde hätten das Land verlassen müssen, ... | you know, Proust’s boyfriends would have to leave the country ... | Proossed friends had to have left the county ... | Prouless friends have to leave the country ...

ties better. As shown in the third and fourth example, the NMT system is not able to copy a named entity from the source to the target, nor to translate rare words. In the third example, the baseline system is not able to generate the correct last name of the first author Isaacson, but is generating the last name from the book title. In the second part of the example, the baseline system completely deletes the author. In contrast, the multi-task system is able to generate the correct sequence. In the fourth example the multi-task example is able to translate Zerebralparese (cerebral palsy), while the baseline system is not able to do it.

We would like to note that as shown in the last example, there are also several cases where the NMT system is not able to translate names or rare words correctly.

6 Conclusion

In this paper we proposed the use of multi-task learning for attention-based encoder-decoder models in order to exploit linguistic resources for NMT. By training the models not only on the machine translation task, but also on other NLP tasks, we yielded clear improvements on the translation performance. Results show that multi-task learning improves the translation up to 1.5 BLEU points and 2 character points. As a by product, we were also able to improved the performance of the POS tagging by 30% to 50% relatively. This is especially helpful since data annotation for many NLP tasks is very time-consuming and expensive. It suggests that multi-task learning is a promising approach to exploit any linguistic annotated data, which is especially important if we have a low-resource condition.

We addressed the influence of three design decisions: the involved tasks, the training schedule and the architecture of the model. The largest influence on the final performance was given by the training schedule. By adapting the system on the individual tasks, we were able to make most use of available additional resources. In this case, we showed that both additional resources, the data for POS tagging as well as the named entity-annotated corpus, were beneficial for the translation quality. It is worth mentioning that this was achieved using corpora from a different domain, i.e. spoken TED talks versus written style. Furthermore, these corpora were significantly smaller than the available parallel data. Finally, the amount of parameter sharing defined by the architecture of the model has less influence on the final performance. Although, the best performance on both tasks was
achieved with a model sharing only the encoder between the tasks.
In this work, the performance of machine translation task was improved by adopting multi-task training with other source language NLP tasks. In future work, we will also investigate methods to include target-language NLP tasks into the joint framework.

Acknowledgments
The project leading to this application has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement n° 645452. This work was supported by the Carl-Zeiss-Stiftung.

References


Abstract
Neural network language and translation models have recently shown their great potentials in improving the performance of phrase-based machine translation. At the same time, word representations using different word factors have been translation quality and are part of many state-of-the-art machine translation systems. In many state-of-the-art machine translation systems, in order to support better translation quality.

In this work, we combined these two ideas by investigating the combination of both techniques. By representing words in neural network language models using different factors, we were able to improve the models themselves as well as their impact on the overall machine translation performance. This is especially helpful for morphologically rich languages which often have a large vocabulary size. Furthermore, it is easy to add additional knowledge, such as source side information, to the model.

Using this model we improved the translation quality of a state-of-the-art phrase-based machine translation system by 0.7 BLEU points. We performed experiments on three language pairs for the news translation task of the WMT 2016 evaluation.

1 Introduction
Recently, neural network models are deployed extensively for better translation quality of statistical machine translation (Le et al., 2011; Devlin et al., 2014). For the language model as well as for the translation model, neural network-based models showed improvements when used during decoding as well as when used in re-scoring.

In phrase-based machine translation (PBMT), word representation using different factors (Koehn and Hoang, 2007) are commonly used in state-of-the-art systems. Using Part-of-Speech (POS) information or automatic word clusters is especially important for morphologically rich languages which often have a large vocabulary size. Language models based on these factors are able to consider longer context and therefore improve the modelling of the overall structure. Furthermore, the POS information can be used to improve the modelling of word agreement, which is often a difficult task when handling morphologically rich languages.

Until now, word factors have been used relatively limited in neural network models. Automatic word classes have been used to structure the output layer (Le et al., 2011) and as input in feed forward neural network language models (Niehues and Waibel, 2012).

In this work, we propose a multi-factor recurrent neural network (RNN)-based language model that is able to facilitate all available information about the word in the input as well as in the output. We evaluated the technique using the surface form, POS-tag and automatic word clusters using different cluster sizes.

Using this model, it is also possible to integrate source side information into the model. By using the model as a bilingual model, the probability of the translation can be modelled and not only the one of target sentence. As for the target side, we use a factored representation for the words on the source side.

The remaining of the paper is structured as following: In the following section, we first review the related work. Afterwards, we will shortly describe the RNN-based language model used in our experiments. In Section 4, we will introduce the factored RNN-based language model. In the next
section, we will describe the experiments on the WMT 2016 data. Finally, we will end the paper with a conclusion of the work.

2 Related Work

Additional information about words, encoded as word factors, e.g. the lemma of word, POS tags, etc., is employed in state-of-the-art phrase-based systems. (Koehn and Hoang, 2007) decomposes the translation of factored representations to smaller mapping steps, which are modelled by translation probabilities from input factor to output factor or by generating probabilities of additional output factors from existing output factors. Then those pre-computed probabilities are jointly combined in the decoding process as a standard translation feature scores. In addition, language models using these word factors have shown to be very helpful to improve the translation quality. In particular, the aligned-words, POS or word classes are used in the framework of modern language models (Mediani et al., 2011; Wuebker et al., 2013).

Recently, neural network language models have been considered to perform better than standard n-gram language models (Schwenk, 2007; Le et al., 2011). Especially the neural language models constructed in recurrent architectures have shown a great performance by allowing them to take a longer context into account (Mikolov et al., 2010; Sundermeyer et al., 2013).

In a different direction, there has been a great deal of research on bringing not only target words but also source words into the prediction process, instead of predicting the next target word based on the previous target words (Le et al., 2012; Devlin et al., 2014; Ha et al., 2014). However, to the best of our knowledge, word factors have been exploited in a relatively limited scope of neural network research. (Le et al., 2011; Le et al., 2012) use word classes to reduce the output layer’s complexity of such networks, both in language and translation models. In the work of (Niehues and Waibel, 2012), their Restricted Boltzmann Machines language models also encode word classes as an additional input feature in predicting the next target word. (Tran et al., 2014) use two separate feed forward networks to predict the target word and its corresponding suffixes with the source words and target stem as input features.

Our work exhibits several essential differences from theirs. Firstly, we leverage not only the target morphological information but also word factors from both source and target sides in our models. Furthermore, we could use as many types of word factors as we can provide. Thus, we are able to make the most of the information encoded in those factors for more accurate prediction.

3 Recurrent Neural Network-based Language Models

In contrast to feed forward neural network-based language models, recurrent neural network-based language models are able to store arbitrary long word sequences. Thereby, they are able to directly model \( P(w|h) \) and no approximations by limiting the history size are necessary. Recently, several authors showed that RNN-based language models could perform very well in phrase-based machine translation. (Mikolov et al., 2010; Sundermeyer et al., 2013)

In this work, we used the torch\(^1\) implementation of an RNN-based language model (Léonard et al., 2015). First, the words were mapped to their word embeddings. We used an input embedding size of 100. Afterwards, we used two LSTM-based layers. The first has the size of the word embeddings and for the second we used a hidden size of 200. Finally, the word probabilities were calculated using a softmax layer.

The models were trained using stochastic gradient descent. The weights were updated using mini-batches with a batch size of 128. We used a maximum epoch size of 1 million examples and selected the model with the lowest perplexity on the development data.

4 Factored Language Model

When using factored representation of words, words are no longer represented as indices in the neural network. Instead, they are represented a tuples of indices \( w = (f_1, \ldots, f_D) \), where \( D \) is the number of different factors used to describe the word. These factors can be the word itself, as well as the POS, automatic learned classes (Och, 1999) or other information about the word. Furthermore, we can use different types of factors for the input and the output of the neural network.

\(^1\)http://torch.ch/
4.1 Input Representation

In a first step, we obtained a factored representation for the input of the neural network. In the experiments, we represented a word by its surface form, POS-tags and automatic word class, but the framework can be used for any number of word factors. Although there are factored approaches for n-gram based language models (Bilmes and Kirchhoff, 2003), most n-gram language models only use one factor. In contrast, in neural network based language models, it is very easy to add additional information as word factors. We can learn different embeddings for each factor and represent the word by concatenating the embeddings of several factors. As shown in the bottom of Figure 1, we first project the different factors to the continuous factor embeddings. Afterwards, we concatenate these embeddings into a word embedding.

The advantage of using several word factors is that we can use different knowledge sources to represent a word. When a word occurs very rarely, the learned embedding from its surface form might not be helpful. The additional POS information, however, is very helpful. While using POS-based language models in PBMT may lead to losing the information about high frequent words, in this approach we can have access to all information by concatenating the factor embeddings.

4.2 Output Representation

In addition to use different factors in the input of the neural network, we can also use different factors on the output. In phrase-based machine translation, n-gram language models based on POS-tags have been shown to be very successful for morphologically rich languages.

Porting this idea to neural network language models, we can not only train a model to predict the original word $f_1$ given the previous words in factor representation $h = (f_{1,1}, \ldots, f_{1,D}), \ldots, (f_{i,1}, \ldots, f_{i,D})$, but also train a model to predict the POS-tags (e.g. $f_2$) given the history $h$.

In a first step, we proposed to train individual models for all factors $1, \ldots, D$ generating probabilities $P_1, \ldots, P_D$ for every sentence. Then these probabilities can be used as features for example in re-scoring of the phrase-based MT system.

Considering that it can be helpful to consider all factors of the word in the input, it can be also helpful to jointly train the models for predicting the different output factors. This is motivated by the fact that multi-task learning has shown to be beneficial in several NLP tasks (Collobert et al., 2011). Predicting all output features jointly requires a modification of the output layer of the RNN model. As shown in Figure 1, we replace the single mapping from the LSTM-layer to the softmax layer, by $D$ mappings. Each mapping then learns to project the LSTM-layer output to the factored output probabilities. In the last layer, we use $D$ different softmax units. In a similar way as the conventional network, the error between the output of the network and the reference is calculated during training.

Using this network, we will no longer predict the probability of one word factor $P_{d, d} \in \{1, \ldots, D\}$, but $D$ different probability distributions $P_1, \ldots, P_D$. In order to integrate this model into the machine translation system we explored two different probabilities. First, we used only the joint probability $P = \prod_{d=1}^D P_d$ as a feature in the log-linear combination. In addition, we also used the joint probability as well as all individual probabilities $P_d$ as features.

4.3 Bilingual Model

Using the model presented before, it is possible to add additional information to the model as well. One example we explored in this work is to use...
the model as a bilingual model (BM). Instead of using only monolingual information by considering the previous target factors as input, we used source factors additionally. Thereby, we can now model the probability of a word given the previous target words and information about the source sentence. So in this case we model the translation probability and no longer the language model probability.

When predicting the target word $w_{i+1}$ with its factors $f_{i+1}, \ldots, f_{i+1,D}$, the input to the RNN is the previous target word $w_i = f_{i,1}, \ldots, f_{i,D}$. Using the alignment, we can find the source word $s_{a(i+1)}$, which is aligned to the target word $w_{i+1}$. When we add the features of source word $s_{a(i+1)} = (f_{s_{a(i+1)},1}, \ldots, f_{s_{a(i+1)},D_s})$ to the ones of the target word $w_i$ and create a new bilingual token $b_i = (f_{i+1,1}, \ldots, f_{i+1,D}; f_{a(i+1),1}, \ldots, f_{a(i+1),D_a})$, we can now predict the target word given the previous target word and the aligned source word.

In the example in Figure 2, we would insert (completed, VVD, 87, ein, ART) to predict (a, DT, 37).

In this case the number of input factors and output factors are no longer the same. In the input, we have $D + D_s$ input factors, while we have only $D$ factors on the output of the network.

5 Experiments

We evaluated the factored RNNLM on three different language pairs of the WMT 2016 News Translation Task. In each language pair, we created an $n$-best list using our phrase-based MT system and used the factored RNNLM as an additional feature in rescoring. It is worth noting that the POS and word class information are already present during decoding of the baseline system by $n$-gram-based language models based on each of these factors. First, we performed a detailed analysis on the English-Romanian task. In addition, we used the model in a German-English and English-German translation system. In all tasks, we used the model in re-scoring of a PBMT system.

5.1 System Description

The baseline system is an in-house implementation of the phrase-based approach. The system used to generate $n$-best lists for the news tasks is trained on all the available training corpora of the WMT 2015 Shared Translation Task. The system uses a pre-reordering technique (Rottmann and Vogel, 2007; Niehues and Kolss, 2009; Herrmann et al., 2013) and facilitates several translation and language models. As shown in Table 1, we use two to three word-based language models and one to two cluster-based models using 50, 100 or 1,000 clusters. The clusters were trained as described in (Och, 1999). In addition, we used a POS-based language model in the English-Romanian system and a bilingual language model (Niehues et al., 2011) in English to German and German to English systems. The POS tags for English-Romanian were generated by the tagger described in (Ion et al., 2012) and the ones for German by RFTagger (Schmid and Laws, 2008).

A full system description can be found in (Ha et al., 2016).

The German to English baseline system uses 20 features and the English to German systems uses 22 features. The English-Romanian system was optimized on the first part of news-dev2016 and the rescoring was optimized on this set and a subset of 2,000.

<table>
<thead>
<tr>
<th>Features</th>
<th>EN-RO</th>
<th>EN-DE</th>
<th>DE-EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordLM</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>POSLM</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>clusterLM</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>BiLM</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>#features</td>
<td>22-23</td>
<td>20</td>
<td>22</td>
</tr>
</tbody>
</table>

In addition, we used discriminative word lexica (Niehues and Waibel, 2013) during decoding and source discriminative word lexica in rescoring (Herrman et al., 2015).

A full system description can be found in (Ha et al., 2016).

The German to English baseline system uses 20 features and the English to German systems uses 22 features.

The English-Romanian system was optimized on the first part of news-dev2016 and the rescoring was optimized on this set and a subset of 2,000.
sentences from the SETimes corpus. This part of the corpus was of course excluded for training the model. The system was tested on the second half of news-dev2016.

The English-German and German-English systems were optimized on news-test2014 and also the re-scoring was optimized on this data. We tested the system on news-test2015.

For English to Romanian and English to German we used an \( n \)-best List of 300 entries and for German to English we used an \( n \)-best list with 3,000 entries.

For decoding, for all language directions, the weights of the system were optimized using minimum error rate training (Och, 2003). The weights in the rescoring were optimized using the ListNet algorithm (Cao et al., 2007) as described in (Niehues et al., 2015).

The RNN-based language models for English to Romanian and German to English were trained on the target side of the parallel training data. For English to German, we trained the model and the Europarl corpus and the News commentary corpus.

5.2 English - Romanian

In the first experiment on the English to Romanian task, we only used the scores of the RNN language models. The baseline system has a BLEU score (Papineni et al., 2002) of 29.67. Using only the language model instead of the 22 features, of course, leads to a lower performance, but we can see clear difference between the different language models. All systems use a word vocabulary of 5K words and we used four different factors. We used the word surface form, the POS tags and word clusters using 100 and 1,000 classes.

If we predict all factors together and use then the joint probability, we can reach the best BLEU score of 28.54 as shown in the last line of the table. This is 0.7 BLEU points better than the initial word based model.

After evaluating the model as the only knowledge source, we also performed experiments using the model in combination with the other models. We evaluated the baseline and the best model in three different configuration in Table 3 using only the joint probability. The three baseline configuration differ in the models used during decoding. Thereby, we are able to generate different \( n \)-best lists and test the models on different conditions.

If we predict all factors together and use then the joint probability, we can reach the best BLEU score of 28.54 as shown in the last line of the table. This is 0.7 BLEU points better than the initial word based model.

After evaluating the model as the only knowledge source, we also performed experiments using the model in combination with the other models. We evaluated the baseline and the best model in three different configuration in Table 3 using only the joint probability. The three baseline configuration differ in the models used during decoding. Thereby, we are able to generate different \( n \)-best lists and test the models on different conditions.

### Table 2: English - Romanian Single Score

<table>
<thead>
<tr>
<th>Input</th>
<th>Prediction</th>
<th>Single</th>
</tr>
</thead>
<tbody>
<tr>
<td>All factors</td>
<td>Word</td>
<td>28.46</td>
</tr>
<tr>
<td>All factors</td>
<td>POS</td>
<td>28.48</td>
</tr>
<tr>
<td>All factors</td>
<td>100 Cl.</td>
<td>28.23</td>
</tr>
<tr>
<td>All factors</td>
<td>1,000 Cl.</td>
<td>28.49</td>
</tr>
<tr>
<td>All factors</td>
<td>All factors</td>
<td>28.54</td>
</tr>
</tbody>
</table>

In Table 3, we tested the word-based and the factored language model using a vocabulary of 5K and 50K words. Features from each model are used in addition to the features of the baseline system. As shown in the table, the word-based RNN language models perform similarly, but both could not improve over the baseline system. One possible reason for this is that we already use several language models in the baseline model and they are partly trained on much larger data. While the RNN models are trained using only the target language model, one word-based language model is trained on the Romanian common crawl corpus. Furthermore, the POS-based and word cluster language models use a 9-gram history and therefore, can already model quite long dependencies.

But if we use a factored language model, we are
able to improve over the baseline system. Using the additional information of the other word factors, we are able to improve the bilingual model in all situations. The model using a surface word vocabulary of 5,000 words can improve by 0.1 to 0.3 BLEU points. The model using a 50K vocabulary can even improve by up to 0.6 BLEU points.

Table 4: English - Romanian Bilingual Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>40.12</td>
<td>29.75</td>
</tr>
<tr>
<td>+ Factored LM 50K</td>
<td>40.87</td>
<td>30.17</td>
</tr>
<tr>
<td>+ Factored BM 5K</td>
<td>41.11</td>
<td>30.44</td>
</tr>
<tr>
<td>+ Factored BM 50K</td>
<td>41.16</td>
<td>30.57</td>
</tr>
</tbody>
</table>

After analyzing the different language models, we also evaluate how we can use the factored representation to include source side information. The results are summarized in Table 4. In these experiments, we used not only the joint probability, but also the four individual probabilities as features. Therefore, we will add five scores for every model, since each model is added to its previous configuration in this experiment.

Exploiting all five probabilities of the language model brought us the similar improvement we achieved using the joint probability from the model. On the test set, the improvements are slightly worse. When adding the model using source side information based on a vocabulary of 5K and 50K words, however, we get additional improvements. Adopting the both bilingual models (BM) along with a factored LM, we improved the BLEU score further leading up to the best score of 30.57 for the test set.

5.3 English - German

In addition to the experiments on English to Romanian, we also evaluated the models on the task of translating English News to German. For the English to German system, we use three factors on the source side and four factors on the target side. In English, we used the surface forms as well as automatic word cluster based on 100 and 1,000 classes. On the target side, we used fine-grained POS-tags generated by the RFTagger (Schmid and Laws, 2008), in addition to the factors for the source side.

The experiments using only the scores of the model are summarized in Table 5. In this experiment, we analyzed a word based- and a factored language models as well as bilingual models. As described in section 4.3, the difference between the language model and the bilingual model is that the latter uses the source side information as additional factor.

Using only the word-based language model we achieved a BLEU score of 20.92. Deploying a factored language model instead, we can improve the BLEU score by 0.7 BLEU points to 21.69. While we achieved a score of 21.33 BLEU points by using a proposed bilingual model, we improved the score up to 21.92 BLEU points by adopting all factors for the bilingual model.

Table 5: English - German Single Score

<table>
<thead>
<tr>
<th>Model</th>
<th>Single</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM 5K</td>
<td>20.92</td>
</tr>
<tr>
<td>Factored LM 5K</td>
<td>21.69</td>
</tr>
<tr>
<td>BM 5K</td>
<td>21.33</td>
</tr>
<tr>
<td>Factored BM 5K</td>
<td>21.92</td>
</tr>
</tbody>
</table>

In addition to the analysis on the single model, we also evaluated the model’s influence by combining the model with the baseline features. We tested the language model as well as the bilingual model on two different configurations. Adopting the factored language model on top of the baseline features improved the translation quality by around 0.4 BLEU points for both configurations, as shown in Table 6. Although the bilingual model could also improve the translation quality, it could not outperform the factored language model. The combination of the two models, LM and BM, did not lead to further improvements. In summary, the factored language model improved the BLEU score by 0.4 points.

5.4 German - English

Similar experiments were conducted on the German to English translation task. For this language pair, we built models using a vocabulary size of 5,000 words. The models cover word surface forms and two automatic word clusters, which are
First, we will evaluate the performance of the system using only this model in rescoring. The results are summarized in Table 7.

Table 7: German - English Single Score

<table>
<thead>
<tr>
<th>Model</th>
<th>Single</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM 5K</td>
<td>26.11</td>
</tr>
<tr>
<td>Factored LM 5K</td>
<td>26.96</td>
</tr>
<tr>
<td>BM 5K</td>
<td>26.77</td>
</tr>
<tr>
<td>Factored BM 5K</td>
<td>26.81</td>
</tr>
</tbody>
</table>

The word based language model achieves a BLEU score 26.11. Extending the model to include factors improves the BLEU score by 0.8 BLEU points to 26.96. If we use a bilingual model, a word based model achieves a BLEU score of 26.77 and the factored one a BLEU score of 26.81. Although the factored model performed better than the word-based models, in this case the bilingual model cannot outperform the language model.

Using these techniques, we are able to improve the translation system on three different language pairs of the WMT 2016 evaluation. We performed experiments on the English-Romanian, English-German and German-English translation task. The suggested technique yielded up to 0.7 BLEU points of improvement on all three tasks.

6 Conclusion

In this paper, we presented a new approach to integrate additional word information into a neural network language model. This model is especially promising for morphologically rich languages. Due to their large vocabulary size, additional information such as POS-tags are expected to model rare words effectively.

Representing words using factors has been successfully deployed in many phrase-based machine translation systems. Inspired by this, we represented each word in our neural network language model using factors, facilitating all available information of the word. We showed that using the factored neural network language models can improve the quality of a phrase-based machine translation system, which already uses several factored language models.

In addition, the presented framework allows an easy integration of source side information. By incorporating the alignment information to the source side, we were able to model the translation process. In this model, the source words as well as the target words can be represented by word factors.

Acknowledgments

The project leading to this application has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 645452. The research by Thanh-Le Ha was supported by Ministry of Science, Research and the Arts Baden-Württemberg.

References


Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2014. Lexical Translation Model Using a Deep Neural Network Architecture. In Proceedings of the 11th International Workshop on Spoken Language Translation (IWSLT’14), Lake Tahoe, CA, USA.


Automatic Post-Editing for the DiscoMT Pronoun Translation Task

Liane Guillou
School of Informatics
University of Edinburgh
Scotland, United Kingdom
L.K.Guillou@sms.ed.ac.uk

Abstract
This paper describes an automated post-editing submission to the DiscoMT 2015 shared task on pronoun translation. Post-editing is achieved by applying pronoun-specific rules to the output of an English-to-French phrase-based SMT system.

1 Introduction
The shared task (Hardmeier et al., 2015) focuses on the translation of the English pronouns “it” and “they” into French. While they both serve multiple functions in English, the most significant is as anaphoric pronouns, referring back to an entity previously mentioned in the discourse, known as the antecedent.

When translated into French, anaphoric pronouns must agree with their antecedent in terms of both number and grammatical gender. Therefore, selecting the correct pronoun in French relies on knowing the number and gender of the antecedent. This presents a problem for current state-of-the-art Statistical Machine Translation (SMT) systems which translate sentences in isolation.

Inter-sentential anaphoric pronouns, i.e. those that occur in a different sentence to their antecedent, will be translated with no knowledge of their antecedent. Pronoun-antecedent agreement therefore cannot be guaranteed. Even intra-sentential pronouns, i.e. those that occur in the same sentence as their antecedent, may lack sufficient local context to ensure agreement.

The English pronoun “it” may also be used as a pleonastic or event pronoun. Pleonastic pronouns such as the “it” in “It is raining” or the “il” in “Il pleut” do not refer to anything but are required by syntax to fill the subject-position slot. Event pronouns may refer to a verb, verb phrase or even an entire clause or sentence. The pronoun “they” may also serve as a generic pronoun, as in “They say it always rains in Scotland” – here “they” does not refer to a specific person or group. For each pronoun type, translations into French must meet different requirements.

This paper presents an automatic post-editing approach which applies two pronoun-specific rules to the output of an English-to-French phrase-based SMT system. One rule handles anaphoric pronouns and the other handles non-anaphoric (i.e. event and pleonastic) pronouns.

The advantage of a post-editing approach is that the translations of both pronouns and their antecedents (for anaphoric pronouns) are already known. There is therefore no need to keep track of this information within the decoder. Instead, the problem becomes one of identifying incorrectly translated pronouns and amending them based on information extracted from the source-language text. The aim is to leverage knowledge about the target-language and through this maximise the number of changes that will improve the pronoun translations, whilst also attempting to minimise those that may have a detrimental effect.

The post-editing rules make use of information automatically obtained from the source-language text. The risk of doing this is that inaccurate information could lead to incorrect translations. As post-editing takes place after translation, the decoder and language model can no longer be relied upon to recover from bad decisions. However, due to the simplicity of the approach and encouraging results from Weiner (2014) for the English-German pair, post-editing is worth exploring.

2 Post-editing Overview
Using the ParCor corpus (Guillou et al., 2014) annotations as a model, automated tools are applied to the full text of each (sentence-split) source-language document in the dataset to extract the following information: anaphoric vs. non-anaphoric pronouns, subject vs. object position and the an-
Detect non-anaphoric “it” (NADA)

Identify subject/object “it” from dependency parse (CoreNLP)

Identify antecedents / coref chains (CoreNLP)

Source document pre-processing

Phrase-based SMT system

Obtain word-alignments from decoder

Post-edit SMT output

Identify translation of each “it/they” instance in SMT output

Anaphoric pronoun?

Subject pronoun?

No change

Default to “c’/ce”

No

Yes

No

Default to “la”

Yes

Get number/gender of head translation (dictionary)

Select pronoun that agrees with antecedent (from “il/ils/elle/elles”)

No

No

Table 1: Baseline training, tuning and development data.

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
<th>Parallel Sentences</th>
<th>Monolingual Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>TED, Europarl, News Commentary</td>
<td>2,372,666</td>
<td></td>
</tr>
<tr>
<td>Tuning</td>
<td>dev2010 + tst2011</td>
<td>1,705</td>
<td></td>
</tr>
<tr>
<td>Development test</td>
<td>tst2010</td>
<td>1,664</td>
<td></td>
</tr>
<tr>
<td>Development test</td>
<td>tst2012</td>
<td>1,124</td>
<td></td>
</tr>
<tr>
<td>Language model</td>
<td>TED, Europarl, News Commentary and News</td>
<td>33,869,133</td>
<td></td>
</tr>
</tbody>
</table>

The baseline system used to produce the SMT output is of a similar design to that provided as part of the shared task resources. It is a phrase-based system built using the Moses toolkit (Koehn et al., 2007) and trained/tuned using only the pre-processed (tokenised, lower-cased) parallel data provided for the shared task. Training, tuning and (development) test data are described in Table 1.

Word alignments are computed using Giza++ with grow-diag-final-and symmetrisation, and with sentences restricted to 80 tokens or fewer (as Giza++ produces more robust alignments for shorter sentences). The maximum phrase length is set to 7. As memory and disk space are not a concern, sig-test filtering which prunes unlikely phrase pairs from the phrase table, is not used in training the baseline system. Tuning is performed using MERT (Och, 2003) with an N-best list of 200, and using the dev2010+tst2011 data.

The language model is a 5-gram KenLM (Heafield, 2011) model, trained using lmplz, with modified Kneser-Ney smoothing and no pruning. The memory optimisations that were made for the shared task baseline\(^1\) are not replicated as they are not required. The language model uses the probing data structure; the fastest and default data structure for KenLM, it makes use of a hash table to store the language model n-grams.

By restricting the training data to sentences of 80 or fewer tokens, the baseline SMT system is trained on 27,481 fewer parallel sentences than the shared task baseline. There are no other differences in the data used; for tuning, development-testing or language model construction.

The baseline SMT system scores nearly one BLEU point higher than the shared task baseline for the IWSLT 2010 (34.57 vs. 33.86) and 2012 (41.07 vs. 40.06) test sets. BLEU scores were calculated using the case-insensitive, multi-bleu perl script provided in the Moses toolkit.

The decoder is set to output word alignments, which are used later for automatic post-editing.

\(^1\)Provided as part of the shared task resources
4 Extracting Source-language Information

Guided by the ParCor annotation scheme, the following is extracted from the source-language text:

- Position: subject or object ("it" only)
- Function: anaphoric or non-anaphoric (i.e. pleonastic / event, for "it" only)
- Antecedent: for anaphoric pronouns only

The first step is to identify whether the pronoun appears in subject or object position. The pronoun "it" may be used in either position, unlike "they" which is always a subject-position pronoun. When translating into French it is necessary to ensure that each instance of "it" is correctly translated, with different French pronouns used depending on the position that the pronoun fills. Instances of "it" are categorised as being either subject- or object-position pronouns using the dependency parser provided as part of the Stanford CoreNLP tool\(^2\). Subject-position pronouns are those that participate in an nsubj or nsubjpass dependency relation.

The next step is to determine the function of each instance of "it". NADA (Bergsma and Yarowsky, 2011) is used as it considers the entire sentence, unlike the pleonastic sieve in the Stanford coreference resolution system (Lee et al., 2011), which uses only fixed expressions to identify pleonastic "it". Instances of "it" with a NADA probability below a specified threshold are treated as non-anaphoric, and those above, as anaphoric. Here, a non-anaphoric pronoun is either an event or pleonastic pronoun; a finer distinction cannot be made using currently available tools. The NADA threshold is set to 0.41 (see Section 6).

For instances of "it" identified as anaphoric, and all instances of "they", the pronoun’s nearest non-pronominal antecedent is extracted using the coreference resolution system (Raghunathan et al., 2010; Lee et al., 2011) provided in the Stanford CoreNLP tool\(^3\). To avoid falsely identifying coreference chains across document boundaries, the source-language text is split into documents prior to coreference resolution. Full coreference chains are retained in case the nearest antecedent is not translated by the baseline SMT system.

NADA and CoreNLP were run on tokenised, but not lower-cased data, in order to ensure parser accuracy. The tokenisation and sentence segmentation is the same as that used in the pre-processed data distributed for the shared task. The CoreNLP tool was run with the following annotators: tokenize, ssplit, pos, lemma, ner, parse and dcoref. The following parameters were set to true: tokenize.whitespace and ssplit.eolonly.

5 Automatic Post-Editing Rules

Automatic post-editing is applied to the 1-best output of the baseline SMT system described in Section 3. The process makes use of information extracted from the source-language text (Section 4) and the word alignments output by the decoder.

For each source-language pronoun, one of two post-editing rules is applied, depending on whether the pronoun is identified as anaphoric or non-anaphoric. The rules are outlined in Figure 1 and described in detail in the following sections.

5.1 Anaphoric Rule

This rule is applied to all instances of "they" and subject-position "it" that are identified as anaphoric, both inter- and intra-sentential. Cataphoric pronouns, where the pronoun appears before its antecedent, are very rare (Guillou et al., 2014) and are ignored for the sake of simplicity. Instances of object-position "it" are excluded as the focus of the shared task is on subject-position pronouns only. Target-language pronoun forms are predicted using the projected translation of the head of the nearest non-pronominal antecedent.

On the source-language side:

1. Identify the nearest non-pronominal antecedent
2. Identify the antecedent head word (provided by CoreNLP for each antecedent)
3. Using word alignments output by the decoder, project source-language pronoun and antecedent head positions to the SMT output

On the target-language side (SMT output):

4. If no antecedent can be found for the pronoun, do not attempt to amend its translation. (It may be non-anaphoric but not detected by NADA)
5. For all other pronouns, use the word alignments to identify the translations of the pronoun and antecedent head
6. Extract the number and gender of the antecedent head translation via a dictionary of

\(^2\)Stanford CoreNLP version 3.3.1 http://nlp.stanford.edu/software/corenlp.shtml
\(^3\)Considers pronoun-antecedent distances ≤ 3 sentences
French nouns extracted from the Lefff (Sagot, 2010) and augmented by entries from dict.cc.

7. If the antecedent head word is aligned to multiple words in the translation select the right-most noun (should be the head in most cases)

8. If the antecedent head translation is a noun:
   (a) Predict “elle” for feminine, singular; “il” for masculine, singular
   (b) Predict “elles” for feminine, plural; “ils” for masculine, plural
   (c) If the antecedent is split-reference of the format N and N, split it into two nouns. If both are feminine, predict “elles”, otherwise predict “ils”

9. If the antecedent head translation is not a noun (i.e. not in the dictionary) or is not translated:
   (a) Traverse further back through the coreference chain and repeat from step 5
   (b) If the antecedent head is not translated, apply a default value. If the source-language pronoun is translated as a pronoun, but not “il/elle” (for “it”) or “ils/elles” (for “they”), predict “it” for “it” and “ils” for “they”. If the pronoun is not translated, do nothing as the SMT system may have correctly learned to drop a pronoun

10. If the pronoun in the SMT output and the predicted translation disagree, the post-editing rule replaces the translation in the SMT output with the predicted value

This method allows for the prediction of a plural noun for cases where an English singular noun is translated into French using a plural noun. For example, “vacation” is singular in English but may be translated as “vacances” (plural) in French.

5.2 Non-Anaphoric Rule

This rule is applied to instances of subject-position “it” that are identified as non-anaphoric, i.e. those with a NADA probability below the specified threshold. It does not apply to instances of “they”.

The first step is to identify the translation of the pronoun (using the word alignments). The translation that should appear in the post-edited SMT output is then predicted.

1) Translation is an event/pleonastic pronoun: As NADA does not appear to distinguish event and pleonastic pronouns (i.e. both are considered equally non-anaphoric; see Section 6) it is not straightforward to predict a correct translation for non-anaphoric “it”. The French pronoun “ce” may function as both an event and a pleonastic pronoun, but “il” is used only as a pleonastic pronoun. All instances of “it” translated as “cela/ce” are left as they are in the SMT output. Changing them may do more harm than good and would be performed in an uninformed manner. The hope is that these pronouns, or at least the pleonastic ones, may be correctly translated using local context.

2) Translation is another pronoun: If an instance of “it” is translated as a pronoun outwith the set “cela/ce”, it will be corrected to the default “ce” (or “c’” if the next word in the SMT output starts with a vowel or silent “h”). The French pronouns “cela/ce/la/la” may be used as neutral pronouns, referring to events/actions/states or general classes of people/things, and “il/ce/la/la” may be used as impersonal pronouns, marking the subject position but not referring to an entity in the text, i.e. pleonastically (Hawkins et al., 2001). “cela/ce/la/la” may all be used as either pleonastic or event pronouns. “ce” is selected as the default as it occurs most frequently in the training data, suggesting common usage. There are some cases in which only “il” should be used as the impersonal pronoun, such as expressions of time. These are not easy to detect and are therefore ignored.

3) Translation is not a pronoun: If an instance of “it” is translated using something other than a pronoun, it is not amended. This may also indicate that the pronoun has been dropped.

4) No translation: There is no provision for handling cases where a pleonastic or event pronoun may in fact be required but was dropped in the SMT output. I am not aware of any tools that can separate pleonastic and event instances of “it” for English and inserting a pronoun might not be the correct thing to do in all cases.

If the pronoun in the SMT output and the predicted translation disagree, the post-editing rule replaces the translation in the SMT output with the predicted value.

6 Setting the NADA Threshold

NADA returns a probability between 0 and 1, and the decision as to whether an instance of “it” is
anaphoric can be made by thresholding this probability. The NADA documentation suggests a general threshold value of 0.5; for probabilities over this value the pronoun is said to be referential (i.e. anaphoric) and for those below this value, that it is non-referential. However, different threshold values may be appropriate for different genres.

The TED-specific NADA threshold was set using the manual ParCor (Guillo et al., 2014) annotations over the TED Talks portion of the corpus. NADA was run over the English TED Talks and the probabilities it assigned for each instance of “it” were compared with the pronoun type labels (i.e. anaphoric/pleonastic/event).

There are 61 instances of “it” marked as pleonastic in the ParCor annotations. Looking at all 133 instances of “it” in the ParCor TED Talks for which their NADA probabilities fall below 0.5, there are a mixture of pleonastic, event, and “anaphoric with no explicit antecedent” pronouns. These could acceptably be treated as non-referential. However, there are also a number of anaphoric pronouns that fall into this range and it would be unacceptable to treat these as non-referential. Setting the threshold is therefore a trade-off between precision and recall. Whatever threshold is set, there will be both false positives and false negatives. At a threshold of ≤ 0.41, 37 (60.66%) of pronouns marked as pleonastic in ParCor are correctly identified and 24 (39.34%) are not. 37 pronouns marked in ParCor as event pronouns and 35 anaphoric pronouns (of which 4 have no explicit antecedent) are also (incorrectly) identified as non-referential.

### 7 Post-Editing Statistics

The shared task test set contains 307 instances of “they” and 809 instances of “it”. Automated preprocessing of the source-language texts identifies 581 instances of “it” as subject-position pronouns and 228 as object-position pronouns (for which no change will be made). Of the 888 instances of “it” and “they” identified as subject-position pronouns, the translation of 316 are changed in the SMT output by the post-editing rules. 303 changes are applied to pronouns identified as anaphoric (36 “they” and 267 “it”) and 13 to pronouns identified as non-anaphoric. The pronoun changes are summarised in Table 2. 10 pronouns were not trans-....

<table>
<thead>
<tr>
<th>Pronoun type</th>
<th>Form</th>
<th>Before</th>
<th>After</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-anaphoric</td>
<td>it</td>
<td>ς</td>
<td>ce/c'</td>
<td>7</td>
</tr>
<tr>
<td>Non-anaphoric</td>
<td>it</td>
<td>cela</td>
<td>ce/c'</td>
<td>3</td>
</tr>
<tr>
<td>Non-anaphoric</td>
<td>it</td>
<td>elle</td>
<td>ce/c'</td>
<td>1</td>
</tr>
<tr>
<td>Non-anaphoric</td>
<td>it</td>
<td>le</td>
<td>ce/c'</td>
<td>1</td>
</tr>
<tr>
<td>Non-anaphoric</td>
<td>it</td>
<td>on</td>
<td>ce/c'</td>
<td>1</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>il</td>
<td>ils</td>
<td>3</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>il</td>
<td>elle</td>
<td>51</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>il</td>
<td>elles</td>
<td>3</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>elle</td>
<td>il</td>
<td>17</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>elle</td>
<td>ils</td>
<td>1</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>lei/</td>
<td>il</td>
<td>3</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>on</td>
<td>il</td>
<td>1</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>η</td>
<td>il</td>
<td>10</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>η</td>
<td>ils</td>
<td>2</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>η</td>
<td>elle</td>
<td>5</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>cela</td>
<td>il</td>
<td>6</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>cela</td>
<td>elle</td>
<td>3</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>cela</td>
<td>elles</td>
<td>1</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>ce/c'</td>
<td>il</td>
<td>84</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>ce/c'</td>
<td>ils</td>
<td>5</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>ce/c'</td>
<td>elle</td>
<td>68</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>it</td>
<td>ce/c'</td>
<td>elles</td>
<td>4</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>they</td>
<td>ils</td>
<td>elles</td>
<td>32</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>they</td>
<td>elles</td>
<td>ils</td>
<td>4</td>
</tr>
</tbody>
</table>

**Total** 316

### Table 2: Automated post-editing changes

The most frequent changes are “c'/ce” → “il” (84), “c'/ce” → “elle” (68), “il” → “elle” (51), and “ils” → “elles” (32). The change “c'/ce” → “il/elle” takes place due to the decision to use gendered translations of all instances of “it” identified as anaphoric (even if “c'/ce” might also have been an acceptable translation). Biases in the training data may account for some of the other changes. For example, the change “ils” → “elles” may result from the common alignment of “they” to “ils” which arises due to the rule in French that “ils” is used unless all of the antecedents are feminine (in which case “elles” is used). This may result in more masculine pronouns requiring replacement with a feminine pronoun than vice versa.

The changes “il” → “elle” and “ils” → “elles” are made to conform with the gender of the translation of the antecedent head of an anaphoric pronoun. The post-editing rules also allow for changes from singular to plural (and vice versa) and from one number and gender to another. For example in translating “it” → “vacation” the anaphoric rule would allow for an instance of “il” (masc. sg.) in the SMT output to be changed to “elles” → “vacances” (fem. pl.).
8 Results

The official shared task results report a BLEU score of 36.91 for the post-edited SMT output. This score is lower than the official baseline system (37.18), comparable with the UU-Tiedemann system (36.92), and higher than the other competing systems. However, the post-editing system outperformed only two of the five competing systems in terms of the accuracy measures, suggesting that BLEU is a poor measure of pronoun translation performance. The accuracy with OTHER measure reveals that the post-edited SMT output contains correct translations for only 114/210 pronoun instances, according to human judgements.

There is a small decrease of 0.36 BLEU between the baseline system used to provide SMT output and the post-edited version for the test set (38.83 vs. 38.47 respectively, as calculated using case-insensitive multi-bleu).

An examination of the human judgements from the shared task manual evaluation reveals that the post-editing process makes many mistakes. 34 instances were worsened by post-editing and only 9 improved. The remaining instances were neither better nor worse following post-editing. Translation accuracy differs for “it” and “they”. For “it” 32 instances are judged to be correct vs. 60 incorrect. The opposite is observed for “they”, with 47 instances judged to be correct vs. 14 incorrect. (Instances marked as “other” or “bad translation” cannot be commented upon further and are excluded from the counts). The poor translation of “it” could be due to the method used to identify anaphoric and non-anaphoric instances (no such method was used for “they”), differences in coreference resolution accuracy for “it” and “they”, or something else entirely.

9 Limitations of Post-Editing

Although specific failures in the baseline SMT system, the external tools and the post-editing rules await detailed analysis, the following possible problems with the external tools should at least be considered: incorrect identification of subject-position “it”, of non-anaphoric pronouns and of antecedents. These problems may arise from a mismatch between the TED Talks domain, and the domain of the data that the tools were trained on.

As the post-editing rules affect only pronouns, agreement issues may occur. For example, if the baseline SMT system outputs “ils sont partis” (“they[masc] have left”) and the post-editing rules amend “ils” to “elles”, the verb “partis” should also be amended: “elles sont parties” (“they[fem] have left”). Agreement issues could be addressed within a dependency-parser-based post-editing framework such as the Depfix system for Czech (Mareček et al., 2011; Rosa, 2014).

Another limitation is the lack of an available tool for detecting event pronouns. Whilst NADA appears to detect some of these, it is an accidental consequence of its inability to distinguish a pleonastic (“il/ce”) from an event pronoun (“ce”).

While post-editing rules could potentially be written to insert a pronoun in the SMT output where one is syntactically required in the the target language, or to delete a pronoun for syntactic or stylistic reasons, this was not done in the current system.

The approach may also be difficult to extend to other languages which are less well provisioned in terms of parsers and coreference resolution systems or for which baseline SMT quality is poor.

10 Summary and Future Work

The post-editing approach makes use of two pronoun-specific rules applied to the output of a baseline English-to-French phrase-based SMT system. One rule handles anaphoric pronouns, the other handles non-anaphoric pronouns.

Before extending this work to develop new rules or applying the technique to other language pairs, it is important to first understand where the post-editing method performs well and where it performs poorly. A detailed analysis of the post-edits as compared with the human judgements from the manual evaluation would be a logical first step. Limitations of both the external tools and the post-editing rules should be assessed.

Acknowledgements

Thanks to Professor Bonnie Webber and the three anonymous reviewers for their feedback. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement n° 645452 (QT21).
References

Liane Guillou, Christian Hardmeier, Aaron Smith, Jörg Tiedemann, and Bonnie Webber. 2014. ParCor 1.0: A Parallel Pronoun-Coreference Corpus to Support Statistical MT. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavík, Iceland, European Language Resources Association (ELRA).


Liane Guillou
School of Informatics
University of Edinburgh
Scotland, United Kingdom
L.K.Guillou@sms.ed.ac.uk

Bonnie Webber
School of Informatics
University of Edinburgh
Scotland, United Kingdom
bonnie@inf.ed.ac.uk

Abstract

Previous work on pronouns in SMT has focussed on third-person pronouns, treating them all as anaphoric. Little attention has been paid to other uses or other types of pronouns. Believing that further progress requires careful analysis of pronouns as a whole, we have analysed a parallel corpus of annotated English-German texts to highlight some of the problems that hinder progress. We combine this with an assessment of the ability of two state-of-the-art systems to translate different pronoun types.

1 Introduction

Previous work on the translation of pronouns in Statistical Machine Translation (SMT) has focussed on the specific problem of translating anaphoric pronouns – i.e., ones that co-refer with an antecedent entity previously mentioned in the discourse (Le Nagard and Koehn, 2010; Hardmeier and Federico, 2010; Guillou, 2012; Novák et al., 2013; Hardmeier, 2014; Weiner, 2014). This is because languages differ in how an anaphoric pronoun relates to its antecedent, and the relationship does not fit naturally into the SMT pipeline. Some pronoun forms also have non-anaphoric uses, and there are other types of pronouns. Languages also differ as to what types of pronouns are used for what purposes.

To investigate similarities and differences in pronoun usage across languages, we conducted an analysis of the ParCor corpus of pronoun annotations over a set of parallel English-German texts. The corpus contains a collection of texts from two different genres: 8 EU Bookshop publications (written text) and 11 TED Talks (scribed planned speech). In the ParCor annotations, each pronoun is marked as being one of eight types: Anaphoric/cataphoric, event reference, extra-textual reference, pleonastic, addressee reference, speaker reference, generic reference, or other function. Additional features are recorded for some pronoun types, for example anaphoric/cataphoric pronouns are linked to their antecedents. Full details of the annotation scheme are provided in Guillou et al. (2014).

Through analysing similarities and differences in pronoun use in these parallel texts, we hope to better understand the problems of translating different types of pronouns. This knowledge may in turn be used to build discourse-aware SMT systems in the future. In addition, through analysing translations produced by state-of-the-art systems, we hope to understand how well current systems translate a range of pronoun types. This information may be used to identify the pronoun types where future efforts would be best directed.

The advantage of using the ParCor corpus is that it allows us to conduct part of the analyses automatically once we have word-aligned the parallel texts. The annotations also allow for the separation of ambiguous pronouns such as “it” which may serve as an anaphoric, event or pleonastic pronoun. This allows for a more granular analysis than has been provided in other similar studies.

2 Previous Work

There has been previous work both on comparing pronoun usage in English and German (in the genre of business letters using comparable rather than parallel texts (Becher, 2011) and for the multi-genre GECCo corpus (Kunz and Lapshinova-Koltunski, 2015)) and on pronoun translation accuracy by SMT systems (Hardmeier and Federico, 2010; Novák et al., 2013;
Weiner, 2014), these being relatively small scale. The main focus, however, has been on building models to improve pronoun translation in SMT through targeting different stages of the translation process. These include pre-annotation of the source-language data (Le Nagard and Koehn, 2010; Guillou, 2012), decoder features (Hardmeier and Federico, 2010; Novák et al., 2013; Hardmeier, 2014; Weiner, 2014) and post-editing / re-ranking (Weiner, 2014). Despite these efforts, little progress has been made.

In the most comprehensive study to date, Hardmeier (2014) concludes that current models for pronoun translation are insufficient and that “future approaches to pronoun translation in SMT will require extensive corpus analysis to study how pronouns of a given source language are rendered in a given target language”. This paper reports on such a corpus analysis.

3 Analysis of Manual Translation

Identifying and understanding systematic differences in pronoun use between a pair of languages may help inform the design of SMT systems. With this in mind, we compared original English texts and their human-authored German translations in the ParCor corpus, for both genres, at the corpus, document and sentence levels.

3.1 Corpus-level

Corpus-level comparison reveals the first differences between pronoun use in the two languages. (See Table 1. Some counts differ from those in (Guillou et al., 2014) due to minor changes prior to corpus release and the automatic addition of first person pronouns and German “man”.) Specifically, the German translations contain more anaphoric and pleonastic pronouns than the original English texts. (A pleonastic pronoun does not refer to an antecedent, e.g. “It is raining” / “Es regnet”.) Paired t-tests show that this difference is significant for pleonastic pronouns in both the TED corpus, \( t(10) = -5.08, p < .01 \), and the EU Bookshop corpus, \( t(10) = -3.68, p < .01 \). The difference in anaphoric pronoun use is significant for the TED corpus, \( t(7) = -3.52, p < .01 \), but not the EU Bookshop corpus, \( t(7) = -1.09, (p=0.31) \).

3.2 Document-level

Again, at the document-level we observe that the German translations typically contain more anaphoric and pleonastic pronouns than the original English texts. (See Table 2 for the pronoun counts of a randomly selected document, 767.)

Table 1: Pronoun type counts for English (source) and German (translation) texts in ParCor. Counts per 1000 tokens are provided in parentheses. N/A indicates that the type is not marked for one of the corpora.

<table>
<thead>
<tr>
<th>Pronoun Type</th>
<th>TED Talks</th>
<th>EU Bookshop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>English</td>
<td>German</td>
</tr>
<tr>
<td>Anaphoric</td>
<td>896 (27.71)</td>
<td>1,228 (40.52)</td>
</tr>
<tr>
<td>Anaphoric (pronominal adverb)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Cataphoric</td>
<td>5 (0.16)</td>
<td>16 (0.53)</td>
</tr>
<tr>
<td>Event</td>
<td>264 (8.26)</td>
<td>331 (10.92)</td>
</tr>
<tr>
<td>Event (pronominal adverb)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Extra-textual reference</td>
<td>52 (1.63)</td>
<td>26 (0.86)</td>
</tr>
<tr>
<td>Pleonastic (non-referential)</td>
<td>61 (1.91)</td>
<td>224 (7.39)</td>
</tr>
<tr>
<td>Addressee reference</td>
<td>499 (16.51)</td>
<td>525 (17.32)</td>
</tr>
<tr>
<td>Speaker reference</td>
<td>1,386 (43.35)</td>
<td>1,467 (48.41)</td>
</tr>
<tr>
<td>Generic</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Pronoun (other)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Pronoun (unsure)</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total</td>
<td>3,153 (98.62)</td>
<td>3,817 (125.95)</td>
</tr>
</tbody>
</table>

Table 2: Pronoun type counts for TED Talk 767. Counts per 1000 tokens provided in parentheses.
not simply a consequence of stylistic differences over authors or speakers. A presentation of the full analysis would, however, require a longer paper.

Documents in ParCor were originally produced in English and then translated into German. To ascertain whether similar patterns of pronoun use can be observed for the opposite translation direction, we annotated two German TEDx talks and their English translations, again using the guidelines described in Guillou et al. (2014).

We observed similar patterns, with more pleonastic pronouns used in German than in English (19 vs. 11 pleonastic pronouns in one document, and 15 vs. 2 in the other). For anaphoric pronouns, one document has 119 in the German original and 140 in the English translation, with near equal numbers (54 vs. 51) in the other document. With only two documents it is not possible to confirm whether German systematically makes use of more anaphoric and pleonastic pronouns, but cf. Becher (2011) who points to several patterns, in particular the insertion of explicit possessive pronouns in English-to-German translation and pronominal adverbs in the opposite direction.

3.3 Sentence-level

Pronoun counts at the corpus and document levels are simply raw counts. They do not tell us anything about cases in which a pronoun is used in the original text and dropped from the translation (deletions), or is absent from the original text but present in the translation (insertions). To discover this, we need to drill down to the sentence–level.

We start with the sentence–aligned parallel texts provided as part of the ParCor release. In order to identify the German translation of each pronoun in the original English text, we compute word alignments using Giza++ (https://code.google.com/p/giza-pp/) with grow-diag-final-and symmetrisation. To ensure robust alignments, we concatenated the ParCor texts and additional data – specifically, the IWSLT 2013 shared task training data (for TED and TEDx) and Europarl data (for EU Bookshop). We consider an English and German pronoun to be equivalent if the following conditions hold: (a) a word alignment exists between them, and (b) they share the same pronoun type label in the ParCor annotations.

To evaluate the word-alignment quality we examined a random sample of 100 parallel sentences from the TED corpus. The sentences contain 213 English and 241 German pronouns. We define a bad alignment as one where a pronoun is aligned to something that is not the corresponding pronoun in the other language, or should be unaligned but is not. We find that 6.57% of English and 9.12% of German pronouns are part of a bad alignment.

Taking TED talk 767 as an example and using the combination of pronoun type and alignments to identify a source-target pronoun match, we observe many mismatches. Table 3 shows that 412 pronouns are unique to either the English original or the German translation, with only 298 matching English-German pronoun pairs. The largest absolute difference lies in the number of anaphoric pronouns in the target for which there is no comparable pronoun in the source (anaphoric insertions), followed by pleonastic insertions.

<table>
<thead>
<tr>
<th>Pronoun Type</th>
<th>English (deletion)</th>
<th>German (insertion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anaphoric</td>
<td>49</td>
<td>117</td>
</tr>
<tr>
<td>Cataphoric</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Event</td>
<td>26</td>
<td>36</td>
</tr>
<tr>
<td>Extra-textual ref.</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Pleonastic</td>
<td>3</td>
<td>49</td>
</tr>
<tr>
<td>Addressee reference</td>
<td>31</td>
<td>20</td>
</tr>
<tr>
<td>Speaker reference</td>
<td>30</td>
<td>37</td>
</tr>
<tr>
<td>Pronoun (unsure)</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>146</td>
<td>266</td>
</tr>
</tbody>
</table>

Table 3: Sentence-level pronoun type + alignment mismatches for TED Talk 767

There is no single reason for anaphoric deletions: Anaphoric pronouns may be omitted from the German output for stylistic reasons, as a result of paraphrasing or possibly to conform with language-specific constraints. With respect to anaphoric insertions, intra-sententially, many correspond to relativizers in English. That is, while in English a relative clause is introduced with a that-, wh- or null-relativizer, an anaphoric pronoun serves as a relativizer in German.8 For example, “that” in “The house that Jack built” is a relativizer and the corresponding “das” in “Das Haus, das Jack gebaut hat” is a relative pronoun. Manual analysis of the German translation for TED Talk 767 identified 42 cases where an anaphoric pronoun was inserted as a relative pronoun corresponding to a relativizer in English. While this does not explain all of the anaphoric insertions, it is frequent enough to deserve further attention.

8The ParCor corpus has not marked instances of that when used as a relativizer in English.
Several fixed expressions in English appear to trigger pleonastic insertions in German. A commonly observed pair is “There +be”?“Es gibt”. These existential there constructions are not annotated in ParCor, but their presence accounts for some (not all) of the insertions of pleonastic pronouns in German. As the fixed expressions are short and occur frequently, phrase-based systems could be expected to provide accurate translations.

3.4 Discussion

We have observed differences in pronoun use in both genres of the ParCor corpus. Since SMT systems are trained on parallel data similar to that in ParCor, it is important to be aware that content words such as nouns and verbs are more likely to be faithfully translated as there are fewer ways to convey the same meaning. On the other hand, there is more variation in the translation of function words such as pronouns — for example in active to passive conversions (and vice versa). Where there is a lot of variation the SMT system may not be able to learn accurate mappings.

To this is added the problem of ambiguous pronouns such as “it”, for which the anaphoric and pleonastic forms both translate as “es” in German. These frequent alignments in the training data may also bias the likelihood that “it” is incorrectly translated as “es” (neuter), even if a feminine or masculine pronoun is required in German.

4 Assessing Automated Translation

Analyses of the output of state-of-the-art SMT systems provide an indication of how well current systems are able to translate pronominal coreference — what they are good and bad at. We follow our analysis of manual translation and examine English-to-German translation for anaphoric pronouns (“it” and “its”) and relativizers.

For our state-of-the-art systems, we selected two systems from the IWSLT 2014 shared task in machine translation (Birch et al., 2014). The first is a phrase-based system that incorporates factored models for words, part-of-speech tags and Brown clusters. The second is a syntax-based, string-to-tree, system. Both systems were trained using a combination of TED data and corpora provided for the WMT shared task. Here, TED talks are considered to be in-domain, with the EU Bookshop texts considered out-of-domain.

We are not interested in making direct comparisons between the two systems, as their different training makes such comparisons unfair. However, similarities in the translation accuracy of two systems can show that our findings are not specific to a single system or type of system.

For manual translation, we can assume that a pronoun is accurately translated, inserted or dropped, as part of a close translation of the original sentence or an acceptable paraphrase. As such, it is reasonable to use automated analysis based on the ParCor annotations and alignments between the texts. With automated translations, however, there is no guarantee that a source pronoun is translated correctly by the system. We therefore need to rely more heavily on manual analysis.

However, manual analysis can be aided by some automated pre-processing steps, to help select pronouns for further study. Using the source text and its translation together with word alignments output by the SMT systems, we can investigate which pronouns may be more difficult to translate than others — i.e. we can produce frequency distributions of the translations produced for each source pronoun surface-form (split by pronoun type).

4.1 Identifying Pronouns for Analysis

Examining the translation frequency distributions for the two state-of-the-art systems, we can observe the following. First, “it” can be translated into German, depending on the context, as either masculine singular (sg.), feminine sg. or neuter sg., or plural. As plural pronouns are not gendered, “they” has fewer translations. The possessive pronoun “its” has additional possible translation options due its multiple dependencies. That is, possessive pronouns in German must agree in number/gender with both the possessor and the object that is possessed. Different base forms are used depending on whether the possessor is feminine/plural (“ihre”) or masculine/neuter (“sein”). Other anaphoric pronouns such as “he” and “she” have far fewer translation options and are therefore less interesting. Based on the possible translation options, we selected (anaphoric) “it” and “its”.

Our analysis of manual translation (Section 3.3) showed that relativizers in English often corresponded to a relative pronoun inserted in the German translation. We wish to see how well SMT systems handle the translation of relativizers. We selected that-relativizers (explicit in English text) and null-relativizers (implicit). We exclude wh-
relativizers, also explicit, but with many forms (what, who, etc.), to reduce the annotation effort.

4.2 Pronoun Selection Task

Our manual analysis of pronoun translation is framed as a pronoun selection task. In this setting a human annotator is asked to identify which pronoun(s) could validly replace a placeholder masking a pronoun at a specific point in the SMT output. By masking the pronoun, we remove the risk that the annotator is biased by the pronoun present in the SMT output. The annotator’s selections may then be compared with the pronouns produced by the system in order to assess translation accuracy.

We used the tool described by Hardmeier (2014) for the pronoun selection task. The interface presents the annotator with the source sentence and its translation plus up to five previous sentences of history, as well as a number of pronoun options. The source pronoun in the final sentence of each example block is highlighted and its translation is replaced with a placeholder.

To determine how many sentences of history to present to the annotator (to help them identify the antecedent of an anaphoric pronoun), we used the manual annotations in ParCor. We calculated both the mean number of sentences between a pronoun and its antecedent, and two standard deviations from the mean (accounting for 95% of pronouns).

For the TED corpus the mean distance between pronoun and antecedent is 1.33 sentences, and two standard deviations from the mean is 4.95 sentences. For the EU Bookshop (whose sentences are longer), the distances between pronoun and antecedent are typically shorter, with a mean distance of 0.67 sentences and two standard deviations from the mean at 3.57 sentences. We nevertheless allow for up to five previous sentences of history for each example, regardless of genre.

4.3 Pronoun Selection Task: Guidelines

The following guidelines were adapted from those used by Hardmeier (2014) in order to cater for the requirements of English-German translation:

1) Select the pronoun that will create the most fluent translation, while preserving the meaning of the English sentence as much as possible. The latter means assigning correct number/gender to the pronoun that replaces the placeholder: Its case may be left “unknown”.

- If the SMT output is sufficiently fluent to be able to determine the case of the pronoun, select the appropriate check-box.
- Use the plural options if the antecedent is translated as a plural, or in any other scenarios in which a plural might seem appropriate.
- If different, equally grammatical options are available, select all appropriate check-boxes.

2) Alternatively select “Other” if the sentence should be completed with a pronoun not included in the list, “Bad translation” if a grammatical and faithful translation cannot be created without making major changes to the surrounding text, or “Discussion required” if you are unsure what to do.

3) Ignore minor disfluencies (e.g., incorrect verb agreement or obviously missing words).

4) Always try to select the pronoun that best agrees with the antecedent in the SMT output, even if the antecedent is translated incorrectly, and even if this forces you to violate the pronoun’s agreement with immediately surrounding words such as verbs, adjectives etc.

5) If the translation does not contain a placeholder, but a pronoun corresponding to the one marked in the English text should be inserted somewhere, indicate which pronoun should be inserted.

6) If the SMT output does not contain a placeholder, but already includes the correct pronoun, annotate the example as if a placeholder were present. This will mean selecting the same pronoun that is included in the SMT output.

4.4 Anaphoric “it”

The anaphoric pronoun “it” can co-refer either intra-sententially (i.e., to an antecedent in the same sentence) or inter-sententially (i.e., to an antecedent in a different sentence). While coreference imposes number–gender constraints on a pronoun and its antecedent, intra-sentential coreference imposes additional constraints.

We randomly selected 50 inter- and 50 intra-sentential tokens of “it” labelled anaphoric in the ParCor annotations. Tokens were selected from the TED Talks, as sentences there are typically shorter than those in the EU Bookshop and hence, potentially easier to work with. Additional guidelines are provided for “it”:

- Select “Pronominal adverb” if the most fluent translation would come from using a Ger-
man pronominal adverb. (Selection of the pronominal adverb is not required.)

- If a demonstrative pronoun (e.g. "diese" or "jene") is possible, select whether it is more or less likely than the personal pronoun(s).
- Genitive options are not available as these are used for possessives.

The annotator is presented with a table of options for number/gender and case combinations. The number/gender options are masculine, feminine, neuter and plural. The case options are: “case unknown”, and three German cases: nominative, accusative and dative. See Figure 1.

4.5 Anaphoric possessive “its”

In German, dependent possessive pronouns (i.e. those that precede a noun) must agree not only with the number/gender of its antecedent (possessor) but also with the number/gender of its object (i.e. the noun that follows the pronoun). For example in: “Der Staat und sein Einwohner” (“The state and its inhabitants”) the antecedent “Staat” (“state”) is masculine (sg.) and so a “sein” form is required for the possessive pronoun. The ending “e” in “sein” is needed because the noun following the possessive pronoun is plural (“Einwohner/inhabitants”).

We randomly selected 50 instances of “its” marked as anaphoric in ParCor. As “its” is uncommon in the TED corpus, all 50 instances came from the EU Bookshop corpus. Additional guidelines are provided for “its”:

- Select the relevant combination of number/gender of possessor and object. Select the case of the pronoun if the quality of the SMT output permits this.
- Select “Pronoun not required” if the translation does not require a pronoun.

The annotator is presented with a table of options capturing the number/gender of the possessor vs. the number/gender of the object. To reduce the number of options, a separate set of check-boxes is provided for case options, including “case unknown”, nominative, accusative, dative and genitive.

4.6 Relativizers

English relativizers may be explicit (that- and wh-relativizers), or implicit (null-relativizers). Both may be translated as relative pronouns in German.

We randomly selected 50 instances of relativizers from the TED corpus; 25 that- and 25 null-relativizers. The selection was semi-automatic, based on identifying relative clauses in the output of the Berkeley Parser (Petrov et al., 2006) and manually selecting those that contained a that- or null-relativizer.

As null-relativizers are implicit, there are no tokens in the English text to highlight. To keep this task in line with the others, we manually insert symbols for the nulls, i.e. the “iT” in “The house iT Jack built”, and (manually) align them to the corresponding token in the SMT output. (Unalignable tokens are left untranslated.) Instead of a pronoun in the English text, the annotator is presented with an instance of “that” or a symbol representing the null-relativizer. Placeholders are included in the translation as normal.

The options table captures pronoun number/gender and case. It is similar to the table for “it”, but with relative pronoun forms and options for “case unknown” and all four German cases.

5 Results

The results of the three pronoun selection tasks are presented in Table 4. We automatically compared the translations produced by the systems with the selections made by the annotator. If the system-generated pronoun matches one of the annotator’s selections, there is a “pronoun match”. If it doesn’t match any of the annotator’s selections or the system did not generate a pronoun there is a “pronoun mismatch”. Matches are recorded in terms of number/gender and case if the annotator supplied it, or number/gender only, if not.
When annotating the English side of ParCor, deciding whether a pronoun was anaphoric, event-

30
related or pleonastic was one of the major causes of annotator disagreement. It is therefore not surprising that problems might arise in identifying the pronoun’s antecedent for the pronoun selection task. This ambiguity did not arise for the “its” or relativizers tasks. With “its”, events are rarely (if ever) possessors and so rarely serve as antecedents. With relativizers, the relative pronoun and its antecedent (in German) are likely to be very close together, and certainly intra-sentential. The syntax-based system is much better at translating intra-sentential pronouns than inter-sentential ones. Although this system contained no such enhancements, one might expect that pronoun-aware syntax-based systems could be designed to leverage the fact that intra-sentential pronouns are syntactically governed, and produce better translations. One possible option would be to combine two systems: a phrase-based system to translate inter-sentential pronouns, and an enhanced syntax-based system to translate intra-sentential pronouns.

7 Discussion: Relative pronouns

When the antecedent is not a noun, i.e. “something” (“etwas”), “anything” (“alles” or “jedes” etc.) or “nothing” (“nichts”), “was” should be used:

(1) Now, when I use the term miracle, I don’t mean something that’s impossible.
(2) Nun, wenn ich den Begriff Wunder verwenden, ich meine nicht etwas, XXX ist unmöglich.

As “was” is not provided as an option in the pronoun selection task, the annotator marked example 2 (and others like it) as “other”. SMT systems must decide whether to use a relative pronoun that conveys the number/gender of the antecedent (i.e. “der/die/das”) or “was/wer/wo” (if the antecedent cannot be determined / there is no antecedent). As this decision depends on the antecedent, relative pronouns may therefore be treated as a more localised sub-set of anaphoric pronouns.

The translation of relativizers may require a preposition preceding the relative pronoun:

(3) That’s the planet we live on.
(4) Das ist die Welt, XXX wir leben.

The correct translation of example 3, which contains a null-relativizer (indicated by ©), would be “Das ist die Welt, in der wir leben”. However, in the SMT output the preposition “in” is missing, and so the annotator was required to select the correct pronoun as if the preposition had been present.

In German, the choice of preposition and case of the pronoun are determined by the verb of the clause. As these choices are connected, SMT systems could also consider the translation of prepositions when translating relative pronouns.

8 Conclusion

The analysis of manual translation revealed that pronouns are frequently dropped and inserted by human translators and that German translations contain many more pleonastic and anaphoric pronouns than the original English texts. Both of these differences can result in SMT systems learning poor translation mappings.

The analysis of state-of-the-art translation revealed that biases in the training data and incorrect selections of the base form pronoun (i.e. “ihr” vs. “sein” for “its”) are both problems which SMT systems must overcome. For relative pronouns selecting the correct preposition is also important as it influences the case of the pronoun.

9 Future Work

Possible directions for future work include further analyses of manual and automated translation and applying the knowledge that is gained to build pronoun-aware SMT systems. Initial efforts could focus on syntax-based SMT — leveraging information within target-side syntax trees constructed by the decoder, to encourage pronoun-antecedent agreement for intra-sentential anaphoric pronouns (i.e. “it/its” and relative pronouns). Pronoun-aware SMT systems could also address translation of the ambiguous second-person pronouns “you” and “your”. In English, they have both deictic and generic use, while in German, different forms are used (“Sie/du” vs. “man”).

Acknowledgements

Thanks to Christian Hardmeier for providing the pronoun-selection tool, to Susanne Tauber for completing the annotations and to Felix Suessenbach and Sam Gibbon for assisting with the significance testing. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 645452 (QT21).
References


A.19 PROTEST: A Test Suite for Evaluating Pronouns in Machine Translation

PROTEST: A Test Suite for Evaluating Pronouns in Machine Translation
Liane Guillou and Christian Hardmeier
University of Edinburgh, University of Uppsala
L.K.Guillou@sms.ed.ac.uk, christian.hardmeier@lingfil.uu.se

Abstract
We present PROTEST, a test suite for the evaluation of pronoun translation by MT systems. The test suite comprises 250 hand-selected pronoun tokens and an automatic evaluation method which compares the translations of pronouns in MT output with those in the reference translation. Pronoun translations that do not match the reference are referred for manual evaluation. PROTEST is designed to support analysis of system performance at the level of individual pronoun groups, rather than to provide a single aggregate measure over all pronouns. We wish to encourage detailed analyses to highlight issues in the handling of specific linguistic mechanisms by MT systems, thereby contributing to a better understanding of those problems involved in translating pronouns. We present two use cases for PROTEST: a) for measuring improvement/degradation of an incremental system change, and b) for comparing the performance of a group of systems whose design may be largely unrelated. Following the latter use case, we demonstrate the application of PROTEST to the evaluation of the systems submitted to the DiscosMT 2015 shared task on pronoun translation.

Keywords: Evaluation, pronouns, machine translation

1. Motivation
In most current statistical machine translation (SMT) methods, output words are generated in correspondence with the input words, according to word-alignments found at training time. In addition to word-alignments, only very limited context information is taken into account in the generation process. While the approach works well for content words, it does not for function words, such as pronouns and negation markers, which are critical to meaning (Hardmeier et al., 2015; Hardmeier et al., 2013; Novák et al., 2013; Guillou, 2012).

Pronouns have different functions, and their use varies between languages. Some pronouns function as referring elements, creating a link to an element occurring elsewhere in the discourse. Others are simply to ensure a grammatical sentence. For example pleonastic pronouns, such as the “it” in “It is raining” or “il” in “Il pleut”, are used to fill the subject position. In many languages, pronouns are morphologically marked for categories such as gender and number, subject to certain agreement constraints that must be satisfied according to the rules of the target language. This is mostly a problem for referring pronouns, where generating the correct form requires identifying what the pronoun refers to (anaphora resolution).

Evaluation poses a particular problem for researchers interested in pronoun generation in machine translation (MT). Owing to the cost and difficulty of manual evaluation (including manual post-editing based methods as a means to assess MT quality), MT researchers rely on automatic evaluation metrics such as BLEU (Papineni et al., 2002) to guide their development efforts. Most automatic metrics assume that overlap of the MT output with a human-generated reference translation may be used as a proxy for correctness. In the case of anaphoric pronouns, this assumption breaks down. If the pronoun’s antecedent is translated in a way that differs from the reference translation, a different pronoun may be required: One that matches the reference translation may in fact be wrong. This shortcoming of existing automatic evaluation metrics is widely recognised (Le Nagard and Koehn, 2010; Hardmeier and Federico, 2010; Guillou, 2011), but so far, no viable alternatives have been proposed. Hardmeier and Federico (2010) suggest using a precision/recall-based measure that is more sensitive to pronouns than general-purpose metrics. However, this metric scales the fundamental shortcomings of all reference-based metrics, and its correlation with human judgements on pronoun correctness is weak (Hardmeier et al., 2015). In view of these difficulties, Hardmeier (2015) suggests using a test suite composed of carefully selected pronoun tokens which can be checked individually using an automatic evaluation script, instead of an aggregate measure over a complete test set, to evaluate pronoun correctness.

2. Overview
The PROTEST test suite comprises 250 hand-selected pronoun tokens and an automatic evaluation script which compares the output of an MT system against a reference translation. Pronoun tokens are categorised according to problems that MT systems face when translating pronouns, and the set is small enough to allow for manual evaluation and inspection of translations that do not match the reference. The test suite facilitates the efficient evaluation and inspection of the translation of individual pronoun tokens. Through the identification of interesting examples and problems, we believe that researchers will be better able to focus the design of their MT systems, and ultimately improve the current state-of-the-art in the field.

3. Related Work
Previous approaches to pronoun translation evaluation include the automatic precision/recall-based measure of
Hardmeier and Federico (2010), the (manual) pronoun selection task used in the DiscoMT 2015 shared task evaluation (Hardmeier et al., 2015) and methods based on manual counting (Le Nagard and Koehn, 2010; Guillou, 2012; Novák et al., 2013).

The ACT metric (Hajlaoui and Popescu-Belis, 2013), for the automatic evaluation of discourse connectives, bears some resemblance to the work in this paper. ACT automatically compares the translation of discourse connectives in MT output with those in the reference translation. Those that match are deemed correct and those that do not are referred for manual evaluation. The translations in the reference are augmented with additional acceptable translations provided in the form of a list. Unlike for PROTEST, no test suite of hand-selected examples is provided for ACT. The assessment of discourse connective translations is also more straightforward; agreement constraints such as the pronoun–antecedent agreement for anaphoric pronouns, do not apply.

4. Test Set Annotations

We build the test suite on top of an existing corpus: The DiscoMT2015.test dataset (Hardmeier et al., 2016) created for the shared task on pronoun translation at the Second Workshop on Discourse in Machine Translation (Hardmeier et al., 2015). This test set contains English transcriptions of 12 TED conference talks (and their French translations), selected in such a way that the texts include a reasonable number of instances of some less frequent pronoun types. Since we provide complete texts, rather than a collection of isolated sentences or passages, any MT system being tested has access to full document context for each pronoun token, which is essential for discourse-enabled translation.

The English source texts were annotated manually for reducing coreference in the style of the ParCor corpus (Guillou et al., 2014). These annotations form the basis for our categorisation and selection of pronoun tokens, and the evaluation procedure. Pronouns are annotated according to the principal functional categories (types) of ParCor.

There are three types of pronominal reference: Anaphoric, event and extra-textual reference.

Anaphoric pronouns are the most typical case. They refer to an entity mentioned earlier, typically in the form of a noun phrase (NP), in the discourse. The mention referred to is called the pronoun’s antecedent. Consider Ex. 1, in which the anaphoric pronoun “it” refers to “bicycle” (its antecedent):

(1) I have a bicycle. **It** is red.

*Event reference* pronouns also have a referring function, but their antecedents are not entities, but propositions, facts, states, situations, opinions, etc. For example, the pronoun “it” in Ex. 2 refers to the event of X invading Y.

(2) X invaded Y. **It** resulted in war.

Pronouns with *extra-textual reference* do not have an antecedent in the text, but refer to an element in the situational context of the utterance such as an overhead slide or an object. For example, during a TED Talk, the speaker may point to a slide and say “Look at this”, where the entity to which the pronoun refers is not explicitly mentioned (and therefore does not appear in the transcript text).

*Pleonastic* pronouns, by contrast, are non-referring pronouns used to satisfy the grammar of the target language, but without semantic function. For example, the “it” in “It is raining” does not refer to anything.

Finally, *speaker reference* and *addressee reference* are used for first- and second-person pronouns referring to the discourse participants or to generic agents. Speaker reference pronouns refer to the speaker and include “I, me, one” etc. Addressee reference pronouns can refer to an individual person, a group of people or to people in general, and include the pronouns “you” and “your(s)”. The annotations include features specific to pronoun function, referred to as type in ParCor. For example, anaphoric pronouns are linked to their nominal antecedents, and instances of anaphoric “it” are marked as subject vs. non-subject position. Addressee reference pronouns are marked as deictic vs. generic. Deictic instances refer to a specific person or group and generic instances refer to people in general (e.g. “In England, if you own a house **you** have to pay taxes”).

As in ParCor, full coreference chains are not annotated, but rather each anaphoric pronoun is simply linked to its closest non-pronominal antecedent, if one exists. Whilst gold-standard test sets exist for the coreference resolution task, they are not suitable for assessing machine translation. In particular, monolingual gold-standard test sets lack reference translations, and there exist neither monolingual nor multi-lingual test sets that provide the additional pronoun type-specific features used to define the fine-grained categories for the test suite pronouns.

5. Test Suite Design

The test suite comprises a set of pronoun tokens and their reference translations, and a script to automatically evaluate pronoun translation in MT output. Those pronoun translations that do not match the reference are referred for manual evaluation and inspection. It is this need for manual evaluation that motivates the use of a hand-selected set of pronoun tokens, as opposed to the complete set of pronouns in the DiscoMT2015.test dataset. 250 pronoun tokens were selected for the test suite, according to the selection criteria outlined in Section 5.1. The methodology for the automatic evaluation of the pronoun tokens is described in Section 5.2. Manual inspection of individual pronoun translations can be used to identify what might have gone wrong in the translation, or systematic mistakes by an MT system.

5.1. Selection of Pronoun Tokens

The distribution of pronoun types in DiscoMT2015.test is presented in Table 1. The anaphoric and cataphoric types have been sub-split into *intra-sentential* (pronoun and antecedent appear in the same sentence) and *inter-sentential* (pronoun and antecedent appear in different sentences). For anaphoric pronouns, two additional sub-types are considered: Those linked to another pronoun (no NP antecedent
was found) and those with no specific antecedent, e.g. “In this study they took 100 people and split them into two groups”, where the antecedent of “they” is implicitly signalled by the nearby noun (“study”). As pronoun-antecedent agreement must hold in French, the translation accuracy of such pronouns would be difficult to assess.

<table>
<thead>
<tr>
<th>Pronoun type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anaphoric</td>
<td></td>
</tr>
<tr>
<td>inter-sentential</td>
<td>761</td>
</tr>
<tr>
<td>intra-sentential</td>
<td>644</td>
</tr>
<tr>
<td>linked to another pronoun</td>
<td>26</td>
</tr>
<tr>
<td>no specific antecedent</td>
<td>93</td>
</tr>
<tr>
<td>Cataphoric</td>
<td>8</td>
</tr>
<tr>
<td>Event</td>
<td>360</td>
</tr>
<tr>
<td>Extra-textual reference</td>
<td>110</td>
</tr>
<tr>
<td>Pleonastic</td>
<td>123</td>
</tr>
<tr>
<td>Speaker reference</td>
<td>1,880</td>
</tr>
<tr>
<td>Addressee reference</td>
<td>727</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>4,732</strong></td>
</tr>
</tbody>
</table>

Table 1: Pronoun distribution by type for the DiscoMT2015.test dataset

Our aim is to extract pronoun tokens that provide good coverage over the range of different pronoun types and surface forms (e.g. “it”, “they” etc.) and represent the different problems that MT researchers must consider:

- Anaphoric (it/they)
  - Inter-sentential vs. intra-sentential
  - Subject vs. non-subject (it only)
  - Singular vs. plural “they”
  - Referring to group nouns (e.g. “company” could be referred to as singular/plural)
- Event (it)
- Pleonastic (it)
- Addressee Reference [you]
  - Generic vs. deictic
  - Singular vs. plural [deictic only]

At the top level, we distinguish between those pronoun forms whose multiple functions in English require different translations in French. For example, the ambiguous pronoun “it” can be anaphoric, requiring pronoun-antecedent agreement in terms of number and gender. It can also be pleonastic or event reference, with no agreement constraints, but requiring the use of different French pronouns. At the lower level, we consider differences exhibited by pronouns of the same type and form. This applies to anaphoric and addressee reference pronouns.

For anaphoric pronouns we distinguish between inter- and intra-sentential pronouns, which given the current framework of sentence-by-sentence translation, pose different challenges to MT systems. From a grammatical perspective, intra-sentential coreference has additional constraints to inter-sentential coreference. We also consider position and number. For example, different French pronouns will be required when translating subject vs. non-subject position instances of “it”. Translating plural vs. singular “they” (a gender-neutral alternative to “he/she” in English), requires different pronouns again.

For addressee reference pronouns, we consider ambiguity caused by both deictic and generic use of the pronoun “you”. For deictic instances, number affects the French translation: “tu” or “vous” may be used to refer to a single person (depending on formality), but when referring to more than one person “vous” must be used. Generic “you” may be translated as “on” (similar to English “one”).

We achieve a balance both in terms of the number of pronoun tokens for each category, and of the expected French translation. The overall number of pronoun tokens selected for each category is related to the number of potential ways in which the (English) pronoun may be translated in French. Within each category, we have tried to balance the selection of individual pronoun tokens based on their translation in the reference. For example, we have selected equal numbers of instances of “it/they” that we might expect to be translated as masculine vs. feminine pronouns (by looking at the reference translation). We have also considered instances of singular pronouns that may be translated as plural in French and vice versa. Some categories, such as anaphoric singular “they” occur infrequently in the DiscoMT2015.test dataset. The number of pronoun tokens selected for such categories is therefore small.

Another option would be to define category sizes in proportion to the number of pronoun tokens for each category in the source-language texts. However, if we wish to build MT systems that are linguistically competent, they should demonstrate an understanding of the linguistic system, rather than mere frequencies. Our aim is to be able to assess the accuracy of an MT system in translating both commonly occurring source-language pronouns and rare ones (e.g. singular “they”).

One use case for the test suite is to complement automatic evaluation with manual evaluation. This motivates the restriction of the set of pronoun tokens to a number that is manageable for manual evaluation and inspection. We therefore exclude a number of pronoun groups, for which we have very few instances in DiscoMT2015.test or for which we believe translation is less problematic. The following pronoun groups are excluded from the test suite:

- Reflexive pronouns, which are very infrequent in TED talks.
- Relative pronouns. Those that are marked for number and gender in French (e.g. “lequel” [masc. sing.], “lesquelles” [fem. pl.], etc.) are infrequent in TED talks, and those that are not marked (e.g. “qui”, “que”, “dont” and “quoi”), are unambiguous as they are in English.
- First-person (i.e. speaker reference) pronouns, and the third-person pronouns “his/her” which are all unambiguous in English.

Notes:

1. “ce” may function as both an event or pleonastic pronoun; “il” may be used as both a pleonastic or anaphoric pronoun
5.2. Automatic Evaluation

We provide an automatic script to check the translations of the test suite pronoun tokens in the output of an MT system. For anaphoric pronouns, the script verifies that both the translation of the pronoun and the antecedent head match those in the reference translation. For all other pronoun types, only the translation of the pronoun is considered. Matches are measured in terms of overlap between the reference token and the MT output string. The evaluation script outputs the count of pronoun tokens correctly translated by the MT system (i.e., "matches"), for each category, as well as an accuracy score for each category and for the test suite as a whole. The tokenisation of the source text is relevant to evaluation and systems may tokenise the source text in ways other than that in DiscoMT2015.test. It is therefore necessary to supply the tokenised source text in addition to the MT output and the word-alignments between the source text and MT output. The sentence-internal word-position of each pronoun token (and antecedent head where relevant), and its MT translation are identified. Whilst the accuracy score output by the evaluation script can be used as an aggregate metric, the main advantage of the test suite over existing metrics is the possibility to study the system's performance on individual pronoun tokens.

6. Use Cases

There are two main use cases for which PROTEST was designed. The first is for the manual evaluation of those translations that did not match the reference in the automatic evaluation. By combining automatic and manual evaluation, we are able to obtain a complete evaluation of one or more systems. In addition to the number of matches for each pronoun category, the evaluation script outputs a list of mismatches between the MT and reference translations to be checked manually – the pronoun translations (and antecedent heads) may be valid alternative translations of the source, not present in the reference. Consider the following example:

(3) I have a bicycle. It is red.

(4) J'ai un vélo. Il est rouge. [reference]

(5) J'ai une bicyclette. Elle est rouge. [MT output]
The results reveal, subject to confirmation following

Table 3: Matches per category for the DiscoMT 2015 shared task

<table>
<thead>
<tr>
<th>anaphoric</th>
<th>event</th>
<th>pleonastic</th>
<th>addressee reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>intra</td>
<td>inter</td>
<td>intra</td>
</tr>
<tr>
<td>Baseline</td>
<td>8</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>auto-postEDit</td>
<td>10</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>UU-Hardmeier</td>
<td>8</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>IDIAP</td>
<td>5</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>UU-Tiedemann</td>
<td>9</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>A3-108</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Here the English anaphoric pronoun “it” in Ex. 3 refers to “bicycle”. The reference translation (Ex. 4) translates “it” as “ils” (masc. sing.) which agrees with the translation of “ bicyclette” ("vélo" [masc. sing.]). In the MT output (Ex. 5), a valid alternative translation is produced, with “elle” referring to “ bicyclette” (both fem. sing.). This translation, although correct, does not match the reference and would therefore be referred for manual evaluation.

This need for manual evaluation is the driving factor behind restricting the test suite to only a sub-set of the pronouns in DiscoMT2015 test.

During development, translations found in the MT output could be added to the set of translations accepted by the evaluation script once they have been manually verified for correctness. Obviously, doing so will make it impossible to compare the scores output by the evaluation scripts with values reported by other groups, but it enables a more precise evaluation of progress for the developer’s internal use.

The second use case is for the measurement of the incremental progress of a system (or systems), where it is important to be able to compare the scores output by the evaluation scripts with values reported by other groups, for example where a new system extends a baseline, or provides a small incremental change over an existing system. However, it produces some pronoun translations that are better than those produced by the baseline. For example, the IDIAP system translates “corporations” and “they” as “les entreprises” and “elles” (Ex. 8) as per the reference (Ex. 7), but the baseline system provides a non-matching (and incorrect) translation of the pronoun (“ils” [masc. pl.] does not agree with “entreprises” [fem. pl.]).

6. You are one of those people who believe that corporations are an agent of change if they are run well. [Source]

7. Vous êtes l’une de ces personnes qui croient que les entreprises sont des agents de changement si elles sont bien dirigées [Reference]

8. vous êtes de ceux qui croient que les entreprises sont un agent de changement, si elles sont bien gérées . [IDIAP]

9. Vous êtes de ceux qui croient que les entreprises sont un agent de changement s’ils sont bien gérés . [Baseline]

Knowing the design of the DiscoMT 2015 systems is also useful when interpreting results. This information can be found in the system description papers, which are available for all systems except A3-108. One pattern that can be observed is that the auto-postEDit (Guillou, 2015) and ITS2 (Loaiciga and Wehrli, 2015) systems both perform particularly poorly for the event and pleonastic categories and this
may be due to design similarities for these systems. Both systems make use of rules; ITS2 is a rule-based MT system and auto-postEDIt uses rules to automatically post-edit the output of a baseline phrase-based SMT system. In addition, the focus of both systems is on producing gendered pronoun translations. The auto-postEDIt system uses a simple rule to replace the translations of non-anaphoric pronouns that do not match a predefined set with the token “ce”. The ITS2 system ignores the problem of translating event reference and plonemonic pronouns altogether. Evidently these strategies will be beaten by more sophisticated approaches such as those provided by some of the other systems. This is reflected in the results in Table 3.

Another clear pattern is the similarity in performance of UU-Tiedemann (Tiedemann, 2015) and the baseline system. Both are phrase-based SMT systems trained using the same data. In contrast to the other systems, the UU-Tiedemann system does not attempt to resolve pronominal anaphora explicitly. Instead, it uses a cross-sentence n-gram model over determiners and pronouns which aims to bias the SMT model towards selecting correct pronouns. In many ways it could be considered the system closest in design to that of the baseline. The systems generally performed well on the translation of adereço reference “you”, as compared with the baseline. However, none of the systems was designed with the aim of handling adereço reference pronouns, given that the focus of the shared task was on translating instances of “it” and “they”.

8. Extending PROTEST

8.1. Extension to Other Language Pairs

The pronoun test suite was developed with English-to-French translation in mind, and the pronoun tokens are for this language pair. However, the method described in this paper could be applied for other language pairs. The underlying methodology of the automatic evaluation script is language independent and pronoun token sets may be extracted for any language pair, using a similar method to that described in Section 5.1. Translations of the DiscoMT2015.test dataset exist for many other languages. It is therefore possible to extend the test suite to cover those other target languages with little additional effort. The same ParCor-style annotations over the English source texts of the DiscoMT2015.test dataset may be used. However, depending on the language pair in question, different pronoun categorisations may be appropriate. Based on the functional ambiguity of pronouns in the source language, i.e. ambiguity arising from the same surface form pronoun having many functions, different categorisations may be required to make accurate distinctions between pronoun tokens. For example, Section 5.1 outlines the need to disambiguate uses of the English pronoun “it”. In addition to this, the translation frequencies of the source-language pronouns should be considered as it is expected that a pronoun with multiple translation options in the target language would be more difficult to translate than one with only a single option. When considering other target languages, the need for additional annotation over the DiscoMT2015.test dataset may arise. The annotation of additional features, however, need not interfere with the existing annotations.

Additionally, the ParCor annotation guidelines may be used to annotate texts for source languages other than English and/or for different genres. We recommend the use of ParCor-style annotations over the source-language text in order to identify pronouns which exhibit functional ambiguity, and other features which may be useful in categorising pronoun tokens.

8.2. Using Multiple Reference Translations

The DiscoMT2015.test dataset contains a single reference translation from which the gold-standard translation is extracted for each pronoun token in the test suite. Pronoun translations in the MT output are automatically compared with those in the reference, and those that do not match are referred for manual evaluation. As manual evaluation is costly, consideration should be given to methods for reducing this effort. One possibility would be to use multiple reference translations, which may provide a number of valid alternative translations for a given pronoun, or in the case of anaphoric pronouns, alternative pronoun-antecedent pairs. Consider the following English-French example from Hardmeier (2014):

(10) The funeral of the Queen Mother will take place on Friday. It will be broadcast live.

(11) Les funérailles de la reine-mère auront lieu vendredi. Elles seront retransmises en direct. [Reference 1]

(12) L’enterrement de la reine-mère aura lieu vendredi. Il sera retransmis en direct. [Reference 2]

Ex. 11 and Ex. 12 are two valid (French) reference translations of Ex. 10. Using both reference translations, the following pronoun-antecedent pairs may be extracted: “Elles”-“funérailles” (”funeral”) and “Il”-”enterrement” (”burial”). When evaluating the performance of an English-French MT system on translating Ex. 10, the automatic evaluation script would look to match the pronoun-antecedent translations in the MT output with either the French translation pair extracted from Ex. 11 or Ex. 12. Differences across multiple reference translations may exist for any target language. In the example above, variation in the reference translation arises from choosing different translations of the English antecedent head and selecting a pronoun with the appropriate gender. This is not the only reason for the use of different pronouns in reference translations that all convey the same meaning. Consider the following English examples:

(13) [You/One] should always tell the truth.
(14) I got the hiccups when I drank Champagne. [This/It] happened again when I drank sparkling cider.

In Ex. 13, the generic pronouns “You” and “One”, may be used interchangeably without altering the meaning of the text. So too in Ex. 14, the pronouns “This” and “It” can both be used to provide the same meaning.
Multiple reference translations do not exist for TED Talks, which have only a single official English transcript and a single official translation for each target language. The manual creation of additional translations for the Disc-oMT2015.test dataset provides one option. Another is to make use of manual annotation over the output of MT systems to provide alternative valid translations, as described in Section 6. For non-anaphoric pronouns, the set of valid alternative translations would be pronoun translations. For anaphoric pronouns, the valid alternatives would be pronoun-antecedent pair translations. These alternative translations, collected over time, could then be used as silver-standard translations in the automatic evaluation of the output of new MT systems.

9. Conclusions and Future Work
The test suite is intended to support developers in evaluating the performance of MT systems on the task of pronoun translation. The set of pronoun tokens covers a range of different pronoun types and forms, tailored to the problems that challenge MT. We have released the test suite – the set of pronoun tokens and automatic evaluation script. The test suite was designed for the English to French translation direction, but the methodology is language-independent. Pronoun token sets may be extracted for other language pairs for which ParCor-style annotation is provided. Depending on the language pair different pronoun categorisations may be appropriate.

To support manual evaluation of pronoun tokens that are not correctly translated per the reference (i.e. mismatches), we propose development of a graphical user interface (GUI) for browsing the test suite translations in context. The GUI would also allow for pronoun-antecedent pairs not present in the reference translation but valid alternatives, to be added to the set of acceptable translations. The GUI would serve as a tool to be used both by annotators carrying out manual evaluation tasks, and by researchers wishing to better understand how their systems perform.

A project is already underway to develop the GUI and conduct the manual evaluation of the output of the DiscoMT 2015 shared task systems. We hope to release both the GUI and manual evaluation results in the near future.

10. Acknowledgements
We would like to thank Sum Gibbon for manually annotating the Disc-oMT2015.test dataset, Bonnie Webber for her many useful comments and suggestions, and the three anonymous reviewers.

This work was funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement n° 645452 (QT21), and the Swedish Research Council under project 2012-916 Discourse-Oriented Statistical Machine Translation. The manual annotation of the Disc-oMT2015.test dataset was funded by the European Association for Machine Translation (EAMT).

11. Bibliographical References


Le Nagard, R. and Koehn, P. (2010). Aiding pronoun...
Edinburgh’s Statistical Machine Translation Systems for WMT16

Philip Williams¹, Rico Sennrich¹, Maria Nádejde¹, Matthias Huck², Barry Haddow₁, Ondřej Bojar³
¹School of Informatics, University of Edinburgh
²Center for Information and Language Processing, LMU Munich
³Institute of Formal and Applied Linguistics, Charles University in Prague
pwillia4@inf.ed.ac.uk rico.sennrich@ed.ac.uk M.Nadejde@sms.ed.ac.uk
mhuck@cis.lmu.de bhaddow@inf.ed.ac.uk bojar@ufal.mff.cuni.cz

Abstract

This paper describes the University of Edinburgh’s phrase-based and syntax-based submissions to the shared translation tasks of the ACL 2016 First Conference on Machine Translation (WMT16). We submitted five phrase-based and five syntax-based systems for the news task, plus one phrase-based system for the biomedical task.

1 Introduction

Edinburgh’s submissions to the WMT 2016 news translation task fall into two distinct groups: neural translation systems and statistical translation systems. In this paper, we describe the statistical systems, which includes a mix of phrase-based and syntax-based approaches. We also include a brief description of our phrase-based submission to the WMT16 biomedical translation task. Our neural systems are described separately in Sennrich et al. (2016a).

In most cases, our statistical systems build on last year’s, incorporating recent modelling refinements and adding this year’s new training data. For Romanian—a new language this year—we paid particular attention to language-specific processing of diacritics. For English→Czech, we experimented with a string-to-tree system, first using Treex¹ (formerly TectoMT; Popel and Žabokrtský, 2010) to produce Czech dependency parses, then converting them to constituency representation and extracting GHKM rules.

In the next two sections, we describe the phrase-based systems, first describing the core setup in Section 2 and then describing system-specific extensions and experimental results for each individual language pair in Section 3. We describe the core syntax-based setup and experiments in Sections 4 and 5.

2 Phrase-based System Overview

2.1 Preprocessing

The training data was preprocessed using scripts from the Moses toolkit. We first normalized the data using the normalize-punctuation.perl script, then performed tokenization (using the -a option), and then truecasing. We did not perform any corpus filtering other than the standard Moses method, which removes sentence pairs with extreme length ratios, and sentences longer than 80 tokens.

2.2 Word Alignment

For word alignment we used fast_align (Dyer et al., 2013)—except for German→English, where we used MGIZA++ (Gao and Vogel, 2008)—followed by the standard grow-diag-final-and symmetrization heuristic.

2.3 Language Models

Our default approach to language modelling was to train individual models on each monolingual corpus (except CommonCrawl) and then linearly-interpolate them to produce a single model. For some systems, we added separate neural or CommonCrawl LMs. Here we outline the various approaches and then in Section 3 we describe the combination used for each language pair.

Interpolated LMs

For individual monolingual corpora, we first used lmplz (Heafield et al., 2013)—except for German→English, where we used MGIZA++ (Gao and Vogel, 2008)—followed by the standard grow-diag-final-and symmetrization heuristic.
using weights tuned to minimize perplexity on the development set.

**CommonCrawl LMs** Our CommonCrawl language models were trained in the same way as the individual corpus-specific standard models, but were not linearly-interpolated with other LMs. Instead, the log probabilities of CommonCrawl LMs were added as separate features of the systems’ linear models.

**Neural LMs** For some of our phrase-based systems we experimented with feed-forward neural network language models, both trained on target n-grams only, and on “joint” or “bilingual” n-grams (Devlin et al., 2014; Le et al., 2012). For training these models we used the NPLM toolkit (Vaswani et al., 2013), for which we have now implemented gradient clipping to address numerical issues often encountered during training.

### 2.4 Baseline Features

We follow the standard approach to SMT of scoring translation hypotheses using a weighted linear combination of features. The core features of our model are a 5-gram LM score (i.e. log probability), phrase translation and lexical translation scores, word and phrase penalties, and a linear distortion score. The phrase translation probabilities are smoothed with Good-Turing smoothing (Foster et al., 2006). We used the hierarchical lexicalized reordering model (Galley and Manning, 2008) with 4 possible orientations (monotone, swap, discontinuous left and discontinuous right) in both left-to-right and right-to-left direction. We also used the operation sequence model (OSM) (Durrani et al., 2013) with 4 count based supportive features. We further employed domain indicator features (marking which training corpus each phrase pair was found in), binary phrase count indicator features, sparse phrase length features, and sparse word source deletion, target word insertion, and word translation features (limited to the top $K$ words in each language, typically with $K = 50$).

### 2.5 Tuning

Since our feature set (generally around 500 to 1000 features) was too large for MERT, we used $k$-best batch MIRA for tuning (Cherry and Foster, 2012). To speed up tuning we applied threshold pruning to the phrase table, based on the direct translation model probability.

### 2.6 Decoding

In decoding we applied cube pruning (Huang and Chiang, 2007) with a stack size of 3000 (reduced to 1000 for tuning). Minimum Bayes Risk decoding (Kumar and Byrne, 2004), a maximum phrase length of 5, a distortion limit of 6, 100-best translation options and the no-reordering-over-punctuation heuristic (Koehn and Haddow, 2009).

### 3 Phrase-based Experiments

#### 3.1 Finnish→English

Similar to last year (Haddow et al., 2015), we built an unconstrained system for Finnish→English using data extracted from OPUS (Tiedemann, 2012). Our parallel training set was the same as we used previously, but the language model training set was extended with the addition of the news2015 monolingual corpus and the large WMT16 English CommonCrawl corpus. We used newsdev2015 for tuning, and newsdev2015 for testing during system development.

One clear problem that we noted with our submission from last year was the large number of OOVs, which were then copied directly into the English output. This is undoubtedly due to the aglutinative nature of Finnish, and probably was the cause of our system being poorly judged by human evaluators, despite having a high BLEU score. To address this, we split the Finnish input into sub-word units at both train and test time. In particular, we applied byte pair encoding (BPE) to split the Finnish source into smaller units, greatly reducing the vocabulary size. BPE is a technique which has been recently used to good effect in neural machine translation (Sennrich et al., 2016b), where the models cannot handle large vocabularies. It is actually a merging algorithm, originally designed for compression, and works by starting with a maximally split version of the training corpus (i.e. split to characters) and iteratively merging common clusters. The merging continues for a specified number of iterations, and the mergers are collected up to form the BPE model. At test time, the recorded merges are applied to the test corpus, with the result that there are no OOVs in the test data. For the experiments here, we used 100,000 BPE merges to create the model.

Applying BPE to Finnish→English was clearly effective at addressing the unknown word problem, and in many cases the resulting translations
are quite understandable, e.g.

source yös Intian on sanottu olevan kiinnostunut
puolustusyhteistyösopimuksesta Japanin
kanssa.
base India is also said to be interested in puolustusyhteistyösopimuksesta with Japan.

bpe India is also said to be interested in defence cooperation agreement with Japan.

reference India is also reportedly hoping for a deal on defence collaboration between the two nations.

However applying BPE to Finnish can also result in some rather odd translations when it overzealously splits:

source Balotelli oli vielä kaukana huippu-
vireestään.
base Balotelli was still far from huippuvireestään.

bpe Baloo, Hotel was still far from the peak of its vitality.

reference Balotelli is still far from his top tune.

We built four language models: an interpolated count-based 5-gram language model with all corpora, apart from the WMT16 CommonCrawl; separate count-based language models with WMT16 CommonCrawl and news2015; and a neural LM on news2015. A performance comparison across different language model combinations, and with and without BPE is shown in Table 1.

<table>
<thead>
<tr>
<th>system</th>
<th>BLEU</th>
<th>fi-en</th>
<th>re-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>only interpolated LM</td>
<td>22.9</td>
<td>34.2</td>
<td></td>
</tr>
<tr>
<td>+ CommonCrawl LM</td>
<td>23.2</td>
<td>35.0</td>
<td></td>
</tr>
<tr>
<td>+ CC LM &amp; news2015 (count)</td>
<td>23.4</td>
<td>34.9</td>
<td></td>
</tr>
<tr>
<td>+ CC LM &amp; news2015 (neural)</td>
<td>23.4</td>
<td>35.2</td>
<td></td>
</tr>
<tr>
<td>+ all</td>
<td>23.4</td>
<td>35.0</td>
<td></td>
</tr>
<tr>
<td>without BPE</td>
<td>22.2</td>
<td>–</td>
<td>32.2</td>
</tr>
</tbody>
</table>

Table 1: Comparison of different language model combinations and preprocessing regimes for Finnish→English and for Romanian→English. The submitted system is shown in bold. The preprocessing variant uses the same language model combination as the submitted system. Cased BLEU scores are on newsdev2016.

3.2 Romanian→English

We trained our Romanian→English system using all data available for the constrained task. For system development, we split the newsdev2016 set into two parts randomly (so as to balance the “born English” and “born Romanian” portions), using one for tuning and one for testing. For building the final system, and for the contrastive experiments, we used the whole of newsdev2016 for tuning, and newsdev2016 for testing.

In early experiments we noted that both the training and the development data were inconsistent in their use of diacritics leading to problems with OOVs and sparse statistics. To address this we stripped off all diacritics from the Romanian texts and the result was a significant increase in performance in our development setup. We also experimented with different language model combinations during development, with our submitted system using three different language model features: a neural LM trained on just news2015 monolingual, an n-gram language model trained on the WMT16 English CommonCrawl corpus, and a linear interpolation of language models trained on all other WMT16 English corpora.

In Table 1 we show how system performance varies under different language model combination and preprocessing conditions.

3.3 English→Romanian

For English→Romanian, we used all the data in the constrained track, including the CommonCrawl language model data, and as with the Romanian→English system, we used newsdev2016 for the final tuning run.

The inconsistent use of diacritics in Romanian text also affected the English→Romanian system, however removing altogether would be problematic as we would then need a method for restoring them for the final system. So the only extra preprocessing we performed on the Romanian was to ensure that “t-comma” and “s-comma” were written correctly, with a comma rather than a cedilla.

Our final system used two different count-based 5-gram language models (one trained on all data, including the WMT16 Romanian CommonCrawl corpus, without pruning, and one trained on news2015 monolingual only), a neural language model trained on news2015 monolingual, and a bilingual language model trained on the parallel data, with source window of 15 and target window of 1. In Table 2 we show ablation experiments where we remove each of these language models.
Table 2: Effect of each of the language models used in the English→Romanian system. The experiments are not cumulative, so we first try pruning the “all” language model, then go back to the unpruned version and remove each LM in turn, observing the effect. The submitted system used all four LMs, and the scores shown are uncased BLEU scores on newstest2016.

3.4 English→German

For the English→German phrase-based system, we exploited several translation factors in addition to word surface forms, in particular: Och clusters (with 50 classes) and part-of-speech tags (Ratnaparkhi, 1996) on the English side, as well as Och clusters (50 classes), morphological tags, and part-of-speech tags on the German side (Schmid, 2000). Recent experiments for our IWSLT 2015 phrase-based system have confirmed that English→German translation quality can benefit from these factors when supplementary models over factored representations are used (Huck and Birch, 2015). For WMT16, we utilized the factors in the translation model, in operation sequence models, and in language models (for linearly interpolated 7-gram LMs over Och clusters and morphological tags).

Sparse source word deletion, target word insertion, and word translation features were integrated over the top 200 word surface forms and over selected factors (source and target Och clusters, source part-of-speech tags and target morphological tags). An unpruned 5-gram LM over words that was trained on all German data except the CommonCrawl monolingual corpus was supplemented by a separate pruned LM trained on the CommonCrawl data that had been provided as permissible data for the “constrained” track. Rather than applying a simple linear distortion score, we opted for sparse distortion features as described by Green et al. (2010), which we reimplemented in Moses. We activated sparse distortion features with a feature template based on jump distance, source part-of-speech tags, and target morphological tags.

The feature weights for our final system were tuned with hypergraph MIRA (i.e. batch MIRA over lattices representing the decoding search space) on a concatenation of newssyscomb2009 and newstest2008–2012.

3.5 German→English

For phrase-based translation from German, we applied syntactic pre-reordering (Collins et al., 2005) and compound splitting (Koehn and Knight, 2003) in a preprocessing step on the source side. The operation sequence model for the German→English phrase-based system was unpruned. We integrated three language models: an unpruned LM over all English data except the CommonCrawl monolingual corpus; a pruned LM over CommonCrawl; and a pruned LM over the monolingual News Crawl 2015 corpus. In addition to lexical smoothing with the standard lexicon models, we utilized a source-to-target IBM Model 1 (Brown et al., 1993) for sentence-level lexical scoring in a similar manner as described by Huck et al. (2011) for hierarchical systems. We tuned on the concatenation of newssyscomb2009 and newstest2008–2012.

Unlike last year’s system (Haddow et al., 2015)—and different from the inverse translation direction (English→German)—we refrained from using any factors and instead set up a system that operates over surface form word representations only. In relation to last year’s system, we were able to maintain high translation quality as measured in BLEU despite the abandonment of factors. However, we suspect that human judgment scores may suffer a bit from the abandonment of a factored model. We decided to drop the factored representations in favour of gains in decoding efficiency. We furthermore did not employ any sparse features (sparse phrase length, source word deletion, target word insertion, or word translation features) in the German→English system since we did not observe any clear gains in preliminary experiments, and sparse features slow down tuning and decoding.

English→German and German→English translation results with our phrase-based systems are given in Table 3.

3.6 Spanish→English Biomedical

For our submission to the Spanish→English biomedical task, we created a parallel corpus using**
Table 3: Experimental results with phrase-based systems for German→English and English→German.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>last year’s phrase-based</td>
<td>27.2</td>
<td>28.8</td>
<td>29.3</td>
<td>33.8</td>
<td>20.8</td>
<td>21.1</td>
<td>22.8</td>
<td>28.3</td>
</tr>
<tr>
<td>this year’s phrase-based</td>
<td>27.8</td>
<td>30.0</td>
<td>29.9</td>
<td>35.1</td>
<td>21.5</td>
<td>21.9</td>
<td>23.7</td>
<td>28.4</td>
</tr>
</tbody>
</table>

4 Syntax-based System Overview

For all syntax-based systems, we used a string-to-tree model based on a synchronous context-free grammar (SCFG) with linguistically-motivated labels on the target side.

4.1 Preprocessing

Except for English-Czech, which we describe separately in Section 5.1, preprocessing was similar to the phrase-based systems (Section 2.3). To parse the target-side of the training data, we used the Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007) for English, and the ParZu dependency parser (Sennrich et al., 2013) for German. Except where stated otherwise, we right-binarized the trees after parsing to increase rule coverage.

4.2 Word Alignment

As in the phrase-based models, we used fast_align for word alignment and the grow-diag-final-and heuristic for symmetrization.

4.3 Language Models

As in the phrase-based systems (Section 2.3), we used linearly-interpolated language models as standard, with some systems adding Common-Crawl and neural LMs. We detail the system-specific combinations in Section 5.

4.4 Rule Extraction

SCFG rules were extracted from the word-aligned parallel data using the Moses implementation (Williams and Koehn, 2012) of the GHKM algorithm (Galley et al., 2004, 2006).

Minimal GHKM rules were composed into larger rules subject to restrictions on the size of the resulting tree fragment. We used the settings shown in Table 4, which were chosen empirically during the development of 2013’s systems (Nadejde et al., 2013).

<table>
<thead>
<tr>
<th>parameter</th>
<th>unbinarized</th>
<th>binarized</th>
</tr>
</thead>
<tbody>
<tr>
<td>rule depth</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>node count</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>rule size</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4: Parameter settings for rule composition. The parameters were relaxed for systems that used binarization to allow for the increase in tree node density.

Further to the restrictions on rule composition, fully non-lexical unary rules were eliminated using the method described in Chung et al. (2011) and rules with scope greater than 3 (Hopkins and Langmead, 2010) were pruned from the translation grammar. Scope pruning makes parsing tractable without the need for grammar binarization.

4.5 Baseline Features

Our core set of string-to-tree feature functions is unchanged from previous years. It includes the n-gram language model’s log probability for the target string, the target word count, the rule count, and several pre-computed rule-specific scores. The rule-specific scores were: the direct and indirect translation probabilities; the direct and indirect lexical weights (Koehn et al., 2003); the monolingual PCFG probability of the tree fragment from which the rule was extracted; and a rule
rareness penalty.

4.6 Decoding
Decoding for the string-to-tree models is based on Sennrich's (2014) recursive variant of the CYK+ parsing algorithm combined with LM integration via cube pruning (Chiang, 2007).

4.7 Tuning
The feature weights for the English→Czech and Finnish→English systems were tuned using the Moses implementation of MERT (Och, 2003). For the remaining systems we used k-best MIRA (Cherry and Foster, 2012) due to the use of sparse features.

We used randomly-chosen subsets of the previous years’ test data to speed up decoding.

5 Syntax-based Experiments
5.1 English→Czech
For English→Czech, we used Treex to preprocess and parse the Czech-side of the training data. Treex uses the MST parser (McDonald et al., 2005), which produces dependency graphs with non-projective arcs. In order to extract SCFG rules, we first applied the following conversion process: i) the dependency graphs were projectivized using the Malt Parser, which implements the method described in Nivre and Nilsson (2005) (we used the ‘Head’ encoding scheme); ii) the projective dependency graphs were converted to CFG trees. In addition, we reduced the complex positional tags to simple POS tags by discarding the morphological attributes. The CFG trees were not binarized.

We also experimented with unification-based agreement and case government constraints (Williams and Koehn, 2011; Williams, 2014). Specifically, our constraints were designed to enforce: i) case, gender, and number agreement between nouns and pre-nominal adjectival modifiers; ii) number and person agreement between subjects and verbs; iii) case agreement between prepositions and nouns; iv) use of nominative case for subject nouns. For every Czech word in the training data, we obtained a set of morphological analyses using MorphoDiTa (Straková et al., 2014). From these analyses, we constructed a lexicon of feature structures. For constraint extraction, we used handwritten rules along the lines of those described in Williams (2014).

In preliminary experiments we used a smaller training set, comprising 2 million sentence pairs sampled from OPUS and monolingual data from last year’s WMT translation task. We used two test sets from the HimL project and the Khresmoi test set. Results with and without constraints are shown in Table 5. We used hard constraints and reused the baseline weights (re-tuning did not appear to give additional gains).

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU HimL1</th>
<th>BLEU HimL2</th>
<th>BLEU Khresmoi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>23.3</td>
<td>18.6</td>
<td>20.4</td>
</tr>
<tr>
<td>+ constraints</td>
<td>23.6</td>
<td>18.8</td>
<td>20.7</td>
</tr>
</tbody>
</table>

Table 5: Translation results on the development system for English→Czech with unification-based constraints. Cased BLEU scores are shown. They are averaged over three tuning runs (note that baseline weights are reused in the experiments with constraints).

Although the gains in BLEU were small, previous analysis for German showed that BLEU lacks sensitivity to grammatical improvements when compared to human evaluators (Williams, 2014).

We trained the final system on all of the provided training and monolingual data. In addition to the interpolated LM, we used a model trained on the CommonCrawl data. Results are shown in Table 6.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU 2015</th>
<th>BLEU 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>17.3</td>
<td>20.1</td>
</tr>
<tr>
<td>+ constraints</td>
<td>17.5</td>
<td>20.2</td>
</tr>
<tr>
<td>+ CC LM</td>
<td>17.9</td>
<td>20.9</td>
</tr>
</tbody>
</table>

Table 6: Translation results on the final system for English→Czech with unification-based constraints. Cased BLEU scores are shown. Note that baseline weights are reused in the experiments with constraints.

5.1.1 Manual Analysis
We carried out a small manual analysis of the submitted system with and without unification-based constraints (the CC LM was used in both cases). In order to remove the effect of tuning variance, we used the same model weights in both cases (the weights were learned on the version without...
The BLEU scores of the two systems were 20.9 (with constraints) and 20.7 (without constraints). A large majority of the outputs (81% of the 2999 sentences in the news Testament2016) are identical.

Looking at a sample of 100 sentences with some differences, we classified differing areas to see in what aspects the outputs of the two systems differ. In total, there were 104 such areas (some sentences had more than one area of interest).

Table 7 summarizes the overall evaluation of these areas (the annotation was not blind, we knew which system was which). The majority of the areas were of an equal quality, in fact equally bad overall, so neither of the compared systems delivered an acceptable translation.

<table>
<thead>
<tr>
<th>Much</th>
<th>Better</th>
<th>Better</th>
<th>Equal</th>
<th>Worse</th>
<th>Crazy Reordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>41</td>
<td>44</td>
<td>12</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Manual evaluation of translations as proposed by the English → Czech system with unification constraints vs. the same system without constraints.

In 4 cases, the system with constraints delivered much better translation, and three of those were overall improvement of the sentence structure.

In 41 cases, the area was better for various reasons. Most frequently (16 cases), this was indeed the agreement within noun and prepositional phrases (adjective matching in case the preposition etc.). In 9 additional cases, the NP or PP was better translated but in other aspects than morphological case, number of gender. For instance the baseline system translated the phrase “between the departments of individual hospitals” as “between the departments of the hospitals” (in morphologically well-formed Czech). Beyond better NPs and PPs, the constraints have also helped overall sentence or clause structure (5 cases), lexical choice (4 cases) and verbs and their belongings (2 cases).

In 15 cases, the constraints forced the system to select a worse translation, damaging sentence structure, lexical choice, spuriously introducing negation etc. We highlight 3 of these cases, where the system with constraints accidentally moved words far away from their correct location (“Crazy Reordering” in Table 7). This suggests that due to sparse data, the application of constraints should be better balanced with respect to other parts of the model. In contrast to German, targeting Czech usually does not need long-distance reordering and doing it risks more serious translation errors than sticking to the English word order.

Since the hard unification constraints effectively only avoid some of the possible translations (i.e. reduce the search space), we conclude that having to obey mere agreement constraints helps to select a hypothesis better in a surprisingly larger span of words, improving overall sentence structure on average.

5.2 English → German

This year’s string-to-tree submission for English → German is similar to last year’s system (Williams et al., 2015). In addition to the baseline feature functions, it contains count-based 5-gram Neural Network language model (NPLM) (Vaswani et al., 2013), a relational dependency language model (RDLM) (Sennrich, 2015), and soft source-syntactic constraints (Huck et al., 2014). The parameters of the model are tuned towards the linear interpolation of BLEU and the syntactic metric HWCM (Liu and Gildea, 2005; Sennrich, 2015). Trees are transformed through binarization and a hierarchical representation of morphologically complex words (Sennrich and Haddow, 2015).

For the soft source-syntactic constraints, we annotate the source text with the Stanford Neural Network dependency parser (Chen and Manning, 2014), along with heuristic projectivization (Nivre and Nilsson, 2005).

Results are shown in Table 8. We report results of last year’s system (Williams et al., 2015), which was ranked (joint) first at WMT 15. Our improvements this year stem from particle verb restructuring (Sennrich and Haddow, 2015), and the use of the new monolingual News Crawl 2015 corpus for...
the Kneser-Ney language model.\footnote{The neural language models were trained on last year’s training data.}

## 5.3 Finnish → English

Our Finnish → English syntax-based system was similar to last year’s (Williams et al., 2015). The main difference from the basic setup of Section 4 is that we preprocessed the Finnish data to segment words into morphemes. We also added a CommonCrawl language model in addition to the interpolated LM.

For segmentation, we used Morfessor 2.0 with default settings, first training a segmentation model, then using it to segment all words in the source-side training and test data. Morfessor takes a set of word types as input and we found that it was important for translation quality to use a large training vocabulary. Table 9 gives mean BLEU scores for this setup, averaged over three MERT runs. Our baseline is the standard string-to-tree setup (i.e. without segmentation and without the CommonCrawl LM). For segmentation, we experimented with varying amounts of training data, initially using the Finnish side of the provided parallel corpora, then adding the monolingual Finnish data (apart from CommonCrawl), and finally adding 10% of the CommonCrawl vocabulary (we extracted the full vocabulary from CommonCrawl and then randomly sampled 10%).

We found that using larger amounts of training data was prohibitively slow.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>16.0</td>
<td>18.2</td>
</tr>
<tr>
<td>+ Morfessor (all parallel)</td>
<td>16.8</td>
<td>19.1</td>
</tr>
<tr>
<td>+ Morfessor (non-CC mono)</td>
<td>17.6</td>
<td>20.1</td>
</tr>
<tr>
<td>+ Morfessor (10% CC)</td>
<td>17.9</td>
<td>20.1</td>
</tr>
<tr>
<td>+ CC LM</td>
<td>18.0</td>
<td>20.3</td>
</tr>
</tbody>
</table>

Table 9: Comparison of different preprocessing and language model regimes for Finnish → English (syntax-based). Cased BLEU scores are given for the newstest2015 and newstest2016 test sets, averaged over three tuning runs.

## 5.4 German → English

For German → English we built a string-to-tree system with a similar setup to the German → English system. However we did not use compound splitting and we allowed glue rules. Similar to the phrase-based setup we used half of the newdev2016 for tuning and the other half as development set. We normalized the corpora by removing all diacritics from the Romanian side. We report the cased BLEU scores for different setups of our system in Table 10.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>28.6</td>
<td>33.5</td>
</tr>
<tr>
<td>+ NER before split</td>
<td>28.8</td>
<td>33.8</td>
</tr>
<tr>
<td>+ CommonCrawl LM*</td>
<td>29.4</td>
<td>34.4</td>
</tr>
<tr>
<td>+ NER before split</td>
<td>28.1</td>
<td>33.0</td>
</tr>
</tbody>
</table>

Table 10: Translation results of German → English string-to-tree translation system on dev (newstest2015) and test (newstest2016). *submitted system.

known words similar to the English → German systems described by Williams et al. (2014) and Sennrich et al. (2015). We also tagged named entities to avoid over-splitting of compounds. For example the script provided with Moses for compound splitting will split Florstadt nach Bad Salzhausen into flor Stadt nach Bad Salt; hausen. This is then wrongly translated by the baseline system as Flor after bath salt station. We applied a 3–class named entity tagger (Finkel et al., 2005; Faruqui and Padó, 2010) on the German side of the corpus prior to splitting and removed the annotations afterwards. We also trained a contrastive system with target–side dependency relations instead of PTB–style phrase-structures. The English side of the parallel corpora was annotated with the Stanford Neural Network dependency parser (Chen and Manning, 2014), along with heuristic projectivization (Nivre and Nilsson, 2005) and head-binaryization (Sennrich and Haddow, 2015). We report the cased BLEU scores for different setups of our system in Table 11.

## 5.5 Romanian → English

For Romanian → English we built a string-to-tree system similar to the German → English system. However we did not use compound splitting and we allowed glue rules. Similar to the phrase-based setup we used half of the newsdev2016 for tuning and the other half as development set. We normalized the corpora by removing all diacritics from the Romanian side. We report the cased BLEU scores for different setups of our system in Table 11.

## 6 Conclusion

The Edinburgh team built a total of 11 phrase-based and syntax-based translation systems us-
Table 11: Translation results of Romanian→English string-to-tree translation system on dev (half of newsdev2016) and test (newstest2016). *submitted system.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (phrase-structure)</td>
<td>33.9</td>
</tr>
<tr>
<td>+ UNK NT labels</td>
<td>34.2</td>
</tr>
<tr>
<td>+ CommonCrawl LM*</td>
<td>35.2</td>
</tr>
<tr>
<td>contrastive (dependency)</td>
<td>33.7</td>
</tr>
<tr>
<td>+ UNK NT labels</td>
<td>32.9</td>
</tr>
</tbody>
</table>

Acknowledgments

We are grateful to Martin Popel for assistance with Treex. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements 645452 (QT21), 644333 (TraMOOC) and 644402 (HimL).

References


Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd
Annual Meeting on Association for Computational Linguistics. pages 363–370.


the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, pages 48–54.


A Joint Dependency Model of Morphological and Syntactic Structure for Statistical Machine Translation

Rico Sennrich and Barry Haddow
School of Informatics, University of Edinburgh
rico.sennrich@ed.ac.uk, bhaddow@inf.ed.ac.uk

Abstract

When translating between two languages that differ in their degree of morphological synthesis, syntactic structures in one language may be realized as morphological structures in the other, and SMT models need a mechanism to learn such translations. Prior work has used morpheme splitting with flat representations that do not encode the hierarchical structure between morphemes, but this structure is relevant for learning morphosyntactic constraints and selectional preferences. We propose to model syntactic and morphological structure jointly in a dependency translation model, allowing the system to generalize to the level of morphemes. We present a dependency representation of German compounds and particle verbs that results in improvements in translation quality of 1.4–1.8 BLEU in the WMT English–German translation task.

1 Introduction

When translating between two languages that differ in their degree of morphological synthesis, syntactic structures in one language may be realized as morphological structures in the other. Machine Translation models that treat words as atomic units have poor learning capabilities for such translation units, and morphological segmentations are commonly used (Koehn and Knight, 2003). Like words in a sentence, the morphemes of a word have a hierarchical structure that is relevant in translation. For instance, compounds in Germanic languages are head-final, and the head is the segment that determines agreement within the noun phrase, and is relevant for selectional preferences of verbs.

1. sie erheben eine Handgepäckgebühr.

Table 1: Surface realizations of particle verb weggehen 'walk away'.

<table>
<thead>
<tr>
<th>function/position</th>
<th>English/German example</th>
</tr>
</thead>
<tbody>
<tr>
<td>finite (main)</td>
<td>he walks away quickly</td>
</tr>
<tr>
<td></td>
<td>or geht schnell weg</td>
</tr>
<tr>
<td>finite (sub.)</td>
<td>[...] because he walks away quickly</td>
</tr>
<tr>
<td></td>
<td>[...] weil er schnell weggeht</td>
</tr>
<tr>
<td>bare infinitive</td>
<td>he can walk away quickly</td>
</tr>
<tr>
<td></td>
<td>or kann schnell weggehen</td>
</tr>
<tr>
<td>to/zu-infinitive</td>
<td>he promises to walk away quickly</td>
</tr>
<tr>
<td></td>
<td>or verspricht, schnell weggehen</td>
</tr>
</tbody>
</table>

In example 1, agreement in case, number and gender is enforced between eine 'a' and Gebühr 'fee', and selectional preference between erheben 'charge' and Gebühr 'fee'. A flat representation, as is common in phrase-based SMT, does not encode these relationships, but a dependency representation does so through dependency links.

In this paper, we investigate a dependency representation of morphologically segmented words for SMT. Our representation encodes syntactic and morphological structure jointly, allowing a single model to learn the translation of both. Specifically, we work with a string-to-tree model with GHKM-style rules (Galley et al., 2006), and a relational dependency language model (Sennrich, 2015). We focus on the representation of German syntax and morphology in an English-German system, and two morphologically complex word classes in German that are challenging for translation, compounds and particle verbs.

German makes heavy use of compounding, and compounds such as Abwasserbehandlungsanlage 'waste water treatment plant' are translated into complex noun phrases in other languages, such as French station d’épuration des eaux résiduaires.

German particle verbs are difficult to model because their surface realization differs depending on the finiteness of the verb and the type of clause. Verb particles are separated from the finite verb in
main clauses, but prefixed to the verb in subordinated clauses, or when the verb is non-finite. The infinitive marker zu ‘to’, which is normally a premodifying particle, appears as an infix in particle verbs. Table 1 shows an illustrating example.

2 A Dependency Representation of Compounds and Particle Verbs

The main focus of research on compound splitting has been on the splitting algorithm (Popovic et al., 2006; Nielen and Ney, 2000; Weller et al., 2014; Macherey et al., 2011). Our focus is not the splitting algorithm, but the representation of compounds. For splitting, we use an approach similar to (Fritzinger and Fraser, 2010), with segmentation candidates identified by a finite-state morphology (Schmid et al., 2004; Sennrich and Kunz, 2014), and statistical evidence from the training corpus to select a split (Koehn and Knight, 2003).

German compounds are head-final, and premodifiers can be added recursively. Compounds are structurally ambiguous if there is more than one modifier. Consider the distinction between (Stadtteil)projekt (literally: ‘city sub-project’) and Stadt (teil)projekt ‘city sub-project’. We opt for a left-branching representation by default.1 We also split linking elements, and represent them as a postmodifier of each non-final segment, including the empty string (‘ε’). We use the same representation for noun compounds and adjective compounds.

An example of the original2 and the proposed compound representation is shown in Figure 1. Importantly, the head of the compound is also the parent of the determiners and attributes in the noun phrase, which makes a bigram dependency language model sufficient to enforce agreement. Since we model morphosyntactic agreement within the main translation step, and not in a separate step as in (Fraser et al., 2012), we deem it useful that inflection is marked at the head of the compound. Consequently, we do not split off inflectional or derivational morphemes.

For German particle verbs, we define a common representation that abstracts away from the various surface realizations (see Table 1). Separated

---

1. We follow prior work in leaving frequent words or subwords unsplit, which has a disambiguating effect. With more aggressive splitting, frequency information could be used for the structural disambiguation of internal structure.
2. The original dependency trees follow the annotation guidelines by Foth (2005).
pendency language model (RDLM). Figure 3 (a) shows the constituency representation of the example in Figure 1.

Our model should not only be able to produce new words productively, but also to memorize words it has observed during training. Looking at the compound *Handgepäckgebühr* in Figure 3 (a), we can see that it does not form a constituent, and cannot be extracted with GHKM extraction heuristics. To address this, we binarize the trees in our training data (Wang et al., 2007).

A complicating factor is that the binarization should not impair the RDLM. During decoding, we map the internal tree structure of each hypothesis back to the unbinarized form, which is then scored by the RDLM. Virtual nodes introduced by the binarization must also be scorable by RDLM if they form the root of a translation hypothesis. A simple right or left binarization would produce virtual nodes without head and without meaningful dependency representation. We ensure that each virtual node dominates the head of the full constituent through a mixed binarization.\(^3\) Specifically, we perform right binarization of the head and all pre-modifiers, then left binarization of all post-modifiers. This head-binarized representation is illustrated in Figure 3 (b).\(^4\)

Head binarization ensures that even hypotheses whose root is a virtual node can be scored by the RDLM. This score is only relevant for pruning, and discarded when the full constituent is scored. Still, these hypotheses require special treatment in the RDLM to mitigate search errors. The virtual node labels (such as OBJA) are unknown symbols to the RDLM, and we simply replace them with the original label (OBJA). The RDLM uses sibling context, and this is normally padded with special start and stop symbols, analogous to BOS/EOS symbols in n-gram models. These start and stop symbols let the RDLM compute the probability that a node is the first or last child of its ancestor node. However, computing these probabilities for virtual nodes would unfairly bias the search, since the first/last child of a virtual node is not necessarily the first/last child of the full constituent. We adapt the representation of virtual nodes in RDLM to take this into account. We distinguish between virtual nodes based on whether their span is a string prefix, suffix, or infix of the full constituent. For prefixes and infixes, we do not add a stop symbol at the end, and use null symbols, which denote unavailable context, for padding to the right. For suffixes and infixes, we do the same at the start.

4 Post-Processing

For SMT, all German training and development data is converted into the representation described in sections 2–3. To restore the original representation, we start from the tree output of the string-to-tree decoder. Merging compounds is trivial: all segments and linking elements can be identified by the tree structure, and are concatenated.

For verbs that dominate a verb particle, the original order is restored through three rules:

1. non-finite verbs are concatenated with the particle, and zu-markers are infixed.
2. finite verbs that head a subordinated clause (identified by its dependency label) are concatenated with the particle.
3. finite verbs that head a main clause have the
particle moved to the right clause bracket.\footnote{We use the last position in the clause as default location, but put the particle before any subordinated and coordinated clauses, which occur in the Nachfeld (the ‘final field’ in topological field theory).}

Previous work on particle verb translation into German proposed to predict the position of particles with an n-gram language model (Nießen and Ney, 2001). Our rules have the advantage that they are informed by the syntax of the sentence and consider the finiteness of the verb.

Our rules only produce projective trees. Verb particles may also appear in positions that violate projectivity, and we leave it to future research to determine if our limitation to projective trees affects translation quality, and how to produce non-projective trees.

5 SMT experiments

5.1 Data and Models

We train English–German string-to-tree SMT systems on the training data of the shared translation task of the Workshop on Statistical Machine Translation (WMT) 2015. The data set consists of 4.2 million sentence pairs of parallel data, and 160 million sentences of monolingual German data.

We base our systems on that of Williams et al. (2014). It is a string-to-tree GHKM translation system implemented in Moses (Koehn et al., 2007), and using the dependency annotation by ParZu (Sennrich et al., 2013). Additionally, our baseline system contains a dependency language model (RDLM) (Sennrich, 2015), trained on the target-side of the parallel training data.

We report case-sensitive BLEU scores on the newstest2014/5 test sets from WMT, averaged over 3 optimization runs of k-batch MIRA (Cherry and Foster, 2012) on a subset of newstest2008-12.

We split all particle verbs and hyphenated compounds, but other compounds are only split if they are rare (frequency in parallel text < 5).

For comparison with the state-of-the-art, we train a full system on our restructured representation, which incorporates all models and settings of our WMT 2015 submission system (Williams et al., 2015).\footnote{We use \texttt{mteval-v13a.pl} for comparability to official WMT results; all significance values reported are obtained with \texttt{MultiEval} (Clark et al., 2011).} Note that our WMT 2015 submission uses the dependency representation of compounds and tree binarization introduced in this paper; we achieve additional gains over the submission system through particle verb restructuring.

5.2 SMT Results

Table 2 shows translation quality (BLEU) with different representations of German compounds and particle verbs. Head binarization not only yields improvements over the baseline, but also allows for larger gains from morphological segmentation. We attribute this to the fact that full compounds, and prefixed particle verbs, are not always a constituent in the segmented representation, and that binarization compensates this theoretical drawback.

With head binarization, we find substantial improvements from compound splitting of 0.7–1.1 BLEU. On newstest2014, the improvement is almost twice of that reported in related work (Williams et al., 2014), which also uses a hierarchical representation of compounds, albeit one that does not allow for dependency modelling. Examples of correct, unseen compounds generated include \textit{Staubsaugertrobo\text{"{e}r}’ vacuum cleaner robot’, \text{Gravitationswellen}’ gravitational waves’, and \text{NPD-verbotsverfahren}’ NPD banning process’}.\footnote{\text{Vaswani et al., 2013}, and soft source-syntactic constraints (Huck et al., 2014).} 5

\footnote{Note that \textit{Staubsauger}, despite being a compound, is not considered a compound in this study.}

\begin{table}[h]
\centering
\begin{tabular}{lrr}
\hline
system & newstest2014 & newstest2015 \\
\hline
baseline & 20.7 & 22.0 \\
+split compounds & 21.3 & 22.4 \\
+particle verbs & 21.4 & 22.8 \\
head binarization & 20.9 & 22.7 \\
+split compounds & 22.0 & 23.4 \\
+particle verbs & 22.1 & 23.8 \\
full system & 22.6 & 24.4 \\
\hline
\end{tabular}
\caption{English–German translation results (BLEU). Average of three optimization runs.}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{llllll}
\hline
\text{compound sep pref zu-infix} & \text{reference} & 2841 & 553 & 1195 & 176 \\
\hline
baseline & 845 & 96 & 847 & 91 \\
+head binarization & 796 & 157 & 858 & 106 \\
+split compounds & 1850 & 160 & 877 & 94 \\
+particle verbs & 1992 & 333 & 953 & 169 \\
\hline
\end{tabular}
\caption{Number of compounds [that would be split by compound splitter] and particle verbs (separated, prefixed and with \text{zu}-infix) in newstest2014/5. Average of three optimization runs.}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{llllll}
\hline
\text{compound sep pref zu-infix} & \text{reference} & 2841 & 553 & 1195 & 176 \\
\hline
baseline & 845 & 96 & 847 & 91 \\
+head binarization & 796 & 157 & 858 & 106 \\
+split compounds & 1850 & 160 & 877 & 94 \\
+particle verbs & 1992 & 333 & 953 & 169 \\
\hline
\end{tabular}
\caption{Number of compounds [that would be split by compound splitter] and particle verbs (separated, prefixed and with \text{zu}-infix) in newstest2014/5. Average of three optimization runs.}
\end{table}
Particle verb restructuring yields additional gains of 0.1–0.4 BLEU. One reason for the smaller effect of particle verb restructuring is that the difficult cases – separated particle verbs and those with infixation – are rarer than compounds, with 2841 rare compounds [that would be split by our compound splitter] in the reference texts, in contrast to 553 separated particle verbs, and 176 particle verbs with infixation, as Table 3 illustrates. If we only evaluate the sentences containing a particle verb with zw-infix in the reference, 165 in total for newstest2014/5, we observe an improvement of 0.8 BLEU on this subset (22.1—22.9), significant with $p < 0.05$.

The positive effect of restructuring is also apparent in frequency statistics. Table 3 shows that the baseline system severely undergenerates compounds and separated/infix particle verbs. Binarization, compound splitting, and particle verb restructuring all contribute to bringing the distribution of compounds and particle verbs closer to the reference.

In total, the restructured representation yields improvements of 1.4–1.8 BLEU over our baseline. The full system is competitive with official submissions to the WMT 2015 shared translation tasks. It outperforms our submission (Williams et al., 2015) by 0.4 BLEU, and outperforms other phrase-based and syntax-based submissions by 0.8 BLEU or more. The best reported result according to BLEU is an ensemble of Neural MT systems (Jean et al., 2015), which achieves 24.9 BLEU. In the human evaluation, both our submission and the Neural MT system were ranked 1–2 (out of 16), with no significant difference between them.

### 5.3 Synthetic LM Experiment

We perform a synthetic experiment to test our claim that a dependency representation allows for the modelling of agreement between morphemes. For 200 rare compounds [that would be split by our compound splitter] in the newstest2014/5 references, we artificially introduce agreement errors by changing the gender of the determiner. For instance, *Handgepäck* is neuter, which could lead a morpheme-level n-gram model to prefer the determiner *ein*, but *Handgepäckgebühr* is feminine and requires *eine*.

---

**References**


---

9We released source code and configuration files at https://github.com/rsennrich/wmt2014-scripts.


How Grammatical is Character-level Neural Machine Translation?
Assessing MT Quality with Contrastive Translation Pairs

Rico Sennrich
School of Informatics, University of Edinburgh
{rico.sennrich}@ed.ac.uk

Abstract

Analysing translation quality in regards to specific linguistic phenomena has historically been difficult and time-consuming. Neural machine translation has the attractive property that it can produce scores for arbitrary translations, and we propose a novel method to assess how well NMT systems model specific linguistic phenomena such as agreement over long distances, the production of novel words, and the faithful translation of polarity. The core idea is that we measure whether a reference translation is more probable under a NMT model than a contrastive translation which introduces a specific type of error. We present LingEval971, a large-scale data set of 97,000 contrastive translation pairs based on the WMT English→German translation task, with errors automatically created with simple rules. We report results for a number of systems, and find that recently introduced character-level decoders for neural machine translation (Chung et al., 2016; Lee et al., 2016) can generate coherent and grammatical sentences. Chung et al. (2016) argue that the answer is yes, because BLEU on long sentences is similar to a baseline with longer subword units created via byte-pair encoding (BPE) (Sennrich et al., 2016a), but BLEU, being based on precision of short n-grams, is an unsuitable metric to measure the global coherence or grammaticality of a sentence. To allow for a more nuanced analysis of different machine translation systems, we introduce a novel method to assess neural machine translation that can capture specific error categories in an automatic, reproducible fashion.

Neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015) opens up new opportunities for automatic analysis because it can assign scores to arbitrary sentence pairs, in contrast to phrase-based systems, which are often unable to reach the reference translation. We exploit this property for the automatic evaluation of specific aspects of translation by pairing a human reference translation with a contrastive example that is identical except for a specific error. Models are tested as to whether they assign a higher probability to the reference translation than to the contrastive example.

A similar method of assessment has previously been used for monolingual language models (Sennrich and Haddow, 2015; Linzen et al., 2016), and we apply it to the task of machine translation. We present a large-scale test set of English—German contrastive translation pairs that allows for the automatic, quantitative analysis of a number of linguistically interesting phenomena that have previously been found to be challenging for machine translation.

1 Introduction

It has historically been difficult to analyse how well a machine translation system can learn specific linguistic phenomena. Automatic metrics such as BLEU (Papineni et al., 2002) provide no linguistic insight, and automatic error analysis (Zeman et al., 2011; Popovic, 2011) is also relatively coarse-grained. A concrete research question that has been unanswered so far is whether character-level decoders for neural machine translation (Chung et al., 2016; Lee et al., 2016) can generate coherent and grammatical sentences. Chung et al. (2016) argue that the answer is yes, because BLEU on long sentences is similar to a baseline with longer subword units created via byte-pair encoding (BPE) (Sennrich et al., 2016a), but BLEU, being based on precision of short n-grams, is an unsuitable metric to measure the global coherence or grammaticality of a sentence. To allow for a more nuanced analysis of different machine translation systems, we introduce a novel method to assess neural machine translation that can capture specific error categories in an automatic, reproducible fashion.

Neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015) opens up new opportunities for automatic analysis because it can assign scores to arbitrary sentence pairs, in contrast to phrase-based systems, which are often unable to reach the reference translation. We exploit this property for the automatic evaluation of specific aspects of translation by pairing a human reference translation with a contrastive example that is identical except for a specific error. Models are tested as to whether they assign a higher probability to the reference translation than to the contrastive example.

A similar method of assessment has previously been used for monolingual language models (Sennrich and Haddow, 2015; Linzen et al., 2016), and we apply it to the task of machine translation. We present a large-scale test set of English—German contrastive translation pairs that allows for the automatic, quantitative analysis of a number of linguistically interesting phenomena that have previously been found to be challenging for machine translation.
translation, including agreement over long distances (Koehn and Hoang, 2007; Williams and Koehn, 2011), discontiguous verb-particle constructions (Nießen and Ney, 2000; Loićica and Gulordava, 2016), generalization to unseen words (specifically, transliteration of names (Durrani et al., 2014)), and ensuring that polarity is maintained (Wetzel and Bond, 2012; Chen and Zhu, 2014; Fancellu and Webber, 2015).

We report results for neural machine translation systems with different choice of subword unit, identifying strengths and weaknesses of recently-proposed models.

2 Contrastive Translation Pairs

We create a test set of contrastive translation pairs from the EN→DE test sets from the WMT shared translation task. Each contrastive translation pair consists of a correct reference translation, and a contrastive example that has been minimally modified to introduce one translation error. We define the accuracy of a model as the number of times it assigns a higher score to the reference translation than to the contrastive one, relative to the total number of predictions. We have chosen a number of phenomena that are known to be challenging for the automatic translation from English to German.

1. noun phrase agreement: German determiners must agree with their head noun in case, number, and gender. We randomly change the gender of a singular definite determiner to introduce an agreement error.

2. subject-verb agreement: subjects and verbs must agree with one another in grammatical number and person. We swap the grammatical number of a verb to introduce an agreement error.

3. separable verb particle: verbs and their separable prefix often form a discontiguous semantic unit. We replace a separable verb particle with one that has never been observed with the verb in the training data.

4. polarity: arguably, polarity errors are under-measured the most by string-based MT metrics, since a single word/morpheme can reverse the meaning of a translation. We reverse polarity by deleting/inserting the negation particle nicht (‘not’), swapping the determiner ein (‘a’) and its negative counterpart kein (‘no’), or deleting/inserting the negation prefix un-.

5. transliteration: subword-level models should be able to copy or transliterate names, even unseen ones. For names that were unseen in the training data, we swap two adjacent characters.

Table 1 shows examples for each error type. Most are motivated by frequent translation errors; for EN→DE, source and target script are the same, so technically, we do not perform transliteration. Since transliteration of names and copying them is handled the same way by the encoder-decoder networks that we tested, we consider this error type a useful proxy to test the models’ transliteration capability.

All errors are introduced automatically, relying on statistics from the training corpus, a syntactic analysis with ParZu (Sennrich et al., 2013), and a finite-state morphology (Schmid et al., 2004; Sennrich and Kunz, 2014) to identify the relevant constructions and introduce errors. For contrastive pairs with agreement errors, we also annotate the distance between the words. For translation errors where we want to assess generalization to rare words (all except negation particles), we also provide the training set frequency of the word involved in the error (in case of multiple words, we report the lower frequency).

The automatic processing has limitations, and we opt for a high-precision approach – for instance, we only change the gender of determiners where case and number are unambiguous, so that we can produce maximally difficult errors.3

Table 1: Example contrastive translations pair for each error category.

<table>
<thead>
<tr>
<th>category</th>
<th>English</th>
<th>German (correct)</th>
<th>German (contrastive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP agreement</td>
<td>of the American Congress</td>
<td>der amerikanischen Kongress</td>
<td>der amerikanischen Kongress</td>
</tr>
<tr>
<td>separable verb particle</td>
<td>by resting</td>
<td>durch das Timing</td>
<td>durch das Timing</td>
</tr>
<tr>
<td>transliteration</td>
<td>Mr. Ensign’s office</td>
<td>Senator Ensignen Büro</td>
<td>Senator Ensignen Büro</td>
</tr>
</tbody>
</table>

3If we mistakenly introduce a case error, this makes it easier to spot from local context.
We expect that parsing errors will not invalidate the contrastive examples — correctly identifying the subject will affect the distance annotation, but changing the number of the verb should always introduce an error. Still, we report ceiling scores achievable by humans to account for the possibility that a generated error is not actually an error. We estimate the human ceiling by trying to select the correct variant for 20 contrastive translation pairs per category where our best system fails. The ceiling is below 100% because of errors in the reference translation, and cases that were undecidable by a human annotator (such as the gender of the 20-year-old).

From the 22 191 sentences in the original newsst20** sets, we create approximately 97 000 contrastive translation pairs.

3 Evaluation
In the evaluation section, our focus is on establishing baselines on the test set, and investigating the following research questions:

- how well do different subword-level models process unseen words, specifically names?
- sequence-length is increased in character-level models, compared to word-level or BPE-level models. Does this have a negative effect on grammaticality?

We mark all undecidable cases as wrong, and could perform better with random guessing.

<table>
<thead>
<tr>
<th>system (test set and size)</th>
<th>2014</th>
<th>2015</th>
<th>2016</th>
<th>2019</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPE-to-BPE</td>
<td>20.1 (21.0)</td>
<td>23.2 (21.0)</td>
<td>26.7 (26.9)</td>
<td>27.5 (27.9)</td>
</tr>
<tr>
<td>BPE-to-char</td>
<td>19.4 (20.6)</td>
<td>22.7 (22.6)</td>
<td>26.0 (25.9)</td>
<td>26.9 (26.9)</td>
</tr>
<tr>
<td>char-to-char</td>
<td>19.7 (20.7)</td>
<td>22.9 (22.7)</td>
<td>26.2 (26.1)</td>
<td>27.6 (27.5)</td>
</tr>
<tr>
<td>BPE-to-BPE (Sennrich et al., 2016a)</td>
<td>25.7 (26.9)</td>
<td>28.3 (28.9)</td>
<td>34.2 (34.2)</td>
<td>35.7 (35.7)</td>
</tr>
</tbody>
</table>

Table 3: Case-sensitive BLEU scores (EN-DE) on WMT newstest. We report scores with detokenized NIST BLEU (mteval-v13a.pl), and in brackets, tokenized BLEU with multi-bleu.perl.

3.1 Data and Methods
We train NMT systems with training data from the WMT 15 shared translation task EN—DE. We train three systems with different text representations on the parallel part of the training set:

- BPE-to-BPE (Sennrich et al., 2016a)
- BPE-to-char (Chung et al., 2016)
- char-to-char (Lee et al., 2016)

We use the implementations released by the respective authors, Nematus for BPE-to-BPE, and dl4mt-c2c for BPE-to-char and char-to-char. dl4mt-c2c also provides preprocessed training data, which we use for comparability.

Both tools are forks of the dl4mt tutorial, so the implementation differences are minimal except for those pertaining to the text representation. We report hyperparameters in Table 2. They correspond to those used by Lee et al. (2016) for BPE-to-char and char-to-char, for BPE-to-BPE, we also adopt some hyperparameters from Sennrich et al. (2016b), most importantly, we extract a joint BPE vocabulary of size 89 500 from the parallel corpus. We trained the BPE-to-BPE system for one week, following Sennrich et al. (2016a), and the *-to-char systems for two weeks, following Lee et al. (2016), on a single Titan X GPU. For both translating and scoring, we normalize probabilities by length (the number of symbols on the target side).

We also report results with the top-ranked system at WMT16 (Sennrich et al., 2016a), which is available online. It is also a BPE-to-BPE system, but in contrast to the previous systems, it includes different preprocessing (including truecasings), other hyperparameters, additional monolin-
Our main result is the assessment via contrastive translation pairs, shown in Table 4. We find that despite obtaining similar BLEU scores, the models have learned different structures to a different degree. The models with character decoder make fewer transliteration errors than the BPE-to-BPE model. However, they perform more poorly on separable verb particles and agreement, especially as distance increases, as seen in Figure 1. While accuracy for subject-verb agreement of adjacent words is similar across systems (95.2%, 94.0%, and 94.5% for BPE-to-BPE, BPE-to-char, and char-to-char, respectively), the gap widens for agreement between distant words – for a distance of over 15 words, the accuracy is 90.7%, 85.2%, and 82.3%, respectively.

Polarity shifts between the source and target text are a well-known translation problem, and our analysis shows that the main type of error is the deletion of negation markers, in line with findings of previous studies (Fancellu and Webber, 2015). We consider the relatively high number of errors related to polarity an important problem in machine translation, and hope that future work will try to improve upon our results, shown in more detail in Table 5.

We have commented that changing the grammatical number of the verb may change the meaning of the sentence instead of making it disjoint. A common example is the German pronoun sie, which is shared between the singular 'she', and the plural 'they'. We keep separate statistics for this type of error (n = 2520), and find that it is challenging for all models, with an accuracy of 87–87.2% for single models, and 90% by the WMT16 submission system.

We conclude from our results that there is currently a trade-off between generalization to unseen words, for which character-level decoders perform best, and sentence-level grammaticality, for which we observe better results with larger subword units of the BPE segmentation. We hope that our test set will help in developing and assessing architectures.

**Table 5: Accuracy (in percent) of models on different categories of contrastive errors related to polarity.** Best single model result in bold (multiple bold results indicate that difference to best system is not statistically significant).

<table>
<thead>
<tr>
<th>System</th>
<th>Insertion</th>
<th>Deletion</th>
<th>Transliteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>char-to-char</td>
<td>97.1</td>
<td>92.7</td>
<td>95.4</td>
</tr>
<tr>
<td>BPE-to-BPE</td>
<td>96.1</td>
<td>93.6</td>
<td>94.8</td>
</tr>
<tr>
<td>BPE-to-char</td>
<td>97.2</td>
<td>94.0</td>
<td>95.2</td>
</tr>
<tr>
<td>Human</td>
<td>97.9</td>
<td>95.4</td>
<td>96.4</td>
</tr>
</tbody>
</table>

Note: Two commonly used BLEU evaluation scripts, the NIST BLEU scorer `mteval-v13a.pl` on detokenized text, and `multi-blcu_parallel` on tokenized text, give different results due to tokenization differences. We here report both for comparison, but encourage the use of the NIST scorer, which is used by the WMT and IWSLT shared tasks, and allows for comparison of systems with different tokenizations.
D2.3: Final Report: Morphologically Rich Languages

<table>
<thead>
<tr>
<th>System</th>
<th>Sentence</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Since then we have only played in the Swedish league which is not the same level.</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td>Seitdem haben wir nur in der schwedischen Liga gespielt, die nicht das gleiche Niveau hat.</td>
<td>0.149</td>
</tr>
<tr>
<td>1-best</td>
<td>Seitdem haben wir nur in der schwedischen Liga gespielt, die nicht das gleiche Niveau haben.</td>
<td>0.137</td>
</tr>
<tr>
<td>Source</td>
<td>FriendsFest: the comedy show that taught us serious lessons about male friendship</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td>FriendsFest: die Comedy-Show, die uns ernsthafte Lektionen über Männerfreundschaften erteilt</td>
<td>0.276</td>
</tr>
<tr>
<td>Contrastive</td>
<td>FriendsFest: die Comedys-Show, die uns ernsthafte Lektionen über Männerfreundschaften erteilen</td>
<td>0.262</td>
</tr>
<tr>
<td>1-best</td>
<td>FriendsFest: die Komödie zeigte, dass uns ernsthafte Lehren aus männlichen Freundschaften</td>
<td>0.129</td>
</tr>
<tr>
<td>Source</td>
<td>Robert Lewandowski has had the best opportunities in the first half.</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td>Die besten Gelegenheiten in Hälfte eins hatte Robert Lewandowski.</td>
<td>0.551</td>
</tr>
<tr>
<td>Contrastive</td>
<td>Die besten Gelegenheiten in Hälfte eins hatten Robert Lewandowski.</td>
<td>0.507</td>
</tr>
<tr>
<td>1-best</td>
<td>Robert Lewandowski hatte in der ersten Hälfte die besten Möglichkeiten.</td>
<td>0.046</td>
</tr>
</tbody>
</table>

Table 6: Examples where char-to-char model prefers contrastive translation (subject-verb agreement errors). 1-best translation can make error of same type (example 1), different type (translation of taught is missing in example 2), or no error (example 3).

that aim to overcome this trade-off and perform best in respect to both morphology and syntax.

We encourage the use of contrastive translation pairs, and LingEval97, for future analysis, but here discuss some limitations. The first one is by design: being focused on specific translation errors, the evaluation is not suitable as a global quality metric. Also, the evaluation only compares the probability of two translations, a reference translation $T^*$ and a contrastive translation $T'$, and makes no statement about the most probable translation $T^*$. Even if a model correctly estimates that $p(T) > p(T')$, it is possible that $T'$ will contain an error of the same type as $T^*$. And even if a model incorrectly estimates that $p(T) < p(T')$, it may produce a correct translation $T^*$. Despite these limitations, we argue that contrastive translation pairs are useful because they can easily be created to analyse any type of error in a way that is model-agnostic, automatic and reproducible.

Table 6 shows different examples of the where the contrastive translation is scored higher than the reference by the char-to-char model, and the corresponding 1-best translation. In the first one, our method automatically recognizes an error that also appears in the 1-best translation. In the second example, the 1-best translation is missing the verb. Such cases could confound a human analysis of agreement errors, and we consider it an advantage of our method that it is not confounded by other errors in the 1-best translation. In the third example, our method identifies an error, but the 1-best translation is correct. We note that the German reference exhibits object fronting, but the 1-best output has the more common SVO word order. While one could consider this instance a false positive, it can be important for an NMT model to properly score translations other than the 1-best, for instance for applications such as prefix-constrained MT (Wuebker et al., 2016).

4 Conclusion

We present LingEval97, a test set of 97,000 contrastive translation pairs for the assessment of neural machine translation systems. By introducing specific translation errors to the contrastive translations, we gain valuable insight into the ability of state-of-the-art neural MT systems to handle several challenging linguistic phenomena. A core finding is that recently proposed character-level decoders for neural machine translation outperform subword models at processing unknown names, but perform worse at modelling morphosyntactic agreement, where information needs to be carried over long distances. We encourage the use of LingEval97 to assess alternative architectures, such as hybrid word-character models (Luong and Manning, 2016), or dilated convolutional networks (Kalchbrenner et al., 2016). For the tested systems, the most challenging error type is the deletion of negation markers, and we hope that our test set will facilitate development and evaluation of models that try to improve in that respect. Finally, the evaluation via contrastive translation pairs is a very flexible approach, and can be applied to new language pairs and error types.

Acknowledgments

This project received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements 645452 (QT21) and 688139 (SUMMA).
References


Morphological Segmentation for Neural Machine Translation

Abstract

The state of the art of handling rich morphology in neural machine translation (NMT) is to break word forms into subword units, so that the overall vocabulary size of these units fits the practical limits given by the NMT model and GPU memory capacity. In this paper, we compare two common but linguistically uninformed methods of subword construction (BPE and STE, the method implemented in Tensor2Tensor toolkit) and two linguistically-motivated methods: Morfessor and one novel method, based on a derivational dictionary. Our experiments with German-to-Czech translation, both morphologically rich, document that so far, the non-motivated methods perform better. Furthermore, we identify a critical difference between BPE and STE and show a simple pre-processing step for BPE that considerably increases translation quality as evaluated by automatic measures.

1 Introduction

One of the key steps that allowed to apply neural machine translation (NMT) in unrestricted setting was the move to subword units. While the natural (target) vocabulary size in a realistic parallel corpus exceeds the limits imposed by model size and GPU RAM, the vocabulary size of custom subwords can be kept small.

The current most common technique of subword construction is called byte-pair encoding (BPE, Sennrich et al., 2016)1 and its counterpart originating in the commercial field is wordpieces (Wu et al., 2016). Yet another variant of the technique is implemented in Google’s open-sourced toolkit Tensor2Tensor2, namely the SubwordTextEncoder class (abbreviated as STE below).

The common property of these approaches is that they are trained in an unsupervised fashion, relying on the distribution of character sequences, but regardless any morphological properties of the languages in question.

On the positive side, BPE and STE (when trained jointly for both the source and target languages) allow to identify and benefit from words that share the spelling in some of their part, e.g. the root of the English “legalization” and Czech “legalizace” (noun) or “legalizační” (adj).

On the downside, the root of different word forms of one lemma can be split in several different ways and the neural network cannot benefit from their relatedness.

In this paper, we compare the performance of several variations of BPE and STE in a setting involving two morphologically rich languages: German-to-Czech translation. We also experiment with two methods aimed at morphologically adequate splitting of words.

2 Data and Common Settings

Our training data consist of Europarl v7 (Koehn, 2005) and OpenSubtitles2016 (Tiedemann, 2009), after some further cleanup. Our final training corpus, processed with the Moses tokenizer (Koehn et al., 2007), is summarized in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>German</th>
<th>Czech</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel sentences</td>
<td>8.8M</td>
<td></td>
</tr>
<tr>
<td>Tokens</td>
<td>89M</td>
<td>78M</td>
</tr>
<tr>
<td>Vocabulary size</td>
<td>807k</td>
<td>953k</td>
</tr>
</tbody>
</table>

Table 1: Training data size.

We use WMT3 newstest2011 as the development set and newstest2013 as the test set, 3k sentence pairs each.

1 http://github.com/rsennrich/subword-nmt/
2 https://github.com/tensorflow/tensor2tensor
3 http://www.statmt.org/wmt13
All experiments were carried out in Tensor2Tensor (abbreviated as T2T), version 1.2.9,\footnote{We report cased variant computed by the bleu\_wrapper method in \url{https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/utils/bleu_hook.py}} using the model transformer\_big\_single\_gpu, batch size of 1500 and learning\_rate\_warmup\_steps set to 30k or 60k if the learning diverged.

Vocabulary size is set to 100k when shared for both source and target and to 50k each with separate vocabularies.

Since T2T SubwordTextEncoder constructs the subword model only from a sample of the training data, we had to manually set the file\_byte\_budget variable in the code to 100M, otherwise not enough distinct wordforms were observed to fill up the intended 100k vocabulary size.

For data preprocessed by BPE, we used T2T TokenTextEncoder which allows to use a user-supplied vocabulary.

Final scores (BLEU,\footnote{https://github.com/tensorflow/tensor2tensor} CharacTER, chrF3 and BEER) are measured after removing any subword splits and detokenizing with Moses detokenizer. Each of the metric implementation handles tokenization on its own.

Since we cannot afford multiple training runs, we report the average score of the test set as translated by several model checkpoints, amongst the same number of training steps where the BLEU score has already flattened. This happens to be approximately after 40 hours of training around 300k training steps. (We do not see any real convergence or overfitting on our dataset.)

### 3 Subword Boundary Mark in BPE

By default, BPE subword boundaries are indicated with "@@". To reconstruct original tokens, one simply removes "@@".

<table>
<thead>
<tr>
<th>Input</th>
<th>Nesnese se se seenu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>Nesnese se se se seenu</td>
</tr>
<tr>
<td>Zigzag</td>
<td>NeS Snes Sse se se Sstrou</td>
</tr>
<tr>
<td>Atat</td>
<td>Ne @@ Sne Sse se se se Sstrou</td>
</tr>
</tbody>
</table>

Figure 1: Different subword boundary marks in the Czech translation of the sentence “He can’t get along with his sister”, one of the well known sentences with four repetitions of the syllable “se”.

The disadvantage of this approach is that e.g. the token “se” as the last subword of the word “nesnese” (“he can’t get along” in Czech) coincides with the standalone word “se” (ambiguous, corresponding to the preposition “with” or the reflexive pronoun), needlessly increasing token ambiguity, see Figure 1.

On the other hand, the early parts of German compounds get vocabulary entries different from standalone words. For example the word “Hof” ("a court" in German) can be joined with “Burg” ("a castle") resulting in “Hofburg” ("court castle"). BPE can learn to split the compound into two subwords, “Hof@@” and “burg” but the standalone “Hof” stays “Hof”, giving us two distinct vocabulary entries for a unit with supposedly one meaning.

The boundary mark can thus affect the performance and we evaluate the following:

- default – non-final subwords of a word marked with "@@"
- zigzag – all inner boundaries marked with a special character "$",
- atat – a token "@@" between subwords.

In BPE, the desired vocabulary size is set by the user and it corresponds to the number of subword merges plus the size of the alphabet. We apply zigzag or atat after the default BPE, so the vocabulary size changes (many entries get collapsed or disambiguated). To arrive again at the 50k vocabulary for NMT training, we varied the number of desired merges and took the final one where the vocabulary size was close to 50k.

Table 2 indicates that the default marking performs best in BLEU, but the results vary a lot across the individual checkpoints and evaluation metrics.

<table>
<thead>
<tr>
<th>ST</th>
<th>BLEU</th>
<th>CharacTER</th>
<th>chrF3</th>
<th>BEER</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>13.70±0.66 -12.99±1.34 0.37±0.01 0.42±0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DZ</td>
<td>13.73±0.24 -12.20±0.83 0.37±0.01 0.43±0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td>13.45±0.40 -12.15±1.11 0.36±0.00 0.42±0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZZ</td>
<td>13.64±0.26 -12.47±0.75 0.36±0.02 0.42±0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZ</td>
<td>13.29±0.20 -13.52±1.26 0.36±0.00 0.42±0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td>13.23±0.23 -12.27±0.77 0.36±0.00 0.42±0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Effect of boundary marking for source (S) and target (T); default (D), atat (A), zigzag (Z). Reported scores are avg±stddev of T2T checkpoints between 275k and 325k training steps.
4 BPE vs SubwordTextEncoder

In our second experiment, we compare BPE with STE. As we can see in Figure 2, a distinct feature of STE seems to be an underscore as a zero suffix mark appended to every word before the subword splits are determined. More adequate units can be thus learned compared to BPE. For example, the Czech word form "tramvaj" ("a tram") can serve as a subword unit that, combined with zero suffix ("_") corresponds to the nominative case or, combined with the suffix "e" to the genitive case "tramvaje". In BPE, there can be either "tram-

vaj" as a standalone word or two subwords "tram-
vaj"@ and "e" (or possibly split further) with no vocabulary entry sharing possible.

Table 3 presents the results: STE performs almost 5% BLEU point better than the default BPE. The underscore feature allowing to model zero suffix almost closes the gap and shared vocabulary also helps a little.

As Figure 3 indicates, the difference in performance is not a straightforward consequence of the number of splits generated. There is basically no difference between BPE with and without underscore but shared vocabulary leads to a lower number of splits on the Czech target side. We can see that STE in both languages splits words to more parts than BPE but still performs better. We conclude that the STE splits allow to exploit morphological behaviour better.

5 Morphological Segmentation: DeriNet and Morfessor

The limitation of morphology-agnostic subword splitting is that one morpheme can be split in several ways in different word forms, so the neural network cannot connect them easily. Hack et al. (2017) show that linguistically aware separation of suffixes prior to BPE helps slightly more than cascade of suffix, prefix and compound splitting on the target side of English to German translation task. The overall improvement of baseline BPE was about 0.8 BLEU points on a medium size translation task. Pinnis et al. (2017) show similar improvements with analogical prefix and suffix splitting on English to Latvian task.

We analyzed the application of Morfessor 2.0 (Virpioja et al., 2013), a tool for unsupervised morpheme induction, and a novel method based on a derivational morphology, for morphological segmentation.
5.1 DeriNet-Based Segmentation

Our novel segmentation method works by exploiting word-to-word relations extracted from DeriNet (Zabokrtský et al., 2016), a network of Czech lexical derivations, and MorfFlex (Hajič and Hlaváčová, 2013), a Czech inflectional dictionary. We unify the two resources by taking all lemmas and forms from MorfFlex as vertices in a graph, connecting the forms with their respective lemmas with edges, and connecting lemmas with one another using edges found in DeriNet.

The segmentation algorithm first detects the longest common substrings along each edge (between each pair of connected words). This segment each word by two splits into a (potentially empty) prefix, the common substring and a (potentially empty) suffix. Each word may have multiple such segmentations, because it may have more than one word connected to it by an edge. After we collect common substrings for all edges in a tree, we propagate the found splits to other words in the tree by copying those that are inside a common substring to the other end of an edge. The splits of a word are a union of all the splits for all its incident edges and the mapped splits from common substrings shared with the neighboring words.

For example, the edge “mávající”→”mávající” has the longest common substring of “máv”, making the splits “mává” and “mává”. The edge “mává” (to be waving) → “máváčí” (waving) has the longest common substring of “máv”, making the splits “mává” and “máváčí”. The segmentation of “máváčí” is therefore “máváčí”. We can also split “máváčí” further using the other split in “máváčí” thanks to it lying in the longest common substring “máváčí”. The segmentation of “máváčí” is therefore “máváčí”.

The above example also shows the limitations of this method – the words are split too eagerly, resulting in many single-character splits. The boundaries between morphemes are fuzzy in Czech because connecting phonemes are often inserted and phonological changes occur. These cause spurious or misplaced splits. For example, the single-letter morph a in máváčí and máváčí doesn’t carry any information useful in machine translation and it would be better if we could detect it as a phonological detail and leave it connected to one of the neighboring morphs.

Table 4 presents several combinations of linguistically motivated and non-motivated segmentation methods. Since the vocabulary size after Morfessor or DeriNet splitting alone often remains too high, we further split the corpus with BPE or STE. Unfortunately, none of the setups performs better than the STE baseline.

6 Conclusion

We experimented with common linguistically non-informed word segmentation methods BPE and SubwordTextEncoder, and with two linguistically-motivated ones. Neither Morfessor nor our novel technique relying on DeriNet, a derivational dictionary for Czech, help. The unified methods thus remain the best choice.

Our analysis however shows an important difference in STE and BPE, which leads to considerably better performance. The same feature (support for zero suffix) can be utilized in BPE, giving similar gains.
References


A.24 KIT-Multi: A Translation-Oriented Multilingual Embedding Corpus

KIT-Multi: A Translation-Oriented Multilingual Embedding Corpus
Thanh-Le Ha, Jan Niehues, Matthias Sperber, Ngoc Quan Pham and Alexander Waibel
Karlsruhe Institute of Technology, Karlsruhe, Germany
firstname.lastname@kit.edu

Abstract
Cross-lingual word embeddings are the representations of words across languages in a shared continuous vector space. Cross-lingual word embeddings have been shown to be helpful in the development of cross-lingual natural language processing tools. In case of more than two languages involved, we call them multilingual word embeddings. In this work, we introduce a multilingual word embedding corpus which is acquired by using neural machine translation. Unlike other cross-lingual embedding corpora, the embeddings can be learned from significantly smaller portions of data and for multiple languages at once. An intrinsic evaluation on monolingual tasks shows that our method is fairly competitive to the prevalent methods but on the cross-lingual document classification task, it obtains the best figures. We are in the process to produce the embeddings for more languages, especially the languages which belong to the same family or semantically close to each others, such as Japanese-Korean, Chinese-Vietnamese, German-Dutch, or Latin-based languages. Furthermore, the corpus is being analyzed regarding its usage and usefulness in other cross-lingual tasks.

Keywords: multilingual embeddings, cross-lingual embeddings, neural machine translation, multi-source translation

1. Introduction
Inducing cross-lingual word embeddings is essentially acquiring word embeddings in different languages. The cross-lingual word embeddings can then be used as pre-trained models in cross-lingual applications such as cross-lingual document classification, information retrieval, textual entailment and question answering. Cross-lingual word embeddings can also help to perform transfer learning from a well-resource language to another low-resource language on various tasks, e.g. in building WordNet or annotating semantic relations.

There have been various methods of cross-lingual embedding induction being proposed, but most of them are essentially bilingual in the perspective that they learn to induce bilingual embeddings from bilingual data. Basically these methods optimize some cross-lingual constraints so that the semantic similarity between words corresponds to the closeness of these representations in a common vector space. Consequently, if they need cross-lingual embeddings for a new language pair, they must apply their inducing method on that new bilingual data. Furthermore, there would be some domain mismatch between the new acquired embeddings and the others if the new bilingual data are from different domain. The aforementioned limitations of those cross-lingual corpora motivates us to design a multilingual embedding inducing method from a single corpus which is available in as many languages as possible.

In this paper, we propose such an approach utilizing a multilingual neural machine translation (NMT) system to constrain the embeddings from $n$ source languages while translating into the same target language (as we call it multi-source NMT). The source embeddings employed in this model are implicitly forced to learn the common semantic regularities in order to maximize the translation quality of every language pair in the system. Once the multi-source NMT model is trained to a good state, the source word embeddings can be simply extracted from the model and used as a multilingual word embeddings.

The contribution of this work is the introduction of a method and its product corpus, KIT-Multi, consisting multilingual word embeddings of English-German-French. Other languages such as Chinese, Japanese, Korean, Vietnamese, Dutch, Italian, Romanian, Spanish or Portuguese are being added. We conducted some preliminary evaluations on KIT-Multi and compares to other cross-lingual embedding corpora. It has been shown that our multilingual corpus achieves competitive performances in standard evaluations as well as it has better coverage while using much less data for the training process. The evaluations on other languages would be published in the final version of the paper.

![Multi-source Neural Machine Translation](image)

Figure 2: Multi-source Neural Machine Translation system and how to get multilingual word embeddings from it.

---

1. For a thoroughly review of the most popular and advantageous techniques of cross-lingual word embedding induction, please refer to Upadhyay et al. (2016). For even more detailed and broader survey, please refer to Ruder (2017).
2. Multilingual Word Embedding Corpus

2.1. Embedding Induction Method

A neural machine translation system (Bahdanau et al., 2014) consists of an encoder representing a source sentence and an attention-based decoder that produces the translated sentence. One of the most notable differences of NMT compared to the conventional statistical approach is that the source words can be represented in a continuous space (i.e. word embeddings) in which the semantic regularities are induced automatically. Being applied to multilingual settings, NMT systems have been proved to be benefited from additional information embedded in a common semantic space across languages (Johnson et al., 2016; Ha et al., 2016; Curey et al., 2017). An interesting and positive side effect of such a system is the simultaneous induction of multilingual embeddings from the source side.

In a multi-source NMT systems where the sentences from several source languages are translated to one target language, the source embeddings are tied to a common semantic space across languages. So the source embeddings has its inherent cross-lingual characteristics, which could be extremely helpful for the cross-lingual applications employing the embeddings. More specifically, in our previous work on multi-source NMT (Ha et al., 2016), the words in each source sentence are coded with the language of that sentence before feeding to the training process of a standard neural machine translation system.

For example, the source sentence in English: they have since abandoned that project would become en_they en_have en_since en_abandoned en_that en_project. language coding is conducted in the preprocessing phase. Our multilingual embeddings are the derived product of this multi-source system. The figure 2 describes the process.

2.2. KIT-Multi Corpus

Our corpus is induced from WIT3’s TED subtitle corpora (Cetolo et al., 2012) including bilingual corpora from French, German, Dutch, Italian and Romanian to English. TED is a much smaller multilingual data compared to Europarl and contains other languages than European languages. The multi-source NMT is trained using the NMT framework OpenNMT (Klein et al., 2016) to translate from aforementioned languages (including English) to the only target language English. The statistics of TED bilingual corpora and our multilingual embedding corpus are shown in Table 1 and 2, respectively.

<table>
<thead>
<tr>
<th>Language pairs</th>
<th>Number of sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>German-English</td>
<td>196794</td>
</tr>
<tr>
<td>French-English</td>
<td>195025</td>
</tr>
<tr>
<td>Dutch-English</td>
<td>230866</td>
</tr>
<tr>
<td>Italian-English</td>
<td>220812</td>
</tr>
<tr>
<td>Romanian-English</td>
<td>210402</td>
</tr>
</tbody>
</table>

Table 1: Statistics of pair-wise TED bilingual corpora

*http://opennmt.net
t-SNE and projected to the 2D space using embeddings extracted from the multi-source NMT system induction.

Figure 1 illustrates the visualization of multilingual word embeddings extracted from the multi-source NMT system and projected to the 2D space using t-SNE (Maaten and Hinton, 2008). It shows how different words in different languages, i.e. English-German-French, can be close in the shared semantic space after being trained to translate into a common language (English).

Table 3 shows the closest words in the semantic space based on Cosine similarity with respect to some examples. We also include the language codes to clarify the origin of each word. From the table, we can see that the most close words are actually the words having the same meaning but in other languages.

Table 2: The size of the KIT-Multi embedding corpus

<table>
<thead>
<tr>
<th>Languages</th>
<th>Number of entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>21001</td>
</tr>
<tr>
<td>French</td>
<td>25685</td>
</tr>
<tr>
<td>German</td>
<td>24182</td>
</tr>
<tr>
<td>Dutch</td>
<td>24167</td>
</tr>
<tr>
<td>Italian</td>
<td>23422</td>
</tr>
<tr>
<td>Romanian</td>
<td>25505</td>
</tr>
</tbody>
</table>

Table 2: The size of the KIT-Multi embedding corpus

3. Preliminary evaluation of Cosine similarity

In this section, we describe some initial evaluation of our multilingual embedding corpus over some standard intrinsic and extrinsic evaluations, in comparisons with some other popular approaches for cross-lingual word embedding induction.

We mostly follow the experimental layout and settings of Upadhyay et al. (2016), conducting intrinsic and extrinsic evaluations on three European languages: English, French and German. The intrinsic evaluation is the monolingual word similarity task. The extrinsic evaluation focuses on the cross-lingual document classification. In this task, a document classifier is trained on a training set composed by a language $L_1$ and then predict the test set which is in the different language $L_2$. The process is then reversed for the language pair, and the classification accuracy is used to judge the quality of the cross-lingual embeddings. The corpora chosen to be compared are the corpora induced by Skip - Bilingual Skip-gram (Luong et al., 2015), CVM - Bilingual Compositional Model (Hermann and Blunsom, 2014) and VCD - Bilingual Vectors from Comparable Data (Vuic and Moens, 2015), which are all trained on much bigger Europarl v7 parallel corpora (Koehn, 2005). To show the impact of the corpus size, we also train the Bilingual Skip-gram embeddings with the same corpora used to train our model, and name it Skip-TED. For the details of those methods, please refer to Upadhyay et al. (2016).

In the intrinsic monolingual evaluation, we consider the word embeddings in one language at a time, i.e. the monolingual word embeddings, in order to conduct the word similarity. The Spearman’s rank correlation coefficient (Myers et al., 1995) between system similarity and human is the measure to judge the quality of the induced word embeddings. The English evaluation datasets are SimLex999 (En-999) and WordSim353 (En-353), in which the former (Hill et al., 2016) is claimed to better capture the similarity rather than both similarity and relatedness like in the latter (Finkelstein et al., 2002). The German (De) and French (Fr) datasets are the WordSim353 counterparts (Camacho-Collados et al., 2015; Leviant and Reichart, 2015).

The scores in Table 4 show that our word embeddings are competent in terms of monolingual aspect even though they are not trained to be adapted to monolingual quality. Moreover, our word embeddings perform better than the Skip embeddings trained on the same data by a large margin.

As shown in Table 5, the classifiers trained on our embeddings achieve highest accuracy on both directions of English⇔German, considerably better than other approaches. It is notable that, our model is trained on a substantially smaller corpus.

4. Related Work and Discussion

In (Upadhyay et al., 2016), the most popular and advantageous techniques for multilingual word embedding induction have been thoroughly evaluated. Corpora induced by Skip and VCD are the methods having the capability of monolingual adaptation by adjusting a hyper-parameter (in Skip models) or the portion of texts in each language (in VCD models). Furthermore, since they are designed based on the skip-gram models (Mikolov et al., 2013), it is unsurprising that they perform well on monolingual tasks. Corpora induced by CVM and our KIT-Multi, in contrast, are designed with cross-lingual orientation so that they focus more on similarity instead of relatedness. Aforementioned, our KIT-Multi corpus has shown its potential by achieving high accuracies on the task despite being induced from a significantly smaller corpus. Compared to the corpora acquired by their method, our embedding inherently induced in multilingual settings, with an arbitrary number of source and target languages, instead of being limited to bilingual. Those advantages allow us to extend our corpus seamlessly to many languages using small multilingual corpora, ideally from TED talks.

Table 3: Top 5 closest words by Cosine similarity.

<table>
<thead>
<tr>
<th>@en@research</th>
<th>Cosine Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>@de@Forschung</td>
<td>0.727675</td>
</tr>
<tr>
<td>@fr@recherches</td>
<td>0.697122</td>
</tr>
<tr>
<td>@de@Forschungs</td>
<td>0.671166</td>
</tr>
<tr>
<td>@fr@recherche</td>
<td>0.643990</td>
</tr>
<tr>
<td>@de@geforscht</td>
<td>0.637604</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>@en@humanity</th>
<th>Cosine Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>@de@Menschlichkeit</td>
<td>0.691524</td>
</tr>
<tr>
<td>@fr@humanité</td>
<td>0.684639</td>
</tr>
<tr>
<td>@de@Menschheit</td>
<td>0.645123</td>
</tr>
<tr>
<td>@de@Menschheit</td>
<td>0.634902</td>
</tr>
<tr>
<td>@en@mankind</td>
<td>0.621472</td>
</tr>
</tbody>
</table>

Table 4: Word Cosine Similarity
5. Conclusion and Future Work

In this proposal, we introduce a method to extract multilingual embedding corpus and its production, KIT-Multi. We would like to extend it for more languages as well as more cross-lingual natural language processing applications. The corpus will be available in Japanese, Korean, Chinese, Vietnamese, English, German, Dutch, Italian, French, Spanish and Portuguese at the time of the conference. We welcome other groups download and use it in other tasks and discuss about its usefulness.

6. Bibliographical References


CUNI NMT System for WAT 2017 Translation Tasks

Tom Kocmi  Dušan Vareš  Ondřej Bojar
Charles University,  
Faculty of Mathematics and Physics  
Institute of Formal and Applied Linguistics  
{kocmi, vareš, bojar}@ufal.mff.cuni.cz

Abstract

The paper presents this year’s CUNI submissions to the WAT 2017 Translation Task focusing on the Japanese-English translation, namely Scientific papers subtask, Patents subtask and Newswire subtask. We compare two neural network architectures, the standard sequence-to-sequence with attention (Seq2Seq) (Bahdanau et al., 2014) and an architecture using convolutional sentence encoder (FBConv2Seq) described by Gehring et al. (2017), both implemented in the NMT framework Neural Monkey that we currently participate in developing. We also compare various types of preprocessing of the source Japanese sentences and their impact on the overall results. Furthermore, we include the results of our experiments with out-of-domain data obtained by combining the corpora provided for each subtask.

1 Introduction

With neural machine translation (NMT) currently becoming the leading paradigm in the field of machine translation, many novel NMT architectures with state-of-the-art results are being proposed. In the past, there were reports on large scale evaluation (Britz et al., 2017), however, the experiments were performed on a limited number of language pairs from related language families (English—German, English—French) or focused on a subset of possible NMT architectures, leaving room for further exploration.

One of the downsides of NMT is the limited allowable size of both input and output vocabularies. Various solutions for dealing with potential out-of-vocabulary (OOV) tokens were proposed either by using a back-off dictionary look-up (Luong et al., 2015), character-level translation of unknown words (Luong and Manning, 2016) or recently quite popular translation via subword units generated by byte pair encoding (Sennrich et al., 2016c). However, in the case of Japanese which has no clear definition of a word unit, there has been less research on how a particular preprocessing can influence the overall NMT performance.

In this system description paper we compare two sequence-to-sequence architectures, one using a recurrent encoder and one using a convolutional encoder. We also report results of our experiments with preprocessing of Japanese. Furthermore, we report how including additional out-of-domain training data influence the performance of NMT.

2 Dataset Preparation

In this section we describe the methods we used for preprocessing both Japanese and English.

Due to Japanese being an unsegmented language with no clear definition of word boundaries, proper text segmentation is essential. We used MeCab\(^2\) (Kudo et al., 2004) with the UniDic\(^3\) dictionary to perform the tokenization.

For English, we used morphological analyser MorphoDiTa\(^4\) (Straková et al., 2014) to tokenize English training sentences. Based on the generated lemmas, we also performed truecasing of the target side of the training data.

To reduce the vocabulary size, we use byte pair encoding (BPE; Sennrich et al., 2016c) which breaks all words into subword units. The vocabulary is initialized with all alphabet characters

---

3. https://osdn.net/projects/unidico/
present in the training data and larger units are added on the basis of corpus statistics. Frequent words make it to the vocabulary, less frequent words are (deterministically) broken into smaller units from the vocabulary. We generated separate BPE merges for each dataset, both source and target side.

Because the BPE algorithm, when generating the vocabulary, performs its own (subword) segmentation, we decided to compare a system trained on the tokenized Japanese (which was then further segmented by BPE) with a system that was segmented only via BPE. Additionally, we also performed a comparison with a system with Japanese text transcribed in Latin alphabet. The romanization was done by generating Hiragana transcription of each token using MeCab and then transcribing these tokens to Romaji using jaconv. The resulting text was then also further segmented by BPE. The results are discussed in Section 4.1

3 Architecture Description

We use Neural Monkey\(^6\) (Helcl and Libovický, 2017), an open-source NMT and general sequence-to-sequence learning toolkit built using the TensorFlow (Abadi et al., 2015) machine learning library.

Neural Monkey is flexible in model configuration supporting the combination of different encoder and decoder architectures as well as solving various tasks and metrics.

We perform most of the experiments on the 8GB GPU NVIDIA GeForce GTX 1080. For the preprocessing of data and final inference, we use our cluster of CPUs.

The main hyperparameters of the neural network are set as follows. We use the batch size of 60. As the optimization algorithm we use Adam (Kingma and Ba, 2014) with initial learning rate of 0.0001. We used only the non-ensembled left-to-right run (i.e. no right-to-left rescoring as done by Sennrich et al. 2016a) with beam size of 20, taking just the single-best output.

We limit the vocabulary size to 30,000 subword units. The vocabulary is constructed separately for the source and target side of the corpus.

We compare two different architectures. We describe both of them in more details as well as the hyperparameters used during the training in the following sections.

3.1 Sequence to Sequence

Our main architecture is the standard encoder-decoder architecture with attention as proposed by Bahdanau et al. (2014).

The encoder is a bidirectional recurrent neural network (BiRNN) using Gated Recurrent Units (GRU; Cho et al., 2014). In each step, it takes an embedded token from the input sequence and its previous output and outputs a representation of the token. The encoder works in both directions; the resulting vector representations at corresponding positions are concatenated. Additionally, the final outputs of both the forward and backward run are concatenated and used as the initial state of the decoder.\(^7\)

The decoder is a standard RNN with the conditional GRU (Calixto et al., 2017) recurrent unit. At each decoding step, it takes its previous hidden state and the embedding of the token produced in the previous step as the input and produces the output vector. This vector is used to compute the attention distribution vector over the encoder outputs. The RNN output and the attention distribution vector are then used as the input of a linear layer to produce the distribution over the target vocabulary. During training, the previously generated token is replaced by the token present in the reference translation. The architecture overview is in Figure 1.

We have used the following setup of network hyperparameters. The encoder uses embeddings of size 500 and the hidden state of bidirectional GRU network of size 600 in both directions. Dropout (Srivastava et al., 2014) is turned off and the maximum length of the source sentence is set to 50 tokens. The size of the decoder hidden state is 600 and the output embedding is 500. In this case, dropout is also turned off. The maximum output length is 50 tokens. In this paper, we will refer to this architecture as Seq2Seq.

3.2 Convolutional Encoder

The second architecture is a hybrid system using convolutional encoder and recurrent decoder. We use the convolutional encoder defined by Gehring et al. (2017). First, the input sequence

\(^5\)https://github.com/ikegami-yukino/jaconv
\(^6\)http://ufal.mff.cuni.cz/neuralmonkey
\(^7\)The concatenated final states are transformed to match the size of the decoder hidden state.
of tokens $x = (x_1, ..., x_n)$ is assigned a sequence of embeddings $w = (w_1, ..., w_n)$ where $w_i \in \mathbb{R}^f$ is produced by embedding matrix $D \in \mathbb{R}^{V \times f}$. When compared to the RNN encoder, the convolutional encoder does not explicitly model positions of the tokens in the input sequence. Therefore, we include this information using positional embeddings. We model the information about the position in the input sequence via $p = (p_1, ..., p_n)$ where $p_i \in \mathbb{R}^f$. The resulting input sequence embedding is computed as $e = (w_1 + p_1, ..., w_n + p_n)$.

The encoder is a convolutional network stacking several convolution blocks over each other. Each block contains a one dimensional convolution followed by a nonlinearity. The convolution with kernel size $k$ and stride 1 with SAME padding is applied on the input sequence using $d$ input channels and $2 \times d$ output channels. This output is then fed to the Gated Linear Unit (GLU; Dauphin et al., 2016) which substitutes a nonlinearity between the convolution blocks. Additionally, residual connections are added to the produced output. At the final layer, we get the encoded sequence $y = (y_1, ..., y_m)$ where $y_i \in \mathbb{R}^d$.

We use same decoder as in the previous section. The initial decoder state $s \in \mathbb{R}^d$ is created by picking element-wise maximum across the length of the encoder output sequence $y$. We tried other methods for creating the initial decoder state and this one produced the best results. Figure 2 shows the overview of the encoder architecture.

In the experiments we use encoder with the embedding size of 500 and maximum length of 50 tokens per sentence. The encoder uses 600 input features in each of its 6 convolutional layers with the kernel size of 5. Dropout is turned off. For the rest of this paper we will refer to this architecture as $\text{FBConv2Seq}$.

4 Experiments

In this section we describe all experiments we conducted for the WAT 2017 Translation Task. We report results over the development set.

4.1 Japanese Tokenization

We experimented with various tokenization methods of the Japanese source side. In Table 1 we compare untokenized, tokenized and romanized Japanese side. This experiment was evaluated over the top 1 million training examples in the ASPEC dataset.

4.2 Architecture Comparison

In Table 2 we compare the architectures we described in Section 3. We ran experiments on 4 different datasets. The JPO, JIJI, ASPEC with 1 million best sentences were used with tokenized
Table 1: Comparison of various tokenization methods measured on the ASPEC dataset.

Table 2: Comparison of two examined architectures.

Table 3: Comparison of in-domain data only and combined corpora.

Japanese. The dataset ASPEC 3M was not tokenized by MeCab.

After examination of the Table 2, we can see that in most cases the Seq2Seq model (Bahdanau et al., 2014) outperforms the FBConv2Seq architecture. On the other hand, the FBConv2Seq model performed better on the untokenized corpus. This might suggest that the model has an advantage in processing inputs which are not properly segmented thanks to the convolutional nature of the encoder. This could be valuable for languages that cannot be segmented.

4.3 ASPEC Size of Data

The ASPEC dataset consists of 3 millions of English to Japanese sentence pairs ordered with a decreasing accuracy of the translation. It is a well known fact about neural networks that the more data is available, the better performance they can get. In this experiment we try to compare the influence of the size of dataset and the quality of the training pairs. We decided to experiment with sub-corpora containing 1, 1.5, 2, 2.5 and 3 million best sentence pairs. We refer to them as ASPEC 1M, ASPEC 1.5M, ASPEC 2M, ASPEC 2.5M, ASPEC 3M respectively.

For simplicity, the experiment was performed with untokenized Japanese side and we used the Seq2Seq architecture. All corpora are shuffled in order to overcome the ordering by the quality of translation.

The results presented in Figure 3 show a clear picture that the overall quality of the training data is more important than the total amount of the data.

4.4 Corpus Combination

In the previous section, we experimented with the quality of the training corpora. In this experiment we show whether more data can help in various domains or if it is also a burden as shown in the previous section comparing quality of the data.

We combined tokenized corpora for JPO (1 million sentences), JIJI (200 thousand sentences) and 2 million of the best sentences from ASPEC. The resulting corpus was shuffled.

The results in Table 3 suggest that the domain is important for both the JPO and JIJI datasets. Interestingly, it improved the score of the ASPEC 2M.

There is also another explanation which is more plausible with respect to the experiments in the previous section. The training data in JPO and JIJI have better quality than the data in ASPEC 2M, which leads to the worse performance on those datasets and on the other hand cleaner data helps...
Figure 3: Learning curves over different sizes of ASPEC data.

Table 4: Performance of the final models on the development data.

<table>
<thead>
<tr>
<th>Corpora</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPO</td>
<td>35.35 BLEU</td>
</tr>
<tr>
<td>JII</td>
<td>16.40 BLEU</td>
</tr>
<tr>
<td>ASPEC</td>
<td>25.56 BLEU</td>
</tr>
</tbody>
</table>

ASPEC to increase the performance.

More research on this topic is needed to answer which of the explanations is more plausible. In future work, we want to experiment with combined corpora of JPO, JII and only 1 million of the cleanest sentences from ASPEC.

4.5 Official Results

Based on the previous experiments we decided to use the tokenized and shuffled in-domain training data for each of the tasks. For the Translation Task submission, we chose the Seq2Seq architecture, because it had a better overall performance. For the ASPEC dataset, we decided to train only on the 1 million cleanest training data. The results of the evaluation done on the corresponding development datasets are in Table 4.

The results of Translation Task are available on the WAT 2017 website.\(^9\) Our system performed mostly on average. It was beaten by more sophisticated architectures using more recent state-of-the-art techniques.

5 Summary

In this system description paper, we presented initial results of our research in Japanese-English NMT. We compared two different architectures implemented on NMT framework, Neural Monkey, however, as the official results of the WAT 2017 Training Task suggest, future improvements needs to be done to catch-up with the current state of the art.

We performed experiments with different input language tokenization combined with the byte-pair-encoding subword segmentation method. In the future, we plan to explore other tokenization options (e.g. splitting to bunsetsu) together with using a shared vocabulary for both the source and target languages. We are curious, whether the latter will bring an improvement when combined with romanization of Japanese.

Lastly, we made several experiments with dataset combination suggesting that including additional out-of-domain data is generally harmful for the NMT system. As the next step we plan to investigate options for creating additional synthetic data and their impact on the overall performance as suggested by Sennrich et al. (2016b).

Acknowledgments

This work has been in part supported by the European Union’s Horizon 2020 research and innovation programme under grant agreements No 644402 (HimL) and 645452 (QT21), by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16_013/0001781), by the Charles University Research Programme “Progres” Q18+Q48, by the Charles University SVV project number 260 453 and by the grant GAUK 8502/2016.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Cor-


