QT21

Deliverable D1.1

Semantic Translation Models

Ondřej Bojar (CUNI), Qun Liu (DCU), Yvette Graham (DCU),
Jan Niehues (KIT), Stella Frank (UvA),

Dissemination Level: Public

31st July, 2016
Grant agreement no.          645452
Project acronym             QT21
Project full title          Quality Translation 21
Type of action              Research and Innovation Action
Coordinator                Prof. Josef van Genabith (DFKI)
Start date, duration        1st February, 2015, 36 months
Dissemination level         Public
Contractual date of delivery 31st July, 2016
Actual date of delivery     31st July, 2016
Deliverable number         D1.1
Deliverable title           Semantic Translation Models
Type                       Report
Status and version          Final (Version 1.0)
Number of pages             88
Contributing partners       CUNI, KIT, DCU
WP leader                   CUNI
Author(s)                   Ondřej Bojar (CUNI), Qun Liu (DCU), Yvette Graham (DCU),
                            Jan Niehues (KIT), Stella Frank (UvA),
EC project officer          Susan Fraser

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/

© 2016, 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 4
2 Analysis and Developments of Deep Transfer-Based MT 5
3 Graph-Based Translation via Graph Segmentation 6
4 Factored Word Representation in Neural Network Models 6
5 Word Factors in Pure Neural MT Models 6
6 Cross-Lingual Vector Representations of Text 7
7 Multimodal Translation 7
References 8
Appendices 9
Appendix A What a Transfer-Based System Brings to the Combination with PBMT 9
Appendix B CUNI-LMU Submissions in WMT2016: Chimera Constrained and Beaten 19
Appendix C Graph-Based Translation Via Graph Segmentation 25
Appendix D Using Factored Word Representation in Neural Network Language Models 36
Appendix E Linguistic Input Features Improve Neural Machine Translation 45
Appendix F Towards Cross-Lingual Distributed Representations without Parallel Text 54
Appendix G Multilingual Image Description with Neural Sequence Models 59
Appendix H DCU-UvA Multimodal MT System Report 75
Appendix I CUNI at Post-editing and Multimodal Translation Tasks 80
1 Executive Summary

This deliverable reports on the progress during the first 18 months of the project in Task 1.1 Semantic Translation Models, as specified in the project proposal:

Task 1.1 Semantic Translation Models [M01–M36] (CUNI, KIT, DCU) This task will focus on “deep” and abstract syntactic and especially semantic representations of the relevant information, exploring deep language analysis as well as graph-based translation methods for such complex representation and a number of related topics, such as general feature selection and use, use of content reference, and the capture of surface- to-semantic relations.

The work of the consortium on translation models that directly handle the meaning of the sentence can be divided into symbolic approaches and neural approaches, with a very clear cut. Symbolic approaches stem from the tradition of formal linguistic representation of sentences and build upon concepts from graph theory. One of their main characteristics is that they try to expose the meaning, break it down into smaller pieces and explicitly rely on meaning compositionality (or the lack of it in known situations). Neural approaches to machine translation, on the other hand, are motivated by the rationale that the meaning of the sentence is never directly visible on the surface. They treat semantics as a hidden variable and rely on computationally-intensive methods of machine learning (ML) to learn to predict the translation of the sentence directly from the source. The neural network (NN) is structured and trained in a way that allows it to automatically find and make use of semantic links and relatedness of input words, and to construct fluent sentences by producing one word at a time, whilst internally remembering which components of the meaning were already produced and which should come next. The trained model of a neural MT system contains an internal representation of semantics different to the formal representation of semantics defined by linguists. As a consequence, manipulation of NN internal abstraction layers has to be addressed with different methods and tools as those used by linguists, and many such methods and tools have yet to be discovered.

Note that the applicability of neural network methods to the task of MT as a whole (i.e. beyond serving in individual components such as the language model, or as a generic ML technique) is rather new, and builds upon very recent advances in core NN structure and training techniques. This success is so new and not really anticipated that the structure of tasks in this project does not perfectly match this novel state of the art. In our work, we respond to this change accordingly, but the reporting in our deliverables may not follow the task descriptions as closely as we originally planned. Also, there is a clear overlap of topics within WP1 and also towards other work packages. Specifically, symbolic approaches have a lot of common with work in Task 1.2 (see Deliverable 1.3) and neural approaches are very close to full structural prediction planned in Task 1.3 (Deliverable 1.5).

For symbolic approaches, CUNI continued the development and application of its transfer-based system, with the transfer at a deep linguistic representation of the sentence, see Section 2.

DCU has been working on graph-based machine translation approaches. As an early step towards this direction, DCU proposed a graph-based model to combine the phrase-based translation model and the dependency-based translation model which outperformed German-English and Chinese-English baseline models by up to +0.5/+1.5 BLEU points, details are provided in Section 3.

For neural approaches, partners in the consortium have been exploring methods that could provide these models with explicit linguistic markup. KIT developed a factored representation for RNN-based language models (Section 4). This was used to add information into the model and was applied to morphologically-rich languages in cooperation with WP2. UEDIN developed a technically similar solution for their neural MT system which improves translation quality on English⇒German WMT16 translation task by up to 1.5 BLEU points, see Section 5.

In search for representations of words that would simplify the transfer of word meanings, UEDIN proposes a method that learns such representations cross-lingually, but relying only on monolingual texts on similar topics, see Section 6.
Finally, tackling the far end of semantics, several partners in the consortium experiment with grounding. Many expressions of natural language remain semantically ambiguous until some extra-linguistic information is provided. An appealing proxy to this real-world grounding can be provided by images. In Section 7, we report on the consortium's work in this area.

2 Analysis and Developments of Deep Transfer-Based MT

Linguistically deep MT systems are hardly ever applicable to large and unconstrained translation tasks. TectoMT, the deep transfer-based system developed at CUNI, is no exception but it is known to significantly improve a high-performing phrase-based system, if used to supply novel phrases to it.

As reported in D1.6, CUNI carried out an analysis of their system Chimera, a tight integration of TectoMT and a factored setup of Moses.

The analysis carried out during the first 9 months of QT21 covered various aspects of the integration, its errors, and the contribution of individual components. We analyzed which n-grams required by the reference translations were produced by each of Chimera's individual components. We also looked at the sources of phrase pairs applied during the translation process. We manually analyzed how many phrases proposed by Moses and TectoMT are correct (in any context). We also looked separately at morphological errors and their distribution over parts of speech. We analyzed how the components complement each other – specifically, we measured how many sentences could be produced by Moses without TectoMT and we investigated the interplay between TectoMT and the strong language models (LMs) included in Chimera.

Overall, we found that the transfer-based component TectoMT provides many novel translations (unavailable to Moses) but at the expense of noise. Moses can mostly select the useful translations thanks to its LMs. Furthermore, good translations which could be reached by Moses are often not produced without TectoMT due to modelling errors. Our combination, while somewhat cumbersome, has some appealing properties compared to standard system combination techniques, and the mix of phrase-based and transfer-based system leads to improved morpho-syntactic coherence of translation and enables better morphological generalization.

In 2016, Chimera again took part in WMT News Translation Task. This time, the system was trained constrained, to allow for a direct comparison with other competing systems. While Chimera was the top performing system of English-to-Czech in 2013, 2014 and 2015, it ended up in fourth place in 2016 outperformed by two NN-based systems and one phrase-based system with a very large number of model components and extensive use of word classes.

As of July 2016, CUNI plans to employ a discriminative model in the deep transfer and to:

- compare current MaxEnt model used in TectoMT with a new VowpalWabbit-based model,
- explore the advantages of one model for all lemmas (the current solution uses a separate MaxEnt model for each source lemma), e.g. label-dependent features for sharing knowledge between translation of different lemmas,
- explore the utility of a wide range of features (morphology, syntax, deep syntax, semantic) for selecting target word lemmas,
- experiment with domain adaptation using online learning (Beygelzimer et al., 2015).

More importantly, though, the phrase-based component of Chimera needs to be replaced with an NMT system, and a novel way of combining NMT with the deep transfer has to be devised and evaluated to see if the transfer-based systems are still beneficial for the new state of the art.

¹http://hunch.net/~vw/
The analysis of the Chimera system is available in Tamchyna and Bojar (2015) and its participation in WMT2016 translation task is described in Tamchyna et al. (2016). The papers can be found in Appendix A and Appendix B, respectively.

3 Graph-Based Translation via Graph Segmentation

DCU has been working on graph-based machine translation approaches. Existing MT approaches either use a sequence-based model (phrase-based model), or a tree-based model (hierarchical phrase-based model or syntax-based models). However, these models are not compatible with each other. Furthermore, most semantic representation (e.g. predicate-argument structures of a sentence) cannot be represented as a sequence or a tree.

The DCU team aims to investigate the idea of using graph as a unified framework to represent various linguistic structures. As an early step towards this direction, we proposed a graph-based model to combine the phrase-based translation model and the dependency-based translation model. We use directed graph to represent both the sequential structure and the dependency structure indiscriminately. The model segments an input graph into connected subgraphs, each of which may cover a discontinuous phrase. Beam search is conducted to combine translations of each subgraph left-to-right to produce a complete translation. Experiments on Chinese-English and German-English datasets show that this approach leads to significantly better results than the phrase-based model, by up to +1.5/+0.5 BLEU points.

The work will appear as a long paper in ACL 2016 (Li et al., 2016) and it is available in Appendix C.

4 Factored Word Representation in Neural Network Models

Neural network language and translation models have recently shown their great potential to improve the performance of phrase-based machine translation. At the same time, word representations using different word factors have been used in many state-of-the-art machine translation systems, in order to support better translation quality.

KIT combined these two ideas. 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, as evaluated on three language pairs in WMT16 news task (improvements around 0.7 BLEU points). This work was done in collaboration with WP2, where this technique was used to improve the modelling of morphologically rich languages like German and Romanian.

Furthermore, this technique allows an easy integration of additional knowledge. In the work so far, KIT added source side information to the original language model.

A detailed description of the work can be found in Niehues et al. (2016), and is provided in Appendix D.

5 Word Factors in Pure Neural MT Models

Beyond their application in improving statistical machine translation, neural models have achieved impressive results when used as standalone translation models. In recent shared tasks (including the QT21 news translation task), pure neural approaches have even surpassed statistical models for some language pairs. So far these models have used little in the way of external linguistic information, but it is an open question whether linguistic information should be used, and if so, how.

In this work, UEDIN 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. UEDIN generalize the embedding layer of the encoder in the attentional encoder-decoder architecture to support the inclusion of arbitrary factors, in addition to the
basic word factor. In this first study, UEDIN add morphological features, part-of-speech tags, and syntactic dependency labels as input factors to English→German and English→Romanian neural machine translation systems. In experiments on WMT16 training and test sets, we found that linguistic input factors improve model quality according to three metrics: perplexity, BLEU and CHRF3. While the factors used in these initial experiments are semantically shallow, the approach is general enough to support the use of deeper semantic factors and this is planned for future work.

This work is described in Sennrich and Haddow (2016), which can be found in Appendix E. 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.

6 Cross-Lingual Vector Representations of Text

Distributed representations that map words, sentences, paragraphs or documents to vectors of real numbers have proven extremely useful for a variety of natural language processing tasks, including machine translation. These representations provide an effective way to inject in machine learning models general prior knowledge about language automatically obtained from inexpensive, non-annotated corpora. Based on the assumption that different languages share a similar semantic structure, various approaches have succeeded in obtaining distributed representations that are compatible across multiple languages, either by learning mappings between different embedding spaces or by jointly training cross-lingual representations. These approaches require some amount of parallel text, aligned at word level, sentence level or at least document level, or some other kind of parallel resources such as dictionaries.

In this work, UEDIN explores whether the assumption of a shared semantic structure between languages is strong enough that it allows the induction of compatible distributed representations without using any parallel resources. The proposed method only requires monolingual corpora that are thematically similar between languages in a general sense. In order to evaluate the hypothesis, UEDIN proposes a scheme to map word vectors trained on a source language to vectors semantically compatible with word vectors trained on a target language using an adversarial autoencoder. Preliminary qualitative results are presented for English→Italian, where it is observed that the closest Italian embeddings of transformed English embeddings are frequently either valid translations or appear to be semantically related. Possible future developments of this technique include applications to cross-lingual sentence representations.

This work is described in Michele Barone (2016), which can be found in Appendix F. 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.

7 Multimodal Translation

Translation based on visual cues from images has long been interesting theoretically as a way of grounding multilingual data in a non-linguistic semantic space. Thanks to recent breakthroughs in computer vision using deep convolutional neural networks, multimodal machine translation is now viable for some datasets (e.g. photographic images with natural, human-generated descriptions).

UvA introduced the task of multimodal translation in Elliott et al., 2015), together with initial models for this task based on recurrent neural network language models. UvA and USFD then collaborated in setting up a new shared task on multimodal machine translation and multilingual image description at WMT (see WP4) Specia et al., 2016, Elliott et al., 2016.

DCU and UvA collaborated with a submission Calixto et al., 2016), for the multimodal machine translation task. This system was based on neural machine translation with an attention
mechanism. The attention mechanism operating over the source language was enhanced with a second attention operating over the image. Adding visual attention led to improvements over an internal text-only baseline. However, the overall results on the task showed that text-only systems currently outperform multimodal translation systems, indicating the need for more work, which is ongoing at UvA.

Independently, CUNI took part in the multimodal translation task, developing their own NN-based system with attention over multiple source strings as well as the representation of the image, (Libovický et al., 2016).

The relevant papers with QT21 support are available as Appendices C through I.

References


Aleš Tamchyna and Ondřej Bojar. 2015. What a Transfer-Based System Brings to the Combination with PBMT. In Bogdan Babych, Kurt Eberle, Patrik Lambert, Reinhard Rapp, Rafael Banchs, and Marta Costa-Jussà, editors, Proceedings of the Fourth Workshop on Hybrid Approaches to Translation (HyTra), pages 11–20, Stroudsburg, PA, USA. Association for Computational Linguistics, Association for Computational Linguistics.

A What a Transfer-Based System Brings to the Combination with PBMT

What a Transfer-Based System Brings to the Combination with PBMT

Aleš Tamchyna and Ondřej Bojar
Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, Prague, Czech Republic
surname@ufal.mff.cuni.cz

Abstract
We present a thorough analysis of a combination of a statistical and a transfer-based system for English→Czech translation, Moses and TectoMT. We describe several techniques for inspecting such a system combination which are based both on automatic and manual evaluation. While TectoMT often produces bad translations, Moses is still able to select the good parts of them. In many cases, TectoMT provides useful novel translations which are otherwise simply unavailable to the statistical component, despite the very large training data. Our analyses confirm the expected behaviour that TectoMT helps with preserving grammatical agreements and valency requirements, but that it also improves a very diverse set of other phenomena. Interestingly, including the outputs of the transfer-based system in the phrase-based search seems to have a positive effect on the search space. Overall, we find that the components of this combination are complementary and the final system produces significantly better translations than either component by itself.

1 Introduction
Chimera (Bojar et al., 2013b; Tamchyna et al., 2014) is a hybrid English-to-Czech MT system which has repeatedly won in the WMT shared translation task (Bojar et al., 2013a; Bojar et al., 2014). It combines a statistical phrase-based system (Moses, in a factored setting), a deep-transfer hybrid system TectoMT (Popel and Žabokrtský, 2010) and a rule-based post-editing tool Depfix (Rosa et al., 2012).

Empirical results show that each of the components contributes significantly to the translation quality, together setting the state of the art for English→Czech translation. While the effects of Depfix have been thoroughly analyzed in Bojar et al. (2013b), the interplay between the two translation systems (Moses and TectoMT) has not been examined so far.

In this paper, we show how exactly a deep transfer-based system helps in statistical MT. We believe that our findings are not limited to our exact setting but rather provide a general picture that applies also to other hybrid MT systems and other translation pairs with rich target-side morphology.

The paper is organized as follows: Section 2 briefly describes the architecture of Chimera and summarizes its results in the WMT shared tasks. In Section 3, we analyze what the individual components of Chimera contribute to translation quality. Section 4 describes how the components complement each other Section 5 outlines some of the problems still present in Chimera and Section 6 concludes the paper.

2 Chimera Overview
Chimera is a system combination of a phrase-based Moses system (Koehn et al., 2007) with TectoMT (Popel and Žabokrtský, 2010), finally processed with Depfix (Rosa et al., 2012), an automatic correction of morphological and some semantic errors (reversed negation). Chimera thus does not quite fit in the classification of hybrid MT systems suggested by Costa-jussà and Fonollosa (2015).

Figure 1 provides a graphical summary of the simple system combination technique dubbed “poor man’s”, as introduced by Bojar et al. (2013b). The system combination does not need any dedicated tool, e.g. those by Matusov et al. (2008), Barrault (2010), or Heafield and Lavie (2010). Instead, it directly includes the output of the transfer-based system into the main phrase-based search.
At its core, Chimera is a (factored) Moses system with two phrase tables. The first is a standard phrase table extracted from English-Czech parallel data. The second phrase table is tailored to the input data and comes from a synthetic parallel corpus provided by TectoMT: the source sides of the dev and test sets are first translated with CU-TECTOMT. Following the standard word alignment on the source side and the translation, phrases are extracted from this synthetic corpus and added as a separate phrase table to the combined system (CH1). The relative importance of this phrase table is estimated in standard MERT (Och, 2003).

The final translation of the test set is produced by Moses (enriched with this additional phrase table) and additionally post-processed by Depfix.

Note that all components of this combination have direct access to the source side which prevents the cumulation of errors.

For brevity, we will use the following names: CH0 to denote the plain Moses, CH1 to denote the Moses combining the two phrase tables (one from CH0 and one from CU-TECTOMT), and CH2 to denote the final Chimera.

In this paper, we focus on the first two components, leaving CH2 aside. The rest of this section summarizes Chimera’s results in the last three years of WMT translation task and adds two technical details: language models used in 2015 and the effects of the default low phrase table limit.

2.1 Chimera and its Components in WMT

Table 1 shows the official BLEU scores and the results of manual evaluation (ranking) in the last three years of WMT. It illustrates the complementary value of each component in Chimera.

TectoMT by itself does not perform well compared to other systems in the task, it consistently achieves low BLEU scores and manual ranking. Moses by itself (CH0) achieves quite a high BLEU score but still significantly lower than CH1 (combination of the “poor” TectoMT and plain Moses). Depfix seems to make almost no difference in the automatic scores (once it even slightly worsened the BLEU score) but still significantly lower than CH2.

In 2014, Chimera would have lost to Edinburgh’s submission if it were not for Depfix.

An illustration of the complementary utility is given in Table 3. Both CH0 and CU-TECTOMT produce translations with major errors. CH1 is able to pick the best of both and produce a grammatical and adequate output, very similar to the reference translation. CH1 can also produce words which were not present in either output.

2.2 Language Models

In 2015, CHIMERA in all its stages used four language models (LMs), as summarized in Table 2. Two of the language models (“big” and “long”) are trained on surface forms (“stc” refers to surface forms, “tag” refers to tagger output, “morph” refers to morphological information, “longm” refers to long-term memory).
the living zone with the dining room and kitchen section in the household of the young couple.

<table>
<thead>
<tr>
<th>source</th>
<th>reference</th>
<th>CH0</th>
<th>CU-TECTOMT</th>
<th>CH1</th>
</tr>
</thead>
<tbody>
<tr>
<td>living zone with dining and kitchen section in household young couple.</td>
<td>obývací zóna s jídelnou a kuchyní v domácnosti mladého páru.</td>
<td>living zone with dining room and kitchen in section household young couple.</td>
<td>živa zóna pokojí s jídelnou a s kuchyňským oddílem v domácnosti mladého páru.</td>
<td>obývací prostor s jídelnou a kuchyní v domácnosti mladého páru.</td>
</tr>
</tbody>
</table>

Table 3: Example of translations by various stages of Chimera. Errors are in bold, glosses are in italics.

<table>
<thead>
<tr>
<th>System</th>
<th>Table limit</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH0</td>
<td>100 20</td>
<td>24.23 ±0.09</td>
</tr>
<tr>
<td>CH1</td>
<td>100 100</td>
<td>24.16 ±0.07</td>
</tr>
</tbody>
</table>

Table 4: Impact of phrase table limit for phrase tables coming from the parallel data (the column “CH0”) and from TectoMT.

2.3 Phrase Table Limit

Until recently we did not pay much attention to the maximum number of different translation options considered per source phrase (the parameter `table-limit`), assuming that the good phrase pairs are scored high and will be present in the list.

This year, we set `table-limit` to 100 instead of the default 20 and found that while it indeed made little or no difference in CH0, it affected the system combination in CH1. It is known that multiple phrase tables clutter the search space with different derivations of the same output (Bojar and Tammyna, 2011), demanding a relaxation of pruning during the search (e.g. `stack-limit` or the various limits of cube pruning). From this point of view, increasing the `table-limit` actually makes the situation worse by bringing in more options. We leave the search pruning limits at their default values, increase only the `table-limit`, and yet observe a gain.

Table 4 shows the average testset BLEU score (incl. the standard deviation) obtained in three independent runs of MERT when setting the `table-limit` to 20 or 100 for one or both phrase tables. Multeval (Clark et al., 2011) confirmed that the difference between 20 and 100 for both tables of CH1 (i.e. 24.00 vs. 24.16) is significant while the difference for the system CH0 is not. A part of this effect has to be attributed to the lower variance of CH1 MERT runs, indicating that the TectoMT phrase table somehow stabilizes the search.

This could be due to the longer phrases from TectoMT, see Section 3.1. The results also suggest that keeping the default limit for the TectoMT phrase table would have been an even better choice – perhaps because low scoring phrases from TectoMT are indeed mostly bad while the relaxed CH0 `table-limit` ensures that the necessary morphological variants of words are considered at all.

3 Contribution of Individual Components

Table 5 breaks n-grams from the reference of WMT14 test set into classes depending on by which Chimera components they were produced. The first column considers unigram tokens, the subsequent columns report n-gram types.

We see that 44.7 % of unigram tokens needed by the reference?
tokens were not available in any of these single-best outputs. For Czech as a morphologically rich target language, it is a common fact that a large portion of the output is not confirmed by the reference (and vice versa) despite not containing any errors (Bojar et al., 2010).

The poor man’s system combination method is essentially phrase-based, so it is not surprising that there are about twice as many unigrams that come from CH0 than from CU-TECTOMT, see 8.6 vs 4.5%. This bias towards PBMT gets more pronounced with longer n-grams (5.1 vs 1.5% for 4-grams). The number of n-grams needed by the reference and coming from either of the individual systems but not appearing in the combination (✓ - and ✓–) is comparable, around 3.5% of unigrams.

It is good news that we gain ~1.5% of n-grams as a side-effect: neither of the systems suggested them on its own but they appeared in the combination (✓–) Note that we see this positive effect also for unigrams, suggesting that our “poor man’s” system combination could in principle outperform more advanced techniques. The output of the secondary system(s) can help the main search to come up with better translation options. In the following, we refine the analysis of contributions of the individual components by finding where they apply and what they improve.

### 3.1 Sources of Used Phrase Pairs

In a separate analysis, we look at the translation of the WMT13 test set and the phrases used to produce it. Table 6 shows both phrase counts and average (source) phrase lengths (in words) broken down according to the phrase source. The test set was translated using 31961 phrases in total (“phrase tokens”), 21106 unique phrase pairs were available in both phrase tables.

The TectoMT phrase table provided 11706 phrase types in total, 3503 of these were unique, i.e. not present in the phrase table extracted from the parallel data. (See Section 4.1 below for the reachability of such phrases on the WMT14 test set.) Given the total number of phrase types, this is a small minority (roughly 17%), however these phrases correspond directly to our test set and the benefit is visible right away: the average phrase length of these unique phrases is much higher (3.73) which allows the decoder to cover longer parts of the input by a single phrase. We believe that such phrases help preserve local (morphological) agreement and overall consistency of the translation.1

As expected, the average length of the shared phrase pairs (present in both phrase tables) is short and this is even more prominent when we look at tokens (phrase occurrences) where the average length is only 1.56. Again, phrase tokens provided by TectoMT are significantly longer, 3.68 words on average.

<table>
<thead>
<tr>
<th>phrase tokens</th>
<th>count</th>
<th>avg. len.</th>
<th>CH0</th>
<th>both</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>phrase types</td>
<td>count</td>
<td>avg. len.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Phrase counts and average phrase pairs divided by their source.
data and had two annotators evaluate them. The basic annotation instruction was: “Mark a phrase pair as correct if you can imagine at least some context where it could provide a valid translation.” In other words, we are checking if a phrase pair introduces an error already on its own.

<table>
<thead>
<tr>
<th>table</th>
<th>OK</th>
<th>Bad</th>
<th>Unsure</th>
<th>IAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH</td>
<td>90%</td>
<td>17.5%</td>
<td>6.5%</td>
<td>87.0</td>
</tr>
<tr>
<td>CUTECTOMT</td>
<td>86.3%</td>
<td>26.3%</td>
<td>7.4%</td>
<td>83.0</td>
</tr>
<tr>
<td>used</td>
<td>87.5%</td>
<td>9.0%</td>
<td>3.5%</td>
<td>84.0</td>
</tr>
<tr>
<td>CUTECTOMT</td>
<td></td>
<td></td>
<td></td>
<td>87.0</td>
</tr>
</tbody>
</table>

Table 7: Correctness of phrases in CHIMERA’s phrase tables.

Table 7 shows the results of the annotation. As expected, the percentage of inadmissible phrase pairs is much higher in the first setting (random samples from phrase tables), 17.5–26.3% compared to 7.5–9.0%. Most phrase pairs which contributed to the final translations were valid translations (87.5–89.0%).

The phrase table extracted from TectoMT translations was worse in both settings. However, while only 66% of its phrase pairs were considered correct in the random selection, it was about 87% of phrases actually used. This shows that the final decoder is able to pick the correct suggestions quite successfully.

Interestingly, despite the rather vague task description, inter-annotator agreement was quite high: 80.5% on average in the first setting and 90.5% in the second one.

### 3.3 Automatic Analysis of Errors in Morphology

We were interested to see whether we can find a pattern in the types of morphological errors fixed by adding the TectoMT phrase table. We translated the WMT14 test set using CH1, CH2 and CUTECTOMT. We aligned each translation to the reference using HMM monolingual aligner (Zeman et al., 2011) on lemmas. We focused on cases where both the translation and the reference contain the same (aligned) lemma but the surface forms differ. Table 8 shows summary statistics along with the distribution of errors among Czech parts of speech. We omitted prepositions, adverbs, conjunctions and punctuation from the table – these POSes do not really inflect in Czech.

The number of successfully matched lemmas

(in the HMM alignment phase) is lowest for CH1 – this is expected as this system also got a lower BLEU score. Both other systems matched roughly 400 more lemmas within the test set (this also means 400 more opportunities for making morphological errors, i.e. CH1 and CH2 have a more difficult position than CH0 in this evaluation).

The good news is that CH1 and CH2 show a significantly lower number of errors in morphology – the total number of errors was reduced by almost 500 from the 6065 made by CH0.

Overall, the number of errors per part of speech (POS) is naturally affected by the frequency of the individual POS in Czech text. We see that CH1 (and CH2) reduce the number of errors across all POSes. However, the most prominent improvement can be observed with nouns (N) and adjectives (A). We can roughly say that they account for 407 errors out of the 491 fixed by CH1.

When we look at the morphological tags for each of the 407 errors, we find that the vast majority (393 errors) only differ in morphological case. TectoMT therefore seems to improve target-side morphological coherence and in particular valency and noun-adjective agreement. This is further supported by the manual analysis in Section 3.4.

This analysis does not provide a good picture of the effect of adding Depfix. The difference in error numbers is negligible and inconsistent across POSes (adjectives seemingly got mildly worse while nouns were somewhat improved). Depfix rules generally prefer precision over recall, so they do not change the output considerably. Moreover, valid corrections may not be confirmed by the single reference that we have available. The accuracy of the individual Depfix rules was already evaluated by Bojar et al. (2013b). Depfix significantly improves translation quality according to human evaluation, as evidenced by Table 1.

### 3.4 Manual Analysis of TectoMT n-Grams

In order to check what phenomena are improved by TectoMT, we manually analyzed a small sample of n-grams needed by the reference and provided specifically by TectoMT, i.e. n-grams produced CUTECTOMT but not CH0 and surviving to the final CH1 output. These come from the 1.5% ✓ ✓ 4-grams from Table 5.

The results are presented in Table 9. For each of the examined 4-grams, the annotator started by checking the corresponding part of CH0 output. In
### Table 8: Morphological errors made by Chimera divided by part of speech.

<table>
<thead>
<tr>
<th>System</th>
<th># lemmas</th>
<th># errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH_0</td>
<td>39255</td>
<td>6065</td>
</tr>
<tr>
<td>CH_1</td>
<td>39684</td>
<td>5574</td>
</tr>
<tr>
<td>CH_2</td>
<td>39610</td>
<td>5559</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># lemmas by part of speech</th>
<th>A</th>
<th>C</th>
<th>N</th>
<th>P</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH_0</td>
<td>1200</td>
<td>90</td>
<td>2727</td>
<td>502</td>
<td>1358</td>
</tr>
<tr>
<td>CH_1</td>
<td>1066</td>
<td>75</td>
<td>2354</td>
<td>480</td>
<td>1307</td>
</tr>
</tbody>
</table>

Table 9: Small manual analysis of 4-grams confirmed by the reference and coming from **CU-TECTOMT** (not produced by CH_0, only by CH_1).

- 31.1% of cases, the CH_0 output was an equally acceptable translation. (Other parts of the sentence were not considered.) The false positive 4-grams are fortunately rather rare: 3% of these 4-grams by CH_1 and confirmed by the reference are actually worse than the proposal by CH_0 (“Worsened”) and 1.5% other cases are bad in both CH_1 and CH_0 output (“Bad Anyway”).

#### 4 Complementary Utility

This section contains some observations on how the individual components of Chimera complement each other and to what extent one can substitute another. Unlike the previous section, we are not interested in why the components help but instead in what happens when they are not available.

#### 4.1 Reachability of TectoMT Outputs for Plain Moses

In order to determine whether Moses itself could have produced the translations acquired by combining it with TectoMT, we ran a forced (constrained) decoding experiment (with table limit set to 100) – we ran CH_0 on the WMT14 test set and targeted the translations produced by CH_1. We first put aside the 338 sentences where the outputs of both systems are identical.

<table>
<thead>
<tr>
<th>all</th>
<th>different?</th>
<th>reachable?</th>
<th>score diff</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2665</td>
<td>1741</td>
<td>140 (&gt;</td>
</tr>
<tr>
<td></td>
<td>(unreachable)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>924</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(identical)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Forced decoding – an attempt of CH_0 to reach the test set translations produced by CH_1.
4.2 Long or Morphological LMs vs. TectoMT

In order to learn more about the interplay between the TectoMT phrase table and our language models (LMs), we carried out an experiment where we evaluated all (sensible) subsets of the LMs. For each subset, we reran tuning (MERT) and evaluated the system using BLEU.

As shown above, a significant part of the contribution of TectoMT lies in improving morphological coherence. Since the strong LMs (especially the ones trained on morphological tags) should have a similar effect, we were interested to see whether they complement each other or whether they are mutually replaceable.

In Table 12, we provide results obtained on the WMT14 test set, sorted in ascending order by the BLEU score with TectoMT included. It is immediately apparent that LMs cannot replace the contribution of TectoMT – the best result in the first column (22.69) is noticeably worse than the weakest result obtained with TectoMT included (22.93).

<table>
<thead>
<tr>
<th>LMs</th>
<th>-TectoMT</th>
<th>+TectoMT</th>
<th>∆</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>21.32</td>
<td>22.93</td>
<td>+1.61</td>
</tr>
<tr>
<td>big</td>
<td>22.00</td>
<td>23.19</td>
<td>+1.19</td>
</tr>
<tr>
<td>long longm</td>
<td>22.14</td>
<td>23.31</td>
<td>+1.17</td>
</tr>
<tr>
<td>long morph</td>
<td>22.01</td>
<td>23.48</td>
<td>+1.47</td>
</tr>
<tr>
<td>long morph longm</td>
<td>22.00</td>
<td>23.52</td>
<td>+1.52</td>
</tr>
<tr>
<td>big longm</td>
<td>22.39</td>
<td>23.55</td>
<td>+1.16</td>
</tr>
<tr>
<td>big long</td>
<td>22.26</td>
<td>23.84</td>
<td>+1.58</td>
</tr>
<tr>
<td>big morph</td>
<td>22.21</td>
<td>23.89</td>
<td>+1.68</td>
</tr>
<tr>
<td>big morph longm</td>
<td>22.28</td>
<td>24.01</td>
<td>+1.73</td>
</tr>
<tr>
<td>big long longm</td>
<td>22.69</td>
<td>24.04</td>
<td>+1.35</td>
</tr>
<tr>
<td>big long morph</td>
<td>22.48</td>
<td>24.10</td>
<td>+1.62</td>
</tr>
<tr>
<td>all</td>
<td>22.59</td>
<td>24.24</td>
<td>+1.65</td>
</tr>
</tbody>
</table>

Table 12: Complementary effect of adding TectoMT and language models.

Concerning the usefulness of LMs, it seems that their effects are also complementary – we get the best results by using all of them. It seems that “big” and “long” capture different aspects of the language – “big” provides very reliable statistics on short n-grams while “long” models common long sequences (patterns). The morphological LMs do seem correlated though. When adding “longm”, our aim was to also capture long common patterns in sentential structure. However, it seems that the n-gram order 10 already serves this purpose quite well and extending the range provides only modest improvement.

5 Outstanding Issues

The current combination is quite complex and as such, it results in non-trivial interactions between the components which are hard to identify and describe. We would like to simplify the architecture somehow, striving for a clean, principled design.
However, as we have shown, we cannot simply remove any of the components without a significant loss of translation quality, so this remains an open question for further research.

5.1 Weaknesses of CH

On many occasions, we were surprised by the low quality of CH’s translations. We considered this system a rather strong baseline, given the LMs trained on billions of tokens and the factored scheme, which specifically targets morphological coherence. Yet we observed many obvious errors both in lexical choice and morphological agreement, which were well within the scope of the phrase length limit and $n$-gram order. We believe that more sophisticated statistical models, such as discriminative classifiers which take source context into account (Carpuat and Wu, 2007) or operation sequence models (Durrani et al., 2011), could be applied to further improve CH.

5.2 Practical Considerations

As we have shown, our approach to system combination has some unique properties and can certainly be an interesting alternative. Yet it can be viewed as impractical – the models (the TectoMT phrase table, specifically) actually require the input to be known in advance. In this section, we outline a possible solution which would allow for using the system in an on-line setting.

The synthetic parallel data consist of the dev set and test set. Our development data can be fixed in advance so re-tuning the system parameters is not required for new inputs.

The only remaining issue is ensuring that the second phrase table contains the TectoMT translation of the input. We propose to first translate the input sentence using TectoMT. Then for word alignment, we can either use the alignment information directly from TectoMT or apply a pre-trained word-alignment model, provided e.g. by MGiza (Gao and Vogel, 2008). Phrase extraction and scoring can be done quickly on the fly.

Phrase scores should ideally be combined with the dev-set part of the phrase table. Moses has support for dynamic updating of its phrase tables (Bertoldi, 2014), so changing the scores or adding new phrase pairs is possible at very little cost.

With pre-trained word alignment and dynamic updating of the phrase table, we believe that our approach could be readily deployed in practice.

6 Conclusion

We have carefully analyzed the system combination Chimera which consists of a statistical system Moses (CHo), a deep-syntactic transfer-based system TectoMT and a rule-based post-processing tool Depfix. We focused on the interaction between CHo and CU-TECTOMT. We described several techniques for inspecting this combination, based on both automatic and manual evaluation.

We have found that the transfer-based component provides a mix of useful, correct translations and noise. Many of its translations are unavailable to the statistical component, so its generalization power is in fact essential. Moses is able to select the useful translations quite successfully thanks to strong language models, which are trained both on surface forms and morphological tags.

Our experiment with forced decoding further showed that translations which are reachable for Moses are often not chosen due to modelling errors. It is the extra prominence these translations get thanks to CU-TECTOMT that helps to overcome these errors.

We show that our approach to system combination (using translations from the transfer-based system as additional training data) has several advantageous properties and that it might be an interesting alternative to standard techniques. We outline a solution to the issue of the practical applicability of our method.

Overall, we find that by adding the transfer-based system, we obtain novel translations and improved morphological coherence. The final translation quality is improved significantly over both CHo and CU-TECTOMT alone, setting the state of the art for English→Czech translation for several years in a row.

Acknowledgements

This research was supported by the grants H2020-ICT-2014-1-645452 (QT21), H2020-ICT-2014-1-644402 (HimL), H2020-ICT-2014-1-644753 (KConnect), SVV 260 224 and GAUK 1356213. 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 LM2010013).
References


B CUNI-LMU Submissions in WMT2016: Chimera Constrained and Beaten

Aleš Tamchyna 1,2, Roman Sudarikov 1, Ondřej Bojar 1, Alexander Fraser 2
1 Charles University in Prague, Prague, Czech Republic
2 LMU Munich, Munich, Germany
surname@ufal.mff.cuni.cz fraser@cis.uni-muenchen.de

Abstract

This paper describes the phrase-based systems jointly submitted by CUNI and LMU to English-Czech and English-Romanian News translation tasks of WMT16. In contrast to previous years, we strictly limited our training data to the constraint datasets, to allow for a reliable comparison with other research systems. We experiment with using several additional models in our system, including a feature-rich discriminative model of phrasal translation.

1 Introduction

We have a long-term experience with English-to-Czech machine translation and over the years, our systems have grown together from rather diverse set of system types to a single system combination called Chimera (Bojar et al., 2013). This system has been successful in the previous three years of WMT (Bojar et al., 2013; Tamchyna et al., 2014; Bojar and Tamchyna, 2015) and we follow a similar design this year. Unlike previous years, we only use constrained data in system training, to allow for a more meaningful comparison with the competing systems. The gains thanks to the additional data in contrast to the gains thanks the system combination have been evaluated in terms of BLEU in Bojar and Tamchyna (2015). The details of our English-to-Czech system are in Section 2.

In this work, we also present our system submission for English-Romanian translation. This system uses a factored setting similar to Chimera but lacks its two key components: the deep-syntactic translation system TectoMT and the rule-based post-processing component Depfix. All details are in Section 3.

2 English-Czech System

Our “baseline” setup is fairly complex, following Bojar et al. (2013). The key components of Chimera are:

• Moses, a phrase-based factored system (Koehn et al., 2007).
• TectoMT, a deep-syntactic transfer-based system (Popel and Žabokrtský, 2010).
• Depfix, a rule-based post-processing system (Rosa et al., 2012).

The core of the system is Moses. We combine it with TectoMT in a simple way which we refer to as “poor man’s” system combination: we translate our development and test data with TectoMT first and then add the source sentences and their translations as additional (synthetic) parallel data to the Moses system. This new corpus is used to train a separate phrase table. At test time, we run Moses which uses both phrase tables and we correct its output using Depfix. The system is described in detail in Bojar et al. (2013).

Our subsequent analysis in Tamchyna and Bojar (2015) shows that the contribution of TectoMT is essential for the performance of Chimera. In particular, TectoMT provides new translations which are otherwise not available to the phrase-based system and it also improves the morphological and syntactic coherence of translations.

2.1 Translation Models

Similarly to previous years, we build two phrase tables – one from parallel data and another from TectoMT translations of the development and test sets. Here we describe the first phrase table.

Our main system uses CzEng16pre (Bojar et al., 2016) as parallel data. We train a factored TM
which uses surface forms on the source and produces target form, lemma and tag. Similarly to previous years, we find that increasing the phrase table limit (the maximum number of possible translations per source phrase) is necessary to obtain good performance.

Our input is also factored (though the phrase tables do not condition on these additional factors) and contains the form, lemma and morphological tag. We use these factors to extract rich features for our discriminative context model.

**Linearly interpolated translation models.** There is some evidence that when dealing with heterogeneous domains, it might be beneficial to construct the final TM as a linear, uniform interpolation of many small phrase tables (Carpuat et al., 2014). We experiment with splitting the data into 20 parts (without any domain selection, simply a random shuffle) and using linear interpolation to combine the partial models. The added benefit is that phrase extraction for all these parts can run in parallel (2h25m per part on average). The merging of these parts took 16h12m, which is still substantially faster than the single extraction (53h7m).

### 2.2 Language Models

Our LM configuration is based on the successful setting from previous years, however all LMs are trained using the constrained data; this is a major difference from our previous submissions which used several gigawords of monolingual text for language modeling.

We train an 7-gram LM on surface forms from all monolingual news data available for WMT. This LM is linearly interpolated (each year is a separate model) to optimize perplexity on a held-out set (WMT newstest2012). The individual LMs were pruned: we discarded all singleton n-grams (apart from unigrams).

All other LMs are trained on simple concatenation of the news part of CzEng16pre and all WMT monolingual news sets. We train 4-gram LMs on forms and lemmas (with a different pruning scheme: we discard 2- and 3-grams which appear fewer than 2 or 3 times, respectively).

We have two LMs over morphological tags to help maintain morphological coherence of translation outputs. The first LM is a 10-gram model and the second one is a 15-gram model, aimed at overall sentence structure. We prune all singleton n-grams (again, with the exception of unigrams).

### 2.3 Discriminative Translation Model

We add a feature-rich, discriminative model of phrasal translation to our system (Tamchyna et al., 2016). This classifier produces a single phrase translation probability which is additionally conditioned on the full source sentence and limited left-hand-side target context. The probability is added as an additional feature to Moses’ log-linear model. The motivation for adding the context model is to improve lexical choice (which can be better inferred thanks to full source-context information) and morphological coherence.

The model uses a rich feature set on both sides: In the source, the model has access to the full input sentence and uses surface forms, lemmas and tags. On the target side, the model has access to limited context (similarly to an LM) and uses target surface forms, lemmas and tags. However, our English-Czech submission to WMT16 does not use target-context information due to time constraints.

### 2.4 Lexicalized Reordering and OSM

We experiment with using a lexicalized reordering model (Koehn et al., 2005) in the common setting: model monotone/swap/discontinuous reordering, word-based extraction, bidirectional, conditioned both on the source and target language.

We also train an operation sequence model (OSM, Durrani et al., 2013), which is a generative model that sees the translation process as a linear sequence of operations which generate a source and target sentence in parallel. The probability of a sequence of operations is defined according to an n-gram model, that is, the probability of an operation depends on the n − 1 preceding operations. We have trained our 5-gram model on surface forms, using the CzEng16pre corpus.

### 2.5 Hard POS for Short Words

In addition to the more principled attempts at improving our model, mainly Section 2.3, we also manually checked the output and added an ad-hoc solution for the single most disturbing error: the abbreviated form “’s” was often translated as the verb “to be” even in the clearly possessive uses.

The ambiguity of “’s” is apparently easy to resolve, our tagger does not have problems distinguishing and tagging the abbreviation as POS (possessive), VBZ (present tense) and other situations. While the POS information is readily avail-
able to the discriminative model, the model might not be able to pick it up due to its wide focus on many phenomena. As an alternative, we simply modify the input token and append the POS tag to it for all tokens under three characters. This hack clearly helps with “'s”: in a small manual analysis of 52 occurrences of “'s”, the discriminative model still translated 7 possessive meanings as present tense, while the hacked model avoided these errors. It would be best to combine these two approaches, but we did not have the time to run this setting for the WMT evaluation.

2.6 Results
We evaluate all system variants on the WMT15 test set and report all BLEU scores in Table 1 prior to applying the last component, Depfix.

The reordering model achieved mixed results in our initial experiments and we opt not to include it in our final submission, relying instead only on the standard distortion penalty feature.

As in previous years, the addition of TectoMT to the main phrase table extracted from the parallel corpus (denoted “CzEng” in Table 1) is highly beneficial, improving the BLEU score by roughly 1.2 points. The addition of OSM also helps, adding about 0.7 points.

The source-context discriminative model does not improve translation quality according to BLEU. We suspect that the space for its contribution is diminished by the addition of TectoMT and possibly also the OSM and the strong LMs. This system (labelled with *) was submitted as a primary system CU-TAMCHYNA. After the deadline, we also ran an experiment which included target-context features in the model and obtained BLEU of 20.96.

Experiments with the interpolated TM (“CzEng+Tm,” in the table) and POS appended to words under three characters show a lower BLEU score (20.70, denoted •) but we also carried out a small manual evaluation where the system output seemed to be better than the baseline (20.91). We therefore submitted this system as our primary CU-CHIMERA.

In the official WMT16 manual evaluation, both our systems end up in the same cluster, ranking #4 and #5 among all systems for this language pair. The hacked system • seems negligibly better (0.302 TrueSkill) than the one with the discriminative model (+, reaching 0.299 TrueSkill).

As a contrastive result, CHIMERA, ranking #1 last year, achieves a BLEU score of 20.46 on newstest2015 (also prior to the application of Depfix). This suggests that even though we limited our training data this year, we did not lose anything in terms of translation quality.

<table>
<thead>
<tr>
<th>TMs</th>
<th>OSM</th>
<th>Disc</th>
<th>POS</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CzEng</td>
<td></td>
<td>-</td>
<td>-</td>
<td>19.08±0.62</td>
</tr>
<tr>
<td>CzEng+TectoMT</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>20.70±1.68</td>
</tr>
<tr>
<td>CzEng+OSM+TectoMT</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>20.91±0.67</td>
</tr>
<tr>
<td>CzEng+OSM+TectoMT*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>20.89±0.69</td>
</tr>
</tbody>
</table>

Table 1: Different experiment configurations for CHIMERA. We report BLEU scores on newstest2015. The system denoted • corresponds to our WMT16 submission cu-tamchyna and the system denoted * corresponds to cu-chimera.

3 English-Romanian System
We also submitted a constrained phrase-based system for English→Romanian translation which is loosely inspired by the basic components of CHIMERA. Additionally, our submission uses the source- and target-context discriminative translation model as well.

3.1 Data and Pre-Processing
We use all the data available to constrained submissions: Europarl v8 (Koehn, 2005) and SETIMES2 (Tiedemann, 2009) parallel corpora and News 2015 and Common Crawl monolingual corpora.1 We split the official development set into two halves; we use the first part for system tuning and the second part serves as our test set.

Data pre-processing differs between English and Romanian. For English, we use Treex (Popel and Žabokrtský, 2010) to obtain morphological tags, lemmas and dependency parses of the sentences. For Romanian, we use the online tagger by Tufis et al. (2008) as run by our colleagues at LIMSI-CNRS for the joint QT21 Romanian system (Peter et al., 2016).

3.2 Factored Translation
Similarly to CHIMERA, we train a factored phrase table which translates source surface forms to tuples (form, lemma, tag). Our input is factored and contains the form, lemma, morphological tag.

1http://commoncrawl.org/
3.3 Language Models

Our full system contains three separate language models (LMs). The first is a 5-gram LM over surface forms, trained on the target side of the parallel data and monolingual news 2015.

The second LM only uses 4-grams but additionally contains the full Common Crawl corpus. We prune this second LM by discarding 2-, 3- and 4-grams which appear fewer than 2, 3, 4 times, respectively.

Finally, we also include a 7-gram LM over morphological tags. We only use target parallel data for estimating the model.

3.4 Reordering Model

Similarly to our experiments with CHIMERA, we utilize a lexicalized reordering model (Koehn et al., 2005). Again, we model monotone/swap/discontinuous reordering, word-based extraction, bidirectional, conditioned both on the source and target language.

3.5 Discriminative Translation Model

We utilize the same discriminative model as for CHIMERA. For English-Romanian, we also use dependency parses of the source sentences and target-side context features as additional source of information in our official submission.

3.6 Results

Table 2 lists BLEU scores of various system settings. Each BLEU score is an average over 5 runs of system tuning (MERT, Och, 2003). The table shows how BLEU score develops as we add the individual components to the system: the 7-gram morphological LM ("tagLM"), the 4-gram LM from Common Crawl ("ccrawl"), the lexicalized reordering ("RR") and finally the discriminative translation model ("discTM").

We test for statistical significance using MultiEval (Clark et al., 2011): we test each new component against the system without it (i.e., +tagLM is compared to baseline, +ccrawl is tested against +tagLM etc.). When the p-value is lower than 0.05, we mark the result in bold.

<table>
<thead>
<tr>
<th>Setting</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>26.2</td>
</tr>
<tr>
<td>+tagLM</td>
<td>26.6</td>
</tr>
<tr>
<td>+ccrawl</td>
<td>28.0</td>
</tr>
<tr>
<td>+RM</td>
<td>28.1</td>
</tr>
<tr>
<td>+discTM</td>
<td>28.3</td>
</tr>
</tbody>
</table>

Table 2: BLEU scores of system variants for English-Romanian translation.

We observe a relatively steady additive effect of the individual components: the addition of each model (apart from lexicalized reordering) leads to a statistically significant improvement in translation quality.

Our discriminative model further improves the system, despite only being trained on the parallel data (roughly 0.6 M sentence pairs) and building upon the strong language models which use orders-of-magnitude larger monolingual data (almost 300 M sentences). This variant (BLEU 28.3) corresponds to our submission LMU-CUNI.

4 Conclusion

We have described our English-Czech and English-Romanian submissions to WMT16: CU-CHIMERA, CU-TAMCHYNA and LMU-CUNI.

For English-Czech, our work is an incremental improvement of the previously successful CHIMERA system. This time, our submission is constrained and additionally uses interpolated TMs, an OSM and a discriminative phrasal translation model.

For English-Romanian, we have built a system somewhat similar to the statistical component of CHIMERA. We have added the discriminative model which conditions both on the source and target context to the system and obtained a small but significant improvement in BLEU.

5 Acknowledgement

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements no. 644402 (HimL) and no. 645452 (QT21). This work has been using language resources stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071). This work was partially supported by SVV project number 260 333.
References


tional Linguistics, Montréal, Canada, pages 362–368.


Graph-Based Translation Via Graph Segmentation

Liangyou Li and Andy Way and Qun Liu
ADAPT Centre, School of Computing
Dublin City University, Ireland
{liangyouli,away,qliu}@computing.dcu.ie

Abstract

One major drawback of phrase-based translation is that it segments an input sentence into continuous phrases. To support linguistically informed source discontinuity, in this paper we construct graphs which combine bigram and dependency relations and propose a graph-based translation model. The model segments an input graph into connected subgraphs, each of which may cover a discontinuous phrase. We use beam search to combine translations of each subgraph left-to-right to produce a complete translation. Experiments on Chinese–English and German–English tasks show that our system is significantly better than the phrase-based model by up to +1.5/+0.5 BLEU scores. By explicitly modeling the graph segmentation, our system obtains further improvement, especially on German–English.

1 Introduction

Statistical machine translation (SMT) starts from sequence-based models. The well-known phrase-based (PB) translation model (Koehn et al., 2003) has significantly advanced the progress of SMT by extending translation units from single words to phrases. By using phrases, PB models can capture local phenomena, such as word order, word deletion, and word insertion. However, one of the significant weaknesses in conventional PB models is that only continuous phrases are used, so generalizations such as French ne . . . pas to English not cannot be learned. To solve this, syntax-based models (Galley et al., 2004; Chiang, 2005; Liu et al., 2006; Marcu et al., 2006) take tree structures into consideration to learn translation patterns by using non-terminals for generalization. However, the expressiveness of these models is confined by hierarchical constraints of the grammars used (Galley and Manning, 2010) since these patterns still cover continuous spans of an input sentence.

By contrast, Quirk et al. (2005), Menezes and Quirk (2005) and Xiong et al. (2007) take treelets from dependency trees as the basic translation units. These treelets are connected and may cover discontinuous phrases. However, their models lack the ability to handle continuous phrases which are not connected in trees but could in fact be extremely important to system performance (Koehn et al., 2003). Galley and Manning (2010) directly extract discontinuous phrases from input sequences. However, without imposing additional restrictions on discontinuity, the amount of extracted rules can be very large and unreliable.

Different from previous work (as shown in Table 1), in this paper we use graphs as input structures and propose a graph-based translation model to translate a graph into a target string. The basic translation unit in this model is a connected subgraph which may cover discontinuous phrases. The main contributions of this work are summarized as follows:

- We propose to use a graph structure to combine a sequence and a tree (Section 3.1). The
graph contains both local relations between words from the sequence and long-distance relations from the tree.

- We present a translation model to translate a graph (Section 3). The model segments the graph into subgraphs and uses beam search to generate a complete translation from left to right by combining translation options of each subgraph.

- We present a set of sparse features to explicitly model the graph segmentation (Section 4). These features are based on edges in the input graph, each of which is either inside a subgraph or connects the subgraph with a previous subgraph.

- Experiments (Section 5) on Chinese–English and German–English tasks show that our model is significantly better than the PB model. After incorporating the segmentation model, our system achieves still further improvement.

2 Review: Phrase-based Translation

We first review the basic PB translation approach, which will be extended to our graph-based translation model. Given a pair of sentences \((S, T)\), the conventional PB model is defined as Equation (1):

\[
p(T | S) = \prod_{i=1}^{f} p(T_i | \Sigma_{a_i}) d(\Sigma_{a_i}, \Sigma_{a_{i-1}})
\]  

(1)

The target sentence \(T\) is broken into \(f\) phrases \(T_1 \cdots T_f\), each of which is a translation of a source phrase \(\Sigma_{a_i}\). \(d\) is a distance-based reordering model. Note that in the basic PB model, the phrase segmentation is not explicitly modeled which means that different segmentations are treated equally (Koehn, 2010).

The performance of PB translation relies on the quality of phrase pairs in a translation table. Conventionally, a phrase pair \(\langle \pi, \tilde{\pi} \rangle\) has two properties: (i) \(\pi\) and \(\tilde{\pi}\) are continuous phrases. (ii) \(\langle \pi, \tilde{\pi} \rangle\) is consistent with a word alignment \(A\) (Och and Ney, 2004): \(\forall (i, j) \in A, \Sigma_i \in \pi \Leftrightarrow t_j \in \tilde{\pi}\) and \(\exists \Sigma_i \in \pi, t_j \in \tilde{\pi}, (i, j) \in A\).

PB decoders generate hypotheses (partial translations) from left to right. Each hypothesis maintains a coverage vector to indicate which source words have been translated so far. A hypothesis can be extended on the right by translating an uncovered source phrase. The translation process ends when all source words have been translated.

Beam search (as in Figure 1) is taken as an approximate search strategy to reduce the size of the decoding space. Hypotheses which cover the same number of source words are grouped in a stack. Hypotheses can be pruned according to their partial translation cost and an estimated future cost.

3 Graph-Based Translation

Our graph-based translation model extends PB translation by translating an input graph rather than a sequence to a target string. The graph is segmented into a sequence of connected subgraphs, each of which corresponds to a target phrase, as in Equation (2):

\[
p(T_1 | G(\hat{s}_1)) = \prod_{i=1}^{f} p(T_i | G(\hat{s}_i)) d(G(\hat{s}_i), G(\hat{s}_{i-1}))
\]  

(2)

where \(G(\hat{s}_i)\) denotes a connected source subgraph which covers a (discontinuous) phrase \(s_i\).

3.1 Building Graphs

As a more powerful and natural structure for sentence modeling, a graph can model various kinds of word-relations together in a unified representation. In this paper, we use graphs to combine two commonly used relations: bigram relations and dependency relations. Figure 2 shows an example of a graph. Each edge in the graph denotes either a dependency relation or a bigram relation. Note that the graph we use in this paper is directed, connected, node-labeled and may contain cycles.

Bigram relations are implied in sequences and provide local and sequential information on pairs
D1.1: Semantic Translation Models

3.2 Training

Different from PB translation, the basic translation units in our model are subgraphs. Thus, during training, we extract subgraph–phrase pairs instead of phrase pairs on parallel graph–string sentences associated with word alignments.

An example of a translation rule is as follows:

Note that the source side of a rule in our model is a graph which can be used to cover either a continuous phrase or a discontinuous phrase according to its match in an input graph during decoding.

The algorithm for extracting translation rules is shown in Algorithm 1. This algorithm traverses each phrase pair (s, t), which is within a length limit and consistent with a given word alignment

Algorithm 1: Algorithm for extracting translation rules from a graph-string pair.

Data: A word-aligned graph–string pair 

Result: A set of translation pairs R

1. for each phrase \( \tilde{t} \) in \( T \); \( |\tilde{t}| \leq L \) do
2. find the minimal (may be discontinuous) phrase \( \hat{s} \) in \( S \) so that \( |\hat{s}| \leq L \) and \((\hat{s}, \tilde{t})\) is consistent with \( A \);
3. Queue \( Q = \{\hat{s}\} \);
4. while \( Q \) is not empty do
5. pop an element \( \hat{s} \) off;
6. if \( G(\hat{s}) \) is connected then
7. add \( (G(\hat{s}), \tilde{t}) \) to \( R \);
8. end
9. if \( |\hat{s}| < L \) then
10. for each unaligned word \( s_i \) adjacent to \( \hat{s} \) do
11. \( \hat{s}' = \) extend \( \hat{s} \) with \( s_i \);
12. add \( \hat{s}' \) to \( Q \);
13. end
14. end
15. end
16. end

(lines 1–2), and outputs \((G(\hat{s}), \tilde{t})\) if \( \hat{s} \) is covered by a connected subgraph \( G(\hat{s}) \) (lines 6–8). A source phrase can be extended with unaligned source words which are adjacent to the phrase (lines 9–14). We use a queue \( Q \) to store all phrases which are consistently aligned to the same target phrase (line 3).

3.3 Model and Decoding

We define our model in the log-linear framework (Och and Ney, 2002) over a derivation \( D = r_1 \cdot r_2 \cdot \ldots \cdot r_N \), as in Equation (3):

\[
p(D) \propto \prod_i \phi_i(D)^{\lambda_i}
\]

where \( r_i \) are translation rules, \( \phi_i \) are features defined on derivations and \( \lambda_i \) are feature weights. In our experiments, we use the standard 9 features: two translation probabilities \( p(G(s)|t) \) and \( p(t|G(s)) \), two lexical translation probabilities \( p_{lex}(s|t) \) and \( p_{lex}(t|s) \), a language model \( lm(t) \) over a translation \( t \), a rule penalty, a word penalty, an unknown word penalty and a distortion feature \( d \) for distance-based reordering.

The calculation of the distortion feature \( d \) in our
model is different from the one used in conventional PB models, as we need to take discontinuity into consideration. In this paper, we use a distortion function defined in Galley and Manning (2010) to penalize discontinuous phrases that have relatively long gaps. Figure 3 shows an example of calculating distortion for discontinuous phrases.

Our graph-based decoder is very similar to the PB decoder except that, in our decoder, each hypothesis is extended by translating an uncovered subgraph instead of a phrase. Positions covered by the subgraph are then marked as translated.

### 4 Graph Segmentation Model

Each derivation in our graph-based translation model implies a sequence of subgraphs (also called a segmentation). By default, similar to PB translation, our model treats each segmentation equally as shown in Equation (2). However, previous work on PB translation has suggested that such segmentations provide useful information which can improve translation performance. For example, boundary information in a phrase segmentation can be used for reordering models (Xiong et al., 2006; Cherry, 2013).

In this paper, we are interested in directly modeling the segmentation using information from graphs. By making the assumption that each subgraph is only dependent on previous subgraphs, we define a generative process over a graph segmentation as in Equation (4):

$$p(G(s_1)\ldots G(s_t)) = \prod_{i=1}^t P(G(s_i)|G(s_1)\ldots G(s_{i-1}))$$  \hspace{1cm} (4)

Instead of training a stand-alone discriminative segmentation model to assign each subgraph a probability given previous subgraphs, we implement the model via sparse features, each of which is extracted at run-time during decoding and then directly added to the log-linear framework, so that these features can be tuned jointly with other features (of Section 3.3) to directly maximize the translation quality.

Since a segmentation is obtained by breaking up the connectivity of an input graph, it is intuitive to use edges to model the segmentation. According to Equation (4), for a current subgraph $G_i$, we only consider those edges which are either inside $G_i$ or connect $G_i$ with a previous subgraph. Based on these edges, we extract sparse features for each node in the subgraph. The set of sparse features is defined as follows:

$$\{n.w, n.c\} \times \{n'.w, n'.c\} \times \begin{cases} C \times P \times \{in\} \times \{out\} \end{cases}$$

where $n.w$ and $n.c$ are the word and class of the current node $n$, and $n'.w$ and $n'.c$ are the word and class of a node $n'$ connected to $n$. $C$, $P$, and $H$ denote that the node $n'$ is in the current subgraph $G_i$ or the adjacent previous subgraph $G_{i-1}$ or other previous subgraphs, respectively. Note that we treat the adjacent previous subgraph differently from others since information from the last previous unit is quite useful (Xiong et al., 2006; Cherry, 2013). $in$ and $out$ denote that the edge is an incoming edge or outgoing edge for the current node $n$. Figure 4 shows an example of extracting sparse features for a subgraph.

Inspired by success in using sparse features in SMT (Cherry, 2013), in this paper we lexicalize only on the top-100 most frequent words. In addition, we group source words into 50 classes by $\text{mkcls}$ which should provide useful generalization (Cherry, 2013) for our model.

### 5 Experiment

We conduct experiments on Chinese–English (ZH–EN) and German–English (DE–EN) translation tasks. Table 2 provides a summary of our corpora. Our ZH–EN training corpus contains 1.5M+ sentences from LDC. NIST 2002 (MT02) is taken as a development set to tune weights, and NIST
2010 (MT04) and NIST 2005 (MT05) are two test sets used to evaluate the systems. The Stanford Chinese word segmenter (Chang et al., 2008) is used to segment Chinese sentences. The Stanford dependency parser (Chang et al., 2009) parses a Chinese sentence into a projective dependency tree which is then converted to a graph by adding bigram relations.

The DE–EN training corpus is from WMT 2014, including Europarl V7 and News Commentary. News-Test 2011 (WMT11) is taken as a development set while News-Test 2012 (WMT12) and News-Test 2013 (WMT13) are test sets. We use mate-tools\(^2\) to perform morphological analysis and parse German sentences (Bohnet, 2010). Then, MaltParser\(^3\) converts a parse result into a projective dependency tree (Nivre and Nilsson, 2005).

5.1 Settings

In this paper, we mainly report results from five systems under the same configuration. PBMT is built by the PB model in Moses (Koehn et al., 2007). Treelet extends PBMT by taking treelets as the basic translation units (Quirk et al., 2005; Menezes and Quirk, 2005). We implement a Treelet model in Moses which produces translations from left to right and uses beam search for decoding. DTU extends the PB model by allowing discontinuous phrases (Galley and Manning, 2010). We implement DTU with source discontinuity in Moses.\(^4\) GBMT is our basic graph-based translation system while GSM adds the graph segmentation model into GBMT. Both systems are implemented in Moses.

Word alignment is performed by GIZA++ (Och and Ney, 2003) with the heuristic function \textit{grow-diag-final-and}. We use SRILM (Stolcke, 2002) to train a 5-gram language model on the Xinhua portion of the English Gigaword corpus 5th edition with modified Kneser-Ney discounting (Chen and Goodman, 1996). Batch MIRA (Cherry and Foster, 2012) is used to tune weights. BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2011), and TER (Snover et al., 2006) are used for evaluation.

\(^2\)The re-implementation of DTU in Moses makes it easier to meaningfully compare systems under the same settings.
5.2 Results and Discussion

Table 3 shows our evaluation results. We find that our GBMT system is significantly better than PBMT as measured by all three metrics across all test sets. Specifically, the improvements are up to +1.5/+0.5 BLEU, +0.3/+0.2 METEOR, and -0.8/-0.4 TER on ZH–EN and DE–EN, respectively. This improvement is reasonable as our system allows discontinuous phrases which can reduce data sparsity and handle long-distance relations (Galley and Manning, 2010). Another argument for discontinuous phrases is that they allow the decoder to use larger translation units which tend to produce better translations (Galley and Manning, 2010). However, this argument was only verified on ZH–EN. Therefore, we are interested in seeing whether we have the same observation in our experiments on both language pairs.

We count the used translation rules in MT02 and WMT11 based on different target lengths. The results are shown in Figure 5. We find that both DTU and GBMT indeed tend to use larger translation units on ZH–EN. However, more smaller translation units are used on DE–EN.\(^5\) We presume this is because long-distance reordering is performed more often on ZH–EN than on DE–EN. Based on the fact that the distortion function \(d\) measures the reordering distance, we find that the average distortion value in PB on ZH–EN MT02 is 18.4 and 3.5 on DE–EN WMT11. Our observations suggest that the argument that discontinuous phrases allow decoders to use larger translation units should be considered with caution when we explain the benefit of discontinuity on different language pairs.

Compared to PBMT, the Treelet system does not show consistent improvements. Our system achieves significantly better BLEU and METEOR scores than Treelet on both ZH–EN and DE–EN, and a better TER score on DE–EN. This suggests that continuous phrases are essential for system robustness since it helps to improve phrase coverage (Hanneman and Lavie, 2009). Lower phrase coverage in Treelet results in more short phrases being used, as shown in Figure 5. In addition, we find that both DTU and our systems do not achieve consistent improvements over Treelet in terms of TER. We observed that both DTU and our systems tend to produce longer translations than Treelet, which might cause unreliable TER evaluation in our experiments as TER favours shorter sentences (He and Way, 2010).

Since discontinuous phrases produced by using syntactic information are fewer in number but

<table>
<thead>
<tr>
<th>Metric</th>
<th>System</th>
<th>ZH–EN MT04</th>
<th>DE–EN WMT12</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU ↑</td>
<td>PBMT 33.2</td>
<td>31.8</td>
<td>19.5</td>
</tr>
<tr>
<td>Treelet 33.8†</td>
<td>31.7</td>
<td>19.6</td>
<td>22.1†</td>
</tr>
<tr>
<td>GBMT 34.7†</td>
<td>32.3†</td>
<td>19.8†</td>
<td>22.4†</td>
</tr>
<tr>
<td>GSM 34.9†</td>
<td>32.7†</td>
<td>20.3†</td>
<td>22.9†</td>
</tr>
</tbody>
</table>

| METEOR ↑ | PBMT 32.1 | 32.3 | 28.0 | 29.2 |
| Treelet 32.9 | 31.8 | 28.0 | 29.1 |
| DTU 32.3* | 32.4 | 28.2* | 29.5* |
| GBMT 32.4* | 32.5* | 28.2* | 29.4* |
| GSM 32.7* | 32.6* | 28.5* | 29.8* |

| TER ↓ | PBMT 60.6 | 61.6 | 63.7 | 60.2 |
| Treelet 60.1* | 61.4 | 63.2* | 59.6* |
| DTU 60.0 | 61.5 | 63.5* | 59.8* |
| GBMT 59.8* | 61.3* | 63.5* | 59.8* |
| GSM 60.5 | 62.1 | 63.1* | 59.3* |

\(^5\) We have the same finding on all test sets.

Table 3: Metric scores for all systems on Chinese–English (ZH–EN) and German–English (DE–EN). Each score is an average over three MIRA runs (Clark et al., 2011). * means a system is significantly better than PBMT at \(p \leq 0.01\). Bold figures mean a system is significantly better than Treelet at \(p \leq 0.01\). + means a system is significantly better than DTU at \(p \leq 0.01\). In this table, we mark a system by comparing it with previous ones.

Table 4: The number of rules in DTU and GBMT.
more reliable (Koehn et al., 2003), our GBMT system achieves comparable performance with DTU but uses significantly fewer rules, as shown in Table 4. After integrating the graph segmentation model to help subgraph selection, GBMT is further improved and the resulted system G2S has significantly better evaluation scores than DTU on both language pairs. However, our segmentation model is more helpful on DE–EN than ZH–EN. We find that the number of features learned on ZH–EN (25K+) is much less than on DE–EN (49K+). This may result in a lower feature coverage during decoding. The lower number of features in ZH–EN could be caused by the fact that the development set MT02 has many fewer sentences than WMT11. Accordingly, we suggest to use a larger development set during tuning to achieve better translation performance when the segmentation model is integrated.

Our current model is more akin to addressing problems in phrase-based and treelet-based models by segmenting graphs into pieces rather than extracting a recursive grammar. Therefore, similar to those models, our model is weak at phrase reordering as well. However, we are interesting in the potential power of our model by incorporating lexical reordering (LR) models and comparing it with syntax-based models.

Table 5 shows BLEU scores of the hierarchi-cal phrase-based (HPB) system (Chiang, 2005) in Moses and GBMT combined with a word-based LR model (Koehn et al., 2005). We find that the LR model significantly improves our system. GBMT+LR is comparable with the Moses HPB model on Chinese–English and better than HPB on German–English.

5.3 Examples

Figure 6 shows three examples from MT04 to better explain the differences of each system. Example 1 shows that systems which allow discontinuous phrases (namely Treelet, DTU, GBMT, and GSM) successfully translate a Chinese collocation “Yu . . . Wuguan” to “have nothing to do with” while PBMT fails to catch the generalization since it only allows continuous phrases.

In Example 2, Treelet translates a discontinuous phrase “Dui . . . Zuofa” (to . . . practice) only as “to” where an important target word “practice” is dropped. By contrast, bigram relations allow our systems (GBMT and GSM) to find a better phrase to translate: “De Zuofa” to “of practice”. In addition, DTU translates a discontinuous phrase “De Zuofa . . . Buman” to “dissatisfaction with the approach of”. However, the phrase is actually not
D1.1: Semantic Translation Models

Example 1

**PBMT**: the united states has indicated that the united states and north korea delegation has visited north korea.

**Treelet**: the united states has indicated that it has nothing to do with the united states delegation visited north korea.

**DTU**: the united states has indicated that it has nothing to do with the united states delegation visited north korea.

**GBMT**: the united states has indicated that it has nothing to do with the united states delegation visited north korea.

**GSM**: the united states has indicated that it has nothing to do with the united states delegation visited north korea.

**REF**: the american government said that it has nothing to do with the american delegation to visit north korea.

---

Example 2

**PBMT**: the united states government to brazil has repeatedly expressed its dissatisfaction.

**Treelet**: the government of brazil has on many occasions expressed their discontent.

**DTU**: the united states has repeatedly expressed its dissatisfaction with the approach of the government to brazil.

**GBMT**: the us government has repeatedly expressed dissatisfaction with the practice of brazil.

**GSM**: the us government has repeatedly expressed dissatisfaction with the practice of brazil.

**REF**: the us government has expressed their resentment against this practice of brazil on many occasions.

---

Example 3

**PBMT**: the government and all sectors of society should continue to explore in depth and draw on collective wisdom.

**Treelet**: the government must continue to make in-depth discussions with various sectors of the community and the collective wisdom.

**DTU**: the government must continue to work together with various sectors of the community to make an in-depth study and draw on collective wisdom.

**GBMT**: the government must continue to work together with various sectors of the community to make an in-depth study and draw on collective wisdom.

**GSM**: the government must continue to work together with various sectors of the community and draw on collective wisdom.

**REF**: the government must continue to hold thorough discussions with all walks of life to pool the wisdom of the masses.

---

Figure 6: Translation examples from MT04 produced by different systems. Each source sentence is annotated by dependency relations and additional bigram relations (dotted red edges). We also annotate phrase alignments produced by our system GSM.
linguistically motivated and could be unreliable. By disallowing phrases which are not connected in the input graph, GBMT and GSM produce better translations.

Example 3 illustrates that our graph segmentation model helps to select better subgraphs. After obtaining a partial translation “the government must”, GSM chooses to translate a subgraph which covers a discontinuous phrase “Jixu … Zuo” to “continue to make” while GBMT translates “Jixu Yu” (continue … with) to “continue to work together with”. By selecting the proper subgraph to translate, GSM performs a better reordering on the translation.

6 Related Work

Starting from sequence-based models, SMT has been benefiting increasingly from complex structures.

Sequence-based MT: Since the breakthrough made by IBM on word-based models in the 1990s (Brown et al., 1993), SMT has developed rapidly. The PB model (Koehn et al., 2003) advanced the state-of-the-art by translating multi-word units, which makes it better able to capture local phenomena. However, a major drawback in PBMT is that only continuous phrases are considered. Galley and Manning (2010) extend PBMT by allowing discontinuity. However, without linguistic structure information such as syntax trees, sequence-based models can learn a large amount of phrases which may be unreliable.

Tree-based MT: Compared to sequences, trees provide recursive structures over sentences and can handle long-distance relations. Typically, trees used in SMT are either phrasal structures (Galley et al., 2004; Liu et al., 2006; Marcu et al., 2006) or dependency structures (Menezes and Quirk, 2005; Xiong et al., 2007; Xie et al., 2011; Li et al., 2014). However, conventional tree-based models only use linguistically well-formed phrases. Although they are more reliable in theory, discarding all phrase pairs which are not linguistically motivated is an overly harsh decision. Therefore, exploring more translation rules usually can significantly improve translation performance (Marcu et al., 2006; DeNeefe et al., 2007; Wang et al., 2007; Mi et al., 2008).

Graph-based MT: Compared to sequences and trees, graphs are more general and can represent more relations between words. In recent years, graphs have been drawing quite a lot of attention from researchers. Jones et al. (2012) propose a hypergraph-based translation model where hypergraphs are taken as a meaning representation of sentences. However, large corpora with annotated hypergraphs are not readily available for MT. Li et al. (2015) use an edge replacement grammar to translate dependency graphs which are converted from dependency trees by labeling edges. However, their model only focuses on subgraphs which cover continuous phrases.

7 Conclusion

In this paper, we extend the conventional phrase-based translation model by allowing discontinuous phrases. We use graphs which combine bigram and dependency relations together as inputs and present a graph-based translation model. Experiments on Chinese–English and German–English show our model to be significantly better than the phrase-based model as well as other more sophisticated models. In addition, we present a graph segmentation model to explicitly guide the selection of subgraphs. In experiments, this model further improves our system.

In the future, we will extend this model to allow discontinuity on target sides and explore the possibility of directly encoding reordering information in translation rules. We are also interested in using graphs for neural machine translation to see how it can translate and benefit from graphs.

Acknowledgments

This research has received funding from the People Programme (Marie Curie Actions) of the European Union’s Framework Programme (FP7/2007-2013) under REA grant agreement n° 317471 and the European Union’s Horizon 2020 research and innovation programme under grant agreement n° 645452 (QT21). The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. The authors thank all anonymous reviewers for their insightful comments and suggestions.

References


D1.1: Semantic Translation Models


Using Factored Word Representation in Neural Network Language Models

Jan Niehues, Thanh-Le Ha, Eunah Cho and Alex Waibel
Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.secondname@kit.edu

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 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 as well as their impact on the overall machine translation performance. This is especially helpful for morphologically rich languages which have a large vocabulary size.

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 as well as their impact on the overall machine translation performance. This is especially helpful for morphologically rich languages which have a large vocabulary size.

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.

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 (Mediani 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\(^7\) 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 tuple 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.

\(^7\)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 $h = (f_1, \ldots, f_1, D), \ldots, (f_1, \ldots, f_1, D)$, 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. These probabilities can be used as features for, e.g., 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 $f_{i+1}, \ldots, f_{i+D}$, the input to the RNN is the previous target word $w_i = f_{i1}, \ldots, f_{iD}$. 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 $s_{a(i+1)} = (f_{s_{a(i+1)}}, 1, \ldots, f_{s_{a(i+1)}})$ to the ones of the target word $w_i$ and create a new bilingual token

$$b_i = (f_{i1}, \ldots, f_{iD}) \cdot f_{a(i+1)}, 1, \ldots, f_{a(i+1)},$$

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).

Table 1: Features

<table>
<thead>
<tr>
<th>Feature</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 List-Net 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.

The baseline model using words as input and words as output reaches a BLEU score of 27.88. If we instead represent the input words by factors, we select entries from the n-best list that generates a BLEU score of 28.46. As done with the n-gram language models, we can also predict the other factors instead of the words themselves. In all cases, we use all four factors as input factors. We used the word surface form, the POS tags and word clusters using 100 and 1,000 classes.

The baseline model using words as input and words as output reaches a BLEU score of 27.88. If we instead represent the input words by factors, we select entries from the n-best list that generates a BLEU score of 28.46. As done with the n-gram language models, we can also predict the other factors instead of the words themselves. In all cases, we use all four factors as input factors. We used the word surface form, the POS tags and word clusters using 100 and 1,000 classes.

In Table 2, 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

<table>
<thead>
<tr>
<th>Input Prediction Single Score</th>
<th>Word Word 27.88</th>
<th>All factors Word 28.46</th>
<th>All factors POS 28.48</th>
<th>All factors 100 Cl. 28.23</th>
<th>All factors 1,000 Cl. 28.49</th>
<th>All factors All factors 28.54</th>
</tr>
</thead>
</table>

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>
<thead>
<tr>
<th>Model</th>
<th>Conf1</th>
<th>Conf2</th>
<th>Conf3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>29.86</td>
<td>30.00</td>
<td>29.75</td>
</tr>
<tr>
<td>LM 5K</td>
<td>29.79</td>
<td>29.84</td>
<td>29.73</td>
</tr>
<tr>
<td>LM 50K</td>
<td>29.64</td>
<td>29.84</td>
<td>29.83</td>
</tr>
<tr>
<td>Factored LM 5K</td>
<td>29.94</td>
<td>30.01</td>
<td>30.01</td>
</tr>
<tr>
<td>Factored LM 50K</td>
<td>30.05</td>
<td>30.27</td>
<td>30.29</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 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-grain 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
based on 100 and 1,000 word classes respectively. First, we will evaluate the performance of the system using only this model in rescoring. The results are summarized in Table 7.

<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>

In a last series of experiments, we used the scores combined with the baseline scores. The results are shown in Table 8. In this language pair, we can improve over the baseline system by using both models. The final BLEU score is 0.3 BLEU points better than the initial system.

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.

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.

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.


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. An open-source implementation of our neural MT system is available¹, as are sample files and configurations².

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. Lemmatisation 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 .
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) topologizes 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 architecture by Bahdanau et al. (2015), which we will briefly summarize here.

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 (Choi et al., 2014) that reads an input sequence \( x = (x_1, \ldots, x_m) \) and calculates a forward sequence of hidden states \( \overrightarrow{h}_1, \ldots, \overrightarrow{h}_m \), and a backward sequence \( \overleftarrow{h}_1, \ldots, \overleftarrow{h}_m \). The hidden states \( \overrightarrow{h}_i \) and \( \overleftarrow{h}_i \) are concatenated to obtain the annotation vector \( h_i \).

The decoder is a recurrent neural network that predicts a target sequence \( y = (y_1, \ldots, y_n) \). 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 \( e_i \). \( e_i \) is computed as a weighted sum of the annotations \( h_1 \). 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 tutorial\(^4\). 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 Kirschhoff, 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):

\[
\overrightarrow{h}_j = \tanh(\overrightarrow{W} x_j + \overrightarrow{U} \overrightarrow{h}_{j-1})
\]

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

We generalize this to an arbitrary number of features \( |F| \):

\[
\overrightarrow{h}_j = \tanh(\overrightarrow{W} (\bigoplus_{k=1}^{|F|} E_k x_j) + \overrightarrow{U} \overrightarrow{h}_{j-1})
\]

where \( \| \) is the vector concatenation, \( E_k \in \mathbb{R}^{m_k \times K_e} \) are the feature embedding matrices, with \( \sum_{k=1}^{|F|} m_k = m \), and \( K_e \) is the vocabulary size of the \( f \)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.

### 3 Linguistic Input Features

Our generalized model of the previous section supports an arbitrary number of input features.

\(^4\)https://github.com/nyu-dl/dl4mt-tutorial
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 training data consists of about 4.2 million sentence pairs.
To enable open-vocabulary translation, we encode words via joint BPE\(^5\) (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 CHR\(^3\) (Popović, 2015), a character n-gram F\(_3\) score which was found to correlate well with human judgments, especially for translations out of English (Stanojević et al., 2015).\(^6\) The two metrics may occasionally disagree, partly because they are highly sensitive to the length of the output. BLEU is precision-based, whereas CHR\(^3\) 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 layer 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 special UNK symbol.

Sennrich et al. (2016b) report large gains from using monolingual in-domain training data, auto-
matically 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-resource 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 (subj and obja), 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 (PB-SMT) 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 to 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 for different systems (Figure 2). We can see that perplexity is consistently lower for the systems

---

\(^7\) The corpora are available at [http://statmt.org/rsennrich/wmt16_backtranslation/](http://statmt.org/rsennrich/wmt16_backtranslation/)
Table 2: German ↔ English translation results: best perplexity on dev (newstest2013), and BLEU and ChrF3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different (p < 0.05) from respective baseline are marked with (*).

Table 3: Contrastive experiments with individual linguistic features: best perplexity on dev (newstest2013), and BLEU and ChrF3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different (p < 0.05) from respective baseline are marked with (*).

Table 4: German ↔ English translation results with additional, synthetic training data: best perplexity on dev (newstest2013), and BLEU and ChrF3 on test15 (newstest2015) and test16 (newstest2016). BLEU scores that are significantly different (p < 0.05) from respective baseline are marked with (*).
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→Romanian translation results: best perplexity on newsdev2016, and BLEU and ChrF3 on newstest2016. BLEU scores that are significantly different (p < 0.05) from respective baseline are marked with (*).

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 perform multi-source translation (Zoph and Knight, 2016). The main technical difference is that in our approach, the encoder and attention layers are shared between features, which we deem appropriate for the types of features that we tested.

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 newstes2016 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).


Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders

Anonymous ACL submission

Abstract

Current approaches to learning vector representations of text that are compatible between different languages usually require some amount of parallel text, aligned at word, sentence or at least document level. We hypothesize however, that different natural languages share enough semantic structure that it should be possible, in principle, to learn compatible vector representations just by analyzing the monolingual distribution of words.

In order to evaluate this hypothesis, we propose a scheme to map word vectors trained on a source language to vectors semantically compatible with word vectors trained on a target language using an adversarial autoencoder.

We present preliminary qualitative results and discuss possible future developments of this technique, such as applications to cross-lingual sentence representations.

1 Introduction

Distributed representations that map words, sentences, paragraphs or documents to vectors real numbers have proven extremely useful for a variety of natural language processing tasks (Bengio et al., 2006; Collobert and Weston, 2008; Turian et al., 2010; Maas et al., 2011; Mikolov et al., 2013b; Socher et al., 2013; Pennington et al., 2014; Levy and Goldberg, 2014; Le and Mikolov, 2014; Baroni et al., 2014; Levy et al., 2015), as they provide an effective way to inject in machine learning models general prior knowledge about language automatically obtained from inexpensive unannotated corpora. Based on the assumption that different languages share a similar semantic structure, various approaches succeeded to obtain distributed representations that are compatible across multiple languages, either by learning mappings between different embedding spaces (Mikolov et al., 2013a; Faruqui and Dyer, 2014) or by jointly training cross-lingual representations (Klementiev et al., 2012; Hermann and Blunsom, 2013; Chandar et al., 2014; Gouws et al., 2014). These approaches all require some amount of parallel text, aligned at word level, sentence level or at least document level, or some other kind of parallel resources such as dictionaries (Ammar et al., 2016).

In this work we explore whether the assumption of a shared semantic structure between languages is strong enough that it allows to induce compatible distributed representations without using any parallel resource. We only require monolingual corpora that are thematically similar between languages in a general sense.

We hypothesize there exist suitable vectorial spaces such that each language can be viewed as a random process that produces vectors at some level of granularity (words, sentences, paragraphs, documents) which are then encoded as discrete surface forms, and we hypothesize that, if languages are used to convey thematically similar information in similar contexts, these random processes should be approximately isomorphic between languages, and that this isomorphism can be learned from the statistics of the realizations of these processes, the monolingual corpora, in principle without any form of explicit alignment.

We motivate this hypothesis by observing that humans, especially young children, who acquire multiple languages, can often do so with relatively little exposure to explicitly aligned parallel linguistic information, at best they may have access to distant and noisy alignment information in the form of multisensory environmental cues. Nevertheless, multilingual speakers are always au-
languages. Let $d$ be the embedding dimensionality, $G_{\theta_G} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be the generator parametrized by $\theta_G$, $D_{\theta_D} : \mathbb{R}^d \rightarrow \{0 \ldots 1\}$ be the discriminator parametrized by $\theta_D$.

At each training step:

1. draw a sample $\{f\}_n$ of $n$ source embeddings, according to their (adjusted) word frequencies

2. transform them into target-like embeddings $\{\hat{e}\}_n = G_{\theta_G}(\{f\}_n)$

3. evaluate them with the discriminator, estimating their probability of being sampled from the true target distribution $\{p\}_n = D_{\theta_D}(\{\hat{e}\})$

4. update the generator parameters $\theta_G$ to reduce the average adversarial loss $L_a = -\log(\{p\}_n)$

5. draw a sample $\{e\}_n$ of $n$ true target embeddings

6. update the discriminator parameters $\theta_D$ to reduce its binary cross-entropy loss on the classification between $\{e\}_n$ (positive class) and $\{\hat{e}\}$ (negative class)

repeat these steps until convergence.

Unfortunately we found that in this setup, even with different network architectures and hyperparameters, the model quickly converges to a pathological solution where the generator always emits constant or near-constant samples that somehow can fool the discriminator. This appears to be an extreme case of the know mode-seeking issue of GANs (Radford et al., 2015; Theis et al., 2015), which is probably exacerbated in our settings because of the point-mass nature of our probability distributions where each word embedding is a mode on its own.

In order to avoid these pathological solutions, we needed a way to penalize the generator for destroying too much information about its input.

2 Learning word embedding cross-lingual mappings with adversarial autoencoders

The problem of learning transformations between probability distributions of real vectors has been studied in the context of generative neural network models, with approaches such as Generative Moment Matching Networks (GMMNs) (Li et al., 2015) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). In this work we consider the latter.

In a typical GAN, we wish to train a generator model, usually a neural network, to transform samples from a known, easy to sample, uninformative distribution (e.g. Gaussian or uniform) into samples distributed according to a target distribution defined implicitly by a training set. In order to do so, we iteratively alternate training a differentiable discriminator model, also a neural network, to distinguish between training samples and artificial samples produced by the generator, and training the generator to fool the discriminator into misclassifying the artificial examples as training examples. This can be done with conventional gradient-based optimization because the discriminator is differentiable thus it can backpropagate gradients into the generator.

It can be proven that, with sufficient model capacity and optimization power, sufficient entropy of the generator input distribution, and in the limit of infinite training set size, the generator learns to produce samples from the correct distribution. Intuitively, if there is any computable test that allows to distinguish the artificial samples from the training samples with better than random guessing probability, then a sufficiently powerful discriminator will eventually learn to exploit it and then a sufficiently powerful generator will eventually learn to counter it, until the generator output distribution becomes undistinguishable from the true training distribution. In practice, actual models have finite capacity and gradient-based optimization algorithms can become unstable or stuck when applied to this multi-objective optimization problem, though they have been successfully used to generate fairly realistic-looking images (Denton et al., 2015; Radford et al., 2015).

In our preliminary experiments we attempted to adapt GANs to our problem, by training the generator to learn a transformation between word embeddings trained on different languages. Let $d$ be the embedding dimensionality, $G_{\theta_G} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be the generator parametrized by $\theta_G$, $D_{\theta_D} : \mathbb{R}^d \rightarrow \{0 \ldots 1\}$ be the discriminator parametrized by $\theta_D$.

At each training step:

1. draw a sample $\{f\}_n$ of $n$ source embeddings, according to their (adjusted) word frequencies

2. transform them into target-like embeddings $\{\hat{e}\}_n = G_{\theta_G}(\{f\}_n)$

3. evaluate them with the discriminator, estimating their probability of being sampled from the true target distribution $\{p\}_n = D_{\theta_D}(\{\hat{e}\})$

4. update the generator parameters $\theta_G$ to reduce the average adversarial loss $L_a = -\log(\{p\}_n)$

5. draw a sample $\{e\}_n$ of $n$ true target embeddings

6. update the discriminator parameters $\theta_D$ to reduce its binary cross-entropy loss on the classification between $\{e\}_n$ (positive class) and $\{\hat{e}\}$ (negative class)

repeat these steps until convergence.

Unfortunately we found that in this setup, even with different network architectures and hyperparameters, the model quickly converges to a pathological solution where the generator always emits constant or near-constant samples that somehow can fool the discriminator. This appears to be an extreme case of the know mode-seeking issue of GANs (Radford et al., 2015; Theis et al., 2015), which is probably exacerbated in our settings because of the point-mass nature of our probability distributions where each word embedding is a mode on its own.

In order to avoid these pathological solutions, we needed a way to penalize the generator for destroying too much information about its input.
Therefore we turned our attention to Adversarial Autoencoders (AAE) (Makhzani et al., 2015). In an AAE, the generator, now called encoder, is paired with another model, the decoder $R_{\theta_n}$ : $\mathcal{R}^{d} \rightarrow \mathcal{R}^{d}$ parametrized by $\theta_n$ which attempts to transform the artificial samples emitted by the encoder back into the input samples. The encoder and the decoder are jointly trained to minimize a combination of the average reconstruction loss $L_{f} (\{ x \}_n, R_{\theta_n} (G_{\theta_2} (\{ f \}_n)))$ and the adversarial loss defined as above. The discriminator is trained as above. In the original formulation of the AAE, the discriminator is used to enforce a known prior (e.g. Gaussian or Gaussian mixture) on the intermediate, latent representation, in our setting instead we use it to match the latent representation to the target embedding distribution so that the encoder can be used to transform source embeddings into target ones.

In our experiments, we use the cosine dissimilarity as reconstruction loss, and as a further penalty we also include the pairwise cosine dissimilarity between the generated latent samples $\{ z \}$ and the true target samples $\{ e \}_n$. Therefore, the total loss incurred by the encoder-decoder at each step is

$$L_{GR} = \lambda_r L_r (\{ f \}_n, R_{\theta_n} (G_{\theta_2} (\{ f \}_n))) - \lambda_\phi \log (p) + \lambda_c L_r (\{ e \}_n, G_{\theta_2} (\{ f \}_n))$$

where $\lambda_r, \lambda_\phi$ and $\lambda_c$ are hyperparameters (all set equal to 1 in our experiments).

## 3 Experiments

We performed some preliminary exploratory experiments on our model. In this section we report salient results.

The first experiment is qualitative, to assess whether our model is able to learn any semantically sensible transformation at all. We consider English to Italian embedding mapping.

We train English and Italian word embeddings on randomly subsampled Wikipedia corpora consisting of about 1.5 million sentences per language. We use word2vec (Mikolov et al., 2013b) in skipgram mode to generate embeddings with dimension $d = 100$. Our encoder and decoder are linear models with tied matrices (one the transpose of the other), initialized as random orthogonal matrices (we also explored deep non-linear autoencoders but we found that they make the optimization more difficult without providing apparent benefits).

Our discriminator is a Residual Network (He et al., 2015) without convolutions, one leaky ReLU non-linearity (Maas et al., 2013) per block, no non-linearities on the passthrough path, batch normalization (Ioffe and Szegedy, 2015) and dropout (Srivastava et al., 2014). The block (layer) equation is: $h_{t+1} = \phi (W_t \times h_{t-1}) + h_{t-1}$ where $W_t$ is a weight matrix and $\phi$ is batch normalization (with its internal parameters) followed by leaky ReLU and $h_t$ is a $k$-dimensional block state (in our experiments $k = 40$). The network has $T = 10$ blocks followed by a 1-dimensional output layer with logistic sigmoid activation. We found that using a Residual Network as discriminator rather than a conventional multi-layer perceptron yields to larger gradients being backpropagated to the generator, facilitating training.

At each step, word embeddings are sampled according to their frequency in the original corpora, adjusted to subsample frequent words, as in word2vec. Updates are performed using the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001 for the encoder-decoder and 0.01 for the discriminator.


---

1. Code with full hyperparameters will be released with an open source license.
We further evaluate our model on German to English and English to German embedding transformations, using the same evaluation setup as (Klementiev et al., 2012) with embeddings trained on the concatenation of the Reuters corpora and the News Commentary 2015 corpora, with embedding dimension $d = 40$ and discriminator depth $T = 4$. On a qualitative analysis notice similar partial semantic similarity patterns. However the cross-lingual document classification task we were able to improve over the baseline only for the smallest training set size.

4 Discussion and future work

From the qualitative analysis of the word embedding mappings, it appears that the model does learn to transfer some semantic information, although it’s not competitive with other cross-lingual representation approaches. This may be possibly an issue of hyperparameter choice and architectural details, since, to our knowledge, this is the first work to apply adversarial training techniques to point-mass distribution arising from NLP tasks. Further experimentation is needed to determine whether the model can be improved or whether we already hit a fundamental limit on how much semantic transfer can be performed by monolingual distribution matching alone.

Even if semantic transfer by monolingual text alone turns out to be infeasible, this technique might help in conjunction with training on parallel data. For instance, in neural machine translation “sequence2sequence” transducers without attention (Cho et al., 2014), it could be useful to train as usual on parallel sentences and train in autoencoder mode on monolingual sentences, using an adversarial loss computed by a discriminator on the intermediate latent representations to push them to be isomorphic between languages. A modification of this technique that allows for the latent representation to be variable-sized could also be applied to the attentive “sequence2sequence” transducers (Bahdanau et al., 2014), as an alternative or in addition to monolingual dataset augmentation by backtranslation (Sennrich et al., 2015).

In conclusion we believe that this work initiates a potentially promising line of research in natural language processing consisting of applying distribution matching techniques such as adversarial training to learn isomorphisms between languages.

References


In conclusion we believe that this work initiates a potentially promising line of research in natural language processing consisting of applying distribution matching techniques such as adversarial training to learn isomorphisms between languages.


MULTILINGUAL IMAGE DESCRIPTION WITH NEURAL SEQUENCE MODELS

Desmond Elliott∗
ILLC, University of Amsterdam; Centrum Wiskunde & Informatica
d.elliott@uva.nl

Stella Frank∗
ILLC, University of Amsterdam
s.c.frank@uva.nl

Eva Hasler
Department of Engineering, University of Cambridge
ech57@cam.ac.uk

ABSTRACT

We introduce multilingual image description, the task of generating descriptions of images given data in multiple languages. This can be viewed as visually-grounded machine translation, allowing the image to play a role in disambiguating language. We present models for this task that are inspired by neural models for image description and machine translation. Our multilingual image description models generate target-language sentences using features transferred from separate models: multimodal features from a monolingual source-language image description model and visual features from an object recognition model. In experiments on a dataset of images paired with English and German sentences, using BLEU and Meteor as a metric, our models substantially improve upon existing monolingual image description models.

1 INTRODUCTION

Automatic image description — the task of generating natural language sentences for an image — has thus far been exclusively performed in English, due to the availability of English datasets. However, the applications of automatic image description, such as text-based image search or providing image alt-texts on the Web for the visually impaired, are also relevant for other languages. Current image description models are not inherently English-language specific, so a simple approach to generating descriptions in another language would be to collect new annotations and then train a model for that language. Nonetheless, the wealth of image description resources for English suggest a cross-language resource transfer approach, which is what we explore here. In other words: How can we best use resources for Language A when generating descriptions for Language B?

We introduce multilingual image description and present a multilingual multimodal image description model for this task. Multilingual image description is a form of visually-grounded machine translation, in which parallel sentences are grounded against features from an image. This grounding can be particularly useful when the source sentence contains ambiguities that need to be resolved in the target sentence. For example, in the German sentence “Ein Rad steht neben dem Haus”, “Rad” could refer to either “bicycle” or “wheel”, but with visual context the intended meaning can be more easily translated into English. In other cases, source language features can be more precise than noisy image features, e.g. in identifying the difference between a river and a harbour.

Our multilingual image description model adds source language features to a monolingual neural image description model (Karpathy & Fei-Fei, 2015; Vinyals et al., 2015, inter-alia). Figure 1 de-

∗Authors contributed equally to this paper.
D1.1: Semantic Translation Models
Under review as a conference paper at ICLR 2016

Figure 1: An illustration of the multilingual multimodal language model. Descriptions are generated by combining features from source- and target-language multimodal language models. The dashed lines denote variants of the model: removing the CNN features from a source model would create language-only conditioning vectors; whereas removing the CNN input in the decoder assumes the source feature vectors know enough about the image to generate a good description.

The models that include visual features also improve over our translation baselines, although to a lesser extent; we attribute this to the dataset being exact translations rather than independently elicited descriptions, leading to high performance for the translation baseline. Our analyses show that the additional features improve mainly lower-quality sentences, indicating that our best models successfully combine multiple noisy input modalities.

2 MODELS

Our multilingual image description models are neural sequence generation models, with additional inputs from either visual or linguistic modalities, or both. We present a family of models in sequence of increasing complexity to make their compositional character clear, beginning with a neural sequence model over words and concluding with the full model using both image and source features. See Figure 2 for a depiction of the model architecture.

2.1 RECURRENT LANGUAGE MODEL (LM)

The core of our model is a Recurrent Neural Network model over word sequences, i.e., a neural language model (LM) (Mikolov et al., 2010). The model is trained to predict the next word in the sequence, given the current sequence seen so far. At each timestep $t$ for input sequence $w_0, \ldots, w_t$, the input word $w_t$, represented as a one-hot vector over the vocabulary, is embedded into a high-dimensional continuous vector using the learned embedding matrix $W_{e}$ (Eqn 1). A nonlinear function $f$ is applied to the embedding combined with the previous hidden state to generate the hidden state $h_t$ (Eqn 2). At the output layer, the next word $o_t$ is predicted via the softmax function over the
Figure 2: Our multilingual multimodal model predicts the next word in the description $o_n$ given the current word $x_i$ and the hidden state $h_i$. Source language and image features can be additional input to the model (signified by dashed arrows). The source features, shown rolled-up to save space, are transferred from a multimodal language model or a language model; see Section 2 for more details.

$$v_i = W_{vh} v_i$$ (1)

$$\epsilon_i = W_{wh} \epsilon_i$$ (2)

$$o_i = \text{softmax}(W_{ho} h_i)$$ (3)

In simple RNNs, $f$ in Eqn 2 can be the tanh or sigmoid function. Here, we use an LSTM\(^1\) to avoid problems with longer sequences (Hochreiter & Schmidhuber, 1997). Sentences are buffered at timestep 0 with a special beginning-of-sentence marker and with an end-of-sequence marker at timestep $n$. The initial hidden state values $h_{-1}$ are learned, together with the weight matrices $W$.

### 2.2 Multimodal Language Model (MLM)

The Recurrent Language Model (LM) generates sequences of words conditioned only on the previously seen words (and the hidden layer), and thus cannot use visual input for image description. In the multimodal language model (MLM), however, sequence generation is additionally conditioned on image features, resulting in a model that generates word sequences corresponding to the image. The image features $v$ (for visual) are input to the model at $h_0$ at the first timestep\(^2\):

$$h_0 = f(W_{hh} h_{-1} + W_{he} e_0 + W_{vh} v)$$ (4)

### 2.3 Translation Model (SOURCE-LM $\rightarrow$ TARGET-LM)

Our translation model is analogous to the multimodal language model above: instead of adding image features to our target language model, we add features from a source language model. This feature vector $s$ is the final hidden state extracted from a sequence model over the source language, the SOURCE-LM. The initial state for the TARGET-LM is thus defined as:

$$h_0 = f(W_{hs} h_{-1} + W_{he} e_0 + W_{hs} s)$$ (5)

We follow recent work on sequence-to-sequence architectures for neural machine translation (Cho et al., 2014; Sutskever et al., 2014) in calling the source language model the ‘encoder’ and the target language model the ‘decoder’. However, it is important to note that the source encoder is

\(^1\)The LSTM produced better validation performance than a Gated Recurrent Unit (Cho et al., 2014).

\(^2\) Adding the image features at every timestep reportedly results in overfitting (Karpathy & Fei-Fei, 2015; Vinyals et al., 2015), with exception of the m-RNN (Mao et al., 2015).
2.4 MULTILINGUAL MULTIMODAL MODEL (SOURCE-MLM → TARGET-MLM)

Finally, we can use both the image and the source language features in a combined multimodal translation model. If the image features are input on both the source and the target side, this results in a doubly multimodal multilingual model (SOURCE-MLM → TARGET-MLM). There are two alternative formulations: image features are input only to the source (SOURCE-MLM → TARGET-LM) or only the target model (SOURCE-LM → TARGET-MLM). The initial state of the TARGET-MLM, regardless of source model type, is:

\[ h_0 = f(W_{hA}h_{-1} + W_{hR}r_0 + W_{hS}s + W_{hv}v) \]  

(6)

2.5 GENERATING SENTENCES

We use the same description generation process for each model. First, a model is initialised with the special beginning-of-sentence token and any image or source features. At each timestep, the generated output is the maximum probability word at the softmax layer, \( o_i \), which is subsequently used as the input token at timestep \( i+1 \). This process continues until the model generates the end-of-sentence token, or a pre-defined number of timesteps (30, in our experiments, which is slightly more than the average sentence length in the training data).

3 METHODOLOGY

3.1 DATA

We use the IAPR-TC12 dataset, originally introduced in the ImageCLEF shared task for object segmentation and later expanded with complete image descriptions (Grubinger et al., 2006). This dataset contains 20,000 images with multiple descriptions in both English and German. Each sentence corresponds to a different aspect of the image, with the most salient objects likely being described in the first description (annotators were asked to describe parts of the image that hadn’t been covered in previous descriptions). We use only the first description of each image. Note that the English descriptions are the originals; the German data was professionally translated from English. Figure 3 shows an example image-bitext tuple from the dataset. We perform experiments using the standard splits of 17,665 images for training, from which we reserve 10% for hyperparameter estimation, and 1,962 for evaluation.

The descriptions are lowercased and tokenised using the ptbtokenizer.py script from the MS COCO evaluation tools. We discarded words in the training data observed fewer than 3 times. This
leaves a total of 272,172 training tokens for English over a vocabulary of 1,763 types; and 223,137 tokens for German over 2,374 types. Compared to the Flickr8K, Flickr30K, or MS COCO datasets, the English descriptions in the IAPR-TC12 dataset are long, with an average length of 23 words.¹

We extract the image features from the pre-trained VGG-16 CNN object recognition model (Simonyan & Zisserman, 2015). Specifically, our image features are extracted as fixed representations from the penultimate layer of the CNN, in line with recent work in this area.

3.2 BASELINES
MLM: the first baseline is a monolingual image description model, i.e. a multimodal language model for the target language with no source language features, but with image features.

SOURCE-LM → TARGET-LM: the second baseline is our translation model trained on only source and target descriptions without visual features. The final hidden state of the SOURCE-LM, after it has generated the source sentence, is input to the TARGET-LM.

3.3 MULTILINGUAL MULTIMODAL MODEL VARIANTS
SOURCE-MLM → TARGET-MLM: In this model, both of the LMs in the translation baseline are replaced with multimodal language models. The source features input to the target model are thus multimodal, i.e. they are word and image features captured over the source-language sentence. The target decoder is also conditioned on the image features directly. Note that the source and target W_vh matrices are parameterised separately.

SOURCE-LM → TARGET-MLM: The source language features are generated by a LM; visual features are input only in the target model.

SOURCE-MLM → TARGET-LM: Visual input is given only to the SOURCE-MLM and the TARGET-LM uses a single input vector from the SOURCE-MLM. This source encoder combines both linguistic and visual cues, to the extent that the visual features are represented in the SOURCE-MLM feature vector.

3.4 HYPERPARAMETERS
We use an LSTM (Hochreiter & Schmidhuber, 1997) as f in the recurrent language model. The hidden layer size |h| is set to 256 dimensions. The word embeddings are 256-dimensional and learned along with other model parameters. We also experimented with larger hidden layers (as well as with deeper architectures), and while that did result in improvements, they also took longer to train. The image features v are the 4096-dimension penultimate layer of the VGG-16 object recognition network (Simonyan & Zisserman, 2015) applied to the image.

3.5 TRAINING AND OPTIMISATION
The models are trained with mini-batches of 100 examples towards the objective function (cross-entropy of the predicted words) using the ADAM optimiser (Kingma & Ba, 2014). We do early stopping for model selection based on BLEU4: if validation BLEU4 has not increased for 10 epochs, and validation language model perplexity has stopped decreasing, training is halted.

We apply dropout over the image features, source features, and word representations with p = 0.5 to discourage overfitting (Srivastava et al., 2014). The objective function includes an L2 regularisation term with λ=1e⁻⁸.

All results reported are averages over three runs with different Glorot-style uniform weight initialisations (Glorot & Bengio, 2010). We report image description quality using BLEU4 (Papineni et al., 2002), Meteor (Denkowski & Lavie, 2014), and language-model perplexity. Meteor has been shown to correlate better with human judgements than BLEU4 for image description (Elliott & Keller, 2014). The BLEU4 and Meteor scores are calculated using MultiEval (Clark et al., 2011).

¹This difference in length resulted in difficulties in initial experiments with pre- or co-training using other datasets. We plan on pursuing this further in future work, since the independence of the source encoder in our model makes this kind of transfer learning very natural.
The results for image description in both German and English are presented in Tables 1 and 2; generation examples can be seen in Figures 6, 7, 8 in Appendix B. To our knowledge, these are the first published results for German image description. Overall, we found that English image description is easier than German description, as measured by BLEU-4 and Meteor scores. This may be caused by the more complex German morphology, which results in a larger vocabulary and hence more model parameters.

The English monolingual image description model (En-MLM) is comparable with state-of-the-art models, which typically report results on the Flickr8K / Flickr30K dataset. En-MLM achieves a BLEU-4 score of 15.8 on the Flickr8K dataset, nearly matching the score from Karpathy & Fei-Fei (2015) (16.0), which uses an ensemble of models and beam search decoding. On the IAPR-TC12 dataset, the En-MLM baseline outperforms Kiros et al. (2014). Mao et al. (2015) report higher performance, but evaluate on all reference descriptions, making the figures incomparable.

All multilingual models beat the monolingual image description baseline, by up to 8.9 BLEU-4 and 8.8 Meteor points for the best models. Clearly the features transferred from the source models are useful for the TARGET-LM or TARGET-MLM description generator, despite the switch in languages.

The translation baseline without visual features performs very well. This indicates the effectiveness of our translation model, even without joint training, but is also an artifact of the dataset. A different dataset with independently elicited descriptions (rather than translations of English descriptions) may result in worse performance for a translation system that is not visually grounded, because the target descriptions would only be comparable to the source descriptions.

Overall, the multilingual models that encode the source using an MLM outperform the SOURCE-LM models. On the target side, simple LM decoders perform better than MLM decoders. This can be explained to some extent by the smaller number of parameters in models that do not input the visual features twice. Incorporating the image features on the source side seems to be more effective, possibly because the source is constrained to the gold description at test time, leading to a more

---

1 Visit https://staff.fnwi.uva.nl/d.elliott/GroundedTranslation/ to see 1,766 examples generated by each model for the validation data.

2Kiros et al. (2014) report BLEU-1-2-3, their best model is reported at 9.8 BLEU-3.

3 The BLEU-4 and Meteor scores in Table 2 for En LM → De LM and En MLM → De LM are not significantly different according to the MultEval approximate randomization significance test.
coherent match between visual and linguistic features. Conversely, the TARGET-MLM variants tend to be worse sentence generators than the LM models, indicating that while visual features lead to useful hidden state values, there is room for improving their role during generation.

5 DISCUSSION

What do source features add beyond image features? Source features are most useful when the baseline MLM does not successfully separate related images. The image description models have to compress the image feature vector into the same number of dimensions as the hidden layer in the recurrent network, effectively distilling the image down to the features that correspond to the words in the description. If this step of the model is prone to mistakes, the resulting descriptions will be of poor quality. However, our best multilingual models are initialised with features transferred from image description models in a different language. In these cases, the source language features have already compressed the image features for the source language image description task.

Qualitatively, we can illustrate this effect using Barnes-Hut t-SNE projections of the initial hidden representations of our models (van der Maaten, 2014). Figure 4 shows the t-SNE projection of the example from Figure 7 using the initial hidden state of an En MLM (left) and the target side of the De MLM → En MLM (right). In the monolingual example, the nearest neighbours of the target image are desert scenes with groups of people. Adding the transferred source features results in a representation that places importance on the background, due to the fact that it is consistently mentioned in the descriptions. Now the nearest neighbours are images of mountainous snow regions with groups of people.

Which descriptions are improved by source or image features? Figure 5 shows the distribution of sentence-level Meteor scores of the baseline models (monolingual MLM and monomodal LM → LM) and the average per-sentence change when moving to our best performing multilingual multimodal model (SOURCE-MLM → TARGET-LM). The additional source language features (compared to MLM) or additional modality (compared to LM → LM) result in similar patterns: low quality descriptions are improved, while the (far less common) high quality descriptions deteriorate.

Adding image features seems to be riskier than adding source language features, which is unsurprising given the larger distance between visual and linguistic space, versus moving from one language to another. This is also consistent with the lower performance of MLM baseline models compared to LM→LM models.
D1.1: Semantic Translation Models

Under review as a conference paper at ICLR 2016

Figure 5: The effect of adding multimodal source features to (a) a monolingual English image description model and (b) a German-English translation model (validation data, averaged over 3 runs). The top plots show the baseline sentence-level Meteor score distributions, while the bottom plots show the difference in score as compared to multilingual multimodal De MLM → En LM. For most sentences, a low baseline score is improved by adding multimodal source features.

An analysis of the LM → MLM model (not shown here) shows similar behaviour to the MLM → LM model above. However, for this model the decreasing performance starts earlier: the LM → MLM model improves over the LM → LM baseline only in the lowest score bin. Adding the image features at the source side, rather than the target side, seems to filter out some of the noise and complexity of the image features, while the essential source language features are retained. Conversely, merging the source language features with image features on the target side, in the TARGET-MLM models, leads to a less helpful entangling of linguistic and noisier image input, maybe because too many sources of information are combined at the same time (see Eqn 6).

6 RELATED WORK

The past few years have seen numerous results showing how relatively standard neural network model architectures can be applied to a variety of tasks. The flexibility of the application of these architectures can be seen as a strong point, indicating that the representations learned in these general models are sufficiently powerful to lead to good performance. Another advantage, which we have exploited in the work presented here, is that it becomes relatively straightforward to make connections between models for different tasks, in this case image description and machine translation.

Automatic image description has received a great deal of attention in recent years (see Bernardi et al. (2016) for a more detailed overview of the task, datasets, models, and evaluation issues). Deep neural networks for image description typically estimate a joint image-sentence representation in a multimodal recurrent neural network (RNN) (Kiros et al., 2014; Donahue et al., 2014; Vinyals et al., 2015; Karpathy & Fei-Fei, 2015; Mao et al., 2015). The main difference between these models and discrete tuple-based representations for image description (Farhadi et al., 2010; Yang et al., 2011; Li et al., 2011; Mitchell et al., 2012; Elliott & Keller, 2013; Yatskar et al., 2014; Elliott & de Vries, 2015) is that it is not necessary to explicitly define the joint representation; the structure of the neural network can be used to estimate the optimal joint representation for the description task. As in our MLM, the image-sentence representation in the multimodal RNN is initialised with image features from the final fully-connected layer of a convolutional neural network trained for multi-class object recognition (Krizhevsky et al., 2012). Alternative formulations input the image features into the model at each timestep (Mao et al., 2015), or first detect words in an image and generate sentences using a maximum-entropy language model (Fang et al., 2015).
In the domain of machine translation, a greater variety of neural models have been used for sub-tasks within the MT pipeline, such as neural network language models (Schwenk, 2012) and joint translation and language models for re-ranking in phrase-based translation models (Le et al., 2012; Auli et al., 2013) or directly during decoding (Devlin et al., 2014). More recently, end-to-end neural MT systems using Long Short-Term Memory Networks and Gated Recurrent Units have been proposed as Encoder-Decoder models for translation (Sutskever et al., 2014; Bahdanau et al., 2015), and have proven to be highly effective (Bojar et al., 2015; Jean et al., 2015).

In the multimodal modelling literature, there are related approaches using visual and textual information to build representations for word similarity and categorization tasks (Silberer & Lapata, 2014; Kiela & Bottou, 2014; Kiela et al., 2015). Silberer & Lapata combine textual and visual modalities by jointly training stacked autoencoders, while Kiela & Bottou construct multi-modal representations by concatenating distributed linguistic and visual feature vectors. More recently, Kiela et al. (2015) induced a bilingual lexicon by grounding the lexical entries in CNN features. In all cases, the results show that the bimodal representations are superior to their unimodal counterparts.

7 CONCLUSIONS

We introduced multilingual image description, the task of generating descriptions of an image given a corpus of descriptions in multiple languages. This new task not only expands the range of output languages for image description, but also raises new questions about how to integrate features from multiple languages, as well as multiple modalities, into an effective generation model.

Our multilingual multimodal model is loosely inspired by the encoder-decoder approach to neural machine translation. Our encoder captures a multimodal representation of the image and the source-language words, which is used as an additional conditioning vector for the decoder, which produces descriptions in the target language. Each conditioning vector is originally trained towards its own objective: the CNN image features are transferred from an object recognition model, and the source features are transferred from a source-language image description model. Our model substantially improves the quality of the descriptions in both directions compared to monolingual baselines.

The dataset used in this paper consists of translated descriptions, leading to high performance for the translation baseline. However, we believe that multilingual image description should be based on independently elicited descriptions in multiple languages, rather than literal translations. Linguistic and cultural differences may lead to very different descriptions being appropriate for different languages (For example, a polder is highly salient to a Dutch speaker, but not to an English speaker; an image of a polder would likely lead to different descriptions, beyond simply lexical choice.) In such cases image features will be essential.

A further open question is whether the benefits of multiple monolingual references extend to multiple multilingual references. Image description datasets typically include multiple reference sentences, which are essential for capturing linguistic diversity within a single language (Rashid et al., 2010; Elliott & Keller, 2013; Hodosh et al., 2013; Chen et al., 2015). In our experiments, we found that useful image description diversity can also be found in other languages instead of in multiple monolingual references.

In the future, we would like to explore attention-based recurrent neural networks, which have been used for machine translation (Bahdanau et al., 2015; Jean et al., 2015) and image description (Xu et al., 2015). We also plan to apply these models to other language pairs, such as the recently released PASCAL 1K Japanese Translations dataset (Funaki & Nakayama, 2015). Lastly, we aim to apply these types of models to a multilingual video description dataset (Chen & Dolan, 2011).

ACKNOWLEDGMENTS

D. Elliott was supported by an Alain Bensoussain Career Development Fellowship. S. Frank is supported by funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement Nr. 645452.

We thank Philip Schulz, Khalid Sima’an, Arjen P. de Vries, Lynda Hardman, Richard Glassey, Wilker Aziz, Joost Bastings, and Ákos Kádár for discussions and feedback on the work. We built the
models using the Keras library, which is built on-top of Theano. We are grateful to the Database Architectures Group at CWI for access to their K20x GPUs.

REFERENCES


Under review as a conference paper at ICLR 2016


Under review as a conference paper at ICLR 2016


Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron C., Salakhutdinov, Ruslan, Zemel, Richard S., and Bengio, Yoshua. Show, attend and tell: Neural image caption generation with visual attention. CoRR, abs/1502.03044, 2015.


Yatskar, M, Vanderwende, L, and Zettlemoyer, L. See no evil, say no evil: Description generation from densely labeled images. SEM, 2014.
## Validation Results

<table>
<thead>
<tr>
<th></th>
<th>BLEU4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>English</strong></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>h</td>
</tr>
<tr>
<td>En MLM</td>
<td>15.99 ± 0.38</td>
</tr>
<tr>
<td>De MLM $\rightarrow$ En MLM</td>
<td>20.63 ± 0.07</td>
</tr>
<tr>
<td>De MLM $\rightarrow$ En LM</td>
<td>27.55 ± 0.41</td>
</tr>
<tr>
<td>De LM $\rightarrow$ En MLM</td>
<td>19.44 ± 0.65</td>
</tr>
<tr>
<td>De LM $\rightarrow$ En LM</td>
<td>23.78 ± 0.71</td>
</tr>
<tr>
<td><strong>German</strong></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>h</td>
</tr>
<tr>
<td>De MLM</td>
<td>11.87 ± 0.37</td>
</tr>
<tr>
<td>En MLM $\rightarrow$ De MLM</td>
<td>16.03 ± 0.35</td>
</tr>
<tr>
<td>En MLM $\rightarrow$ De LM</td>
<td>21.88 ± 0.13</td>
</tr>
<tr>
<td>En LM $\rightarrow$ De MLM</td>
<td>15.42 ± 0.26</td>
</tr>
<tr>
<td>En LM $\rightarrow$ De LM</td>
<td>21.22 ± 0.74</td>
</tr>
</tbody>
</table>

Table 3: Image description performance in the validation data set. It always helps to condition on features from a different language, in both English $\rightarrow$ German and German $\rightarrow$ English. See Sections 3.2 and 3.3 for detailed explanations of the model variants. We report the mean and standard deviation calculated over three runs with random weight initialisation.
B Example Descriptions

We present examples of the descriptions generated by the models studied in this paper. In Figure 6, the monolingual MLM generates the best descriptions. However, in Figures 7 and 8, the best descriptions are generated by transferring source MLM features into a target MLM or a target LM.

**English**

- Ref: three adults and a child are sitting at a brown sandy beach with a few grey stones in the foreground
- MLM: tourists are sitting at a sandy beach with the sea in the background
- MLM→MLM: a group of people is standing and sitting on a brown sandy beach with the sea in the background
- MLM→LM: two women are standing and sitting on a brown slope with green grass in the foreground
- LM→LM: two men and two women are standing on a brown hill with a few trees in the foreground

**German**

- Ref: drei erwachsene und ein kind sitzen an einem braunen sandstrand mit einigen grauen steinen im vordergrund
- MLM: touristen posieren auf einem aus-sichtsplatz mit einem see und einem see im hintergrund
- MLM→MLM: drei männer und zwei frauen sitzen auf einem braunen sandstrand mit dem meer im hintergrund
- MLM→LM: zwei männer und zwei frauen stehen auf einem hellbraunen sandstrand mit einem braunen ufer im vordergrund
- LM→LM: sieben personen stehen und sitzen an einem braunen ufer eines sees im vordergrund

Figure 6: The monolingual MLM models are sufficient to generate accurate descriptions.
Figure 7: The best English descriptions are generated by transferring features from a German MLM model.
English
Ref: view of the houses of a city at a broad river
MLM: a bay with a wooded shore and a bridge in the foreground
MLM→MLM: view of a city with many boats and boats
MLM→LM: view of a river at the edge of a city
LM→LM: view from above of the houses and streets of a city at a river

German
Ref: blick auf die h"auser einer stadt an einem breiten fluss
MLM: blick auf eine stadt mit vielen grauen hochh"aubern und dunkelgr"unen b"aumen
MLM→MLM: blick von oben auf die h"auser und b"aume einer stadt
MLM→LM: blick von oben auf die h"auser einer stadt an einem hafen
LM→LM: blick von oben auf eine stadt an einem hafen

Figure 8: The best English and German descriptions are generated by TARGET-LM decoders with language (SOURCE-LM) or multimodal (SOURCE-MLM) source features.
DCU-UvA Multimodal MT System Report

Iacer Calixto
ADAPT Centre
School of Computing
Dublin City University
Dublin, Ireland
iacer.calixto@adaptcentre.ie

Desmond Elliott
ILLC
University of Amsterdam
Science Park 107
1098 XG Amsterdam
d.elliott@uva.nl

Stella Frank
ILLC
University of Amsterdam
Science Park 107
1098 XG Amsterdam
s.c.frank@uva.nl

Abstract

We present a doubly-attentive multimodal machine translation model. Our model learns to attend to source language and spatial-preserving CONV5,4 visual features as separate attention mechanisms in a neural translation model. In image description translation experiments (Task 1), we find an improvement of 2.3 Meteor points compared to initialising the hidden state of the decoder with only the FC7 features and 2.9 Meteor points compared to a text-only neural machine translation baseline, confirming the useful nature of attending to the CONV5,4 features.

1 Introduction

Our system learns to translate image descriptions using both the source language descriptions and the images. We integrate an attention-based neural network for machine translation and image description in a unified model, in which two separate attention mechanisms operate over the language and visual modalities. We believe that this is a principled approach to learning which source words and which areas of the image to attend to when generating words in the target description.

We are inspired by recent successes in using attentive models in both neural machine translation (NMT) and neural image description. Originally, in non-attentive NMT models, the entire source sentence is encoded into a single vector which is in turn used by the decoder to generate a translation (Cho et al., 2014; Sutskever et al., 2014). In a similar vein, image description models can use a vector encoding the image as input for the description generation process (Vinyals et al., 2015; Mao et al., 2015, inter-alia).

Bahdanau et al. (2014) first proposed a NMT model with an attention mechanism over the source sentence. Their model is trained so that the decoder learns to attend to words in the source sentence when translating each token in the target sentence. Xu et al. (2015) introduced a similar attention-based neural image description model. In this case, the attention mechanism learns which parts of the image to attend to while generating words in the description.

When translating image descriptions, given both the source description and the source image (i.e., the setting for Task 1), we believe that both modalities can provide cues for generating the target language description. The source description provides the content for translation, but in cases where this may be ambiguous, the image features can provide contextual disambiguation. The system we propose is a first step towards integrating both modalities using attention mechanisms.

Previous work has demonstrated the plausibility of multilingual multimodal natural language processing. Elliott et al. (2015) showed how to generate descriptions of images in English and German by learning and transferring features between independent neural image description models. In comparison, our approach is a single end-to-end model over the source and target languages with attention mechanisms over both the source language and the visual features.

2 Model Description

We represent the source language with a bidirectional recurrent neural network (RNN) with a gated recurrent unit (GRU) that computes, for each word, forward and backward source annotation vectors $\vec{h}_i$ and $\vec{h}_i$. The final source annotation vector for a word $b_i$ is the concatenation of both $[\vec{h}_i; \vec{h}_i]$. 

631

We use the visual features released by the shared task organisers, extracted from the pre-trained VGG-19 convolutional neural network (CNN) (Simonyan and Zisserman, 2015). The organisers release two types of visual features according to the layer they were extracted from: FC7 features are extracted from the final fully-connected layer (FC7), which encode information about the entire image in a 4096-dimensional feature vector; and CONV5,4 features, extracted from the final convolutional layer (CONV5,4), namely a 196 x 512 dimensional matrix where each row (i.e., a 512D vector) represents features from a specific spatial ‘patch’ and therefore encodes information about that specific ‘patch’ (i.e., area) of the image.

2.1 FC7-initialised model
In this model, we use visual features from the final fully-connected FC7 layer from the pre-trained VGG-19 CNN. These features represent an abstract summary of the entire image and crucially are not spatially aware, unlike the CONV5,4 features we use in the subsequent double-attention model. We integrate the FC7 features into the initial state of the decoder.

We first affine-transform the 4096D FC7 image feature vector $i$ into the source language bidirectional RNN hidden states dimensionality, where the affine transformation parameters $(W_f, b_f)$ are trained jointly with the model:

$$i_{\text{proj}} = i \cdot W_f + b_f.$$  \hspace{1cm} (1)

We then simply sum these projected image features $i_{\text{proj}}$ with the first source language context vector $h_1$, obtained by the encoder bidirectional RNN, and use the resulting vector as input to a feed-forward neural network $f_{\text{init}}$ used to initialise the decoder hidden state:

$$s_0 = f_{\text{init}} (h_1 + i_{\text{proj}}).$$  \hspace{1cm} (2)

2.2 Doubly-attentive model
The goal of the doubly-attentive model is to integrate separate attention mechanisms over the source language words and visual features in a single decoder. Similarly to the FC7 model, we represent the source language using a bi-directional RNN with GRUs. We use visual features extracted from the CONV5,4 layer of the VGG-19 CNN alongside the FC7 features. The CONV5,4 features consist of a 196 x 512 dimensional matrix, where each row represents features from a specific spatial ‘patch’. Analogous to how the attention mechanism for the source language can focus on specific words or phrases in the source description, the image attention mechanism can focus on specific parts of the image (Xu et al., 2015).

Our doubly-attentive decoder is conditioned on the source sentence and the image via the two separate attention mechanisms, as well as the previous hidden state of the decoder and the previously emitted word. Therefore, in computing the decoder hidden state $s_t$ at time step $t$, the decoder has access to the following information:

- $i_t$ – the image context vector for the current time step obtained via attention over the image representation;
- $c_t$ – the source language context vector for the current time step obtained via attention over the source sentence representation;
- $s_{t-1}$ – the decoder’s previous hidden state;
- $y_{t-1}$ – the target word emitted by the decoder in the previous time step.

Figure 1 illustrates the computation of the decoder hidden state $s_t$ according to our doubly-attentive model.

2.3 Source sequence context vector
To compute the time-dependent source sentence context vector, we follow Bahdanau et al. (2014) and use a single-layer feed-forward network $f_{\text{score}}$ for computing an expected alignment $e_{t,i}^s$ between each source annotation vector $h_i$ — computed as the concatenation of forward and backward source annotation vectors $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ — and the target word to be emitted at the current time step $t$.

$$e_{t,i}^s = f_{\text{score}} (h_i, s_{t-1}, y_{t-1}),$$  \hspace{1cm} (3)

where $f_{\text{score}}$ uses all source annotation vectors $h$, the decoder’s previous hidden state $s_{t-1}$ and the previously emitted word $y_{t-1}$ in computing the expected alignments for the target word at current time step $t$. In Equation 4, these alignments are then normalised and converted into probabilities.

$$\alpha_{t,i} = \frac{\exp (e_{t,i}^s)}{\sum_{j=1}^N \exp (e_{t,j}^s)},$$  \hspace{1cm} (4)
Figure 1: A doubly-attentive decoder learns to independently attend to image patches and source language words when generating translations.

where $\alpha_{t,i}$ are weights representing the attention over the source annotation vectors. The final time-dependent source context vector $c_t$ is a weighted sum over the source annotation vectors, where each vector is weighted by the attention weight $\alpha_{t,i}$:

$$c_t = \sum_{i=1}^{N} \alpha_{t,i} h_i.$$  \hspace{1cm} (5)

### 2.4 Image context vector

The time-dependent image context vectors are based on the “soft” visual attention mechanism (Xu et al., 2015). As outlined above, the image annotation vectors are the features extracted from CONV5_4 layer, resulting in 196 vectors (each corresponding to one of the $14 \times 14$ patches in the image) of 512 dimensions each. These annotation vectors are denoted $a_l$ (with $l = 1 \ldots 196$) and are used analogously to the hidden states $h_i$ of the source sentence encoder.

The expected alignments $e_{l,t}^i$ over the image features are computed by a single layer feed-forward network $f_{\text{score}}$:

$$e_{l,t}^i = f_{\text{score}}(a_l, s_{t-1}, y_{t-1}),$$  \hspace{1cm} (6)

where $f_{\text{score}}$ uses all image annotation vectors $a_l$, the decoder previous hidden state $s_{t-1}$ and the previous emitted word $y_{t-1}$ in computing the expected image–target word alignments at current time step $t$. In Equation 7 these expected alignments are further normalised and converted into probabilities, as in the source context vector.

$$\alpha_{t,i} = \frac{\exp(e_{i,t}^l)}{\sum_{j=1}^{N} \exp(e_{i,t}^j)},$$  \hspace{1cm} (7)

where $\alpha_{t,i}$ are the model’s image attention weights. A time-dependent image context vector $i_t$ is then computed by using these attention weights.

$$i_t = \sum_{l=1}^{N} \alpha_{t,i}^l a_l.$$  \hspace{1cm} (8)

Ideally, this image context vector $i_t$ captures the image patches that are relevant to the current state of the decoder and for generating the next word.

### 3 Experiments

We report results for Task 1, which uses the translated data in the Multi30K corpus (Elliott et al., 2016). English and German descriptions in the Multi30K were normalised and tokenized, and compounds in German descriptions were further split in a pre-processing step$^{1}$.

$^{1}$We use the scripts in the Moses SMT Toolkit to normalise, tokenize and split compounds (Koehn et al., 2007).
Table 1: Results for our models on Task 1. We find that attending over the source language and CONV₅,₄ visual features is better than not using image features (text-only, attentive NMT model) and also just initialising an attention-based decoder with FC₇ features.

<table>
<thead>
<tr>
<th>Moses</th>
<th>52.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONV₅,₄-Multimodal NMT</td>
<td>46.4</td>
</tr>
<tr>
<td>FC₇-Multimodal NMT</td>
<td>44.1</td>
</tr>
<tr>
<td>Text-only Attention NMT</td>
<td>43.5</td>
</tr>
<tr>
<td>Elliott et al. (2015)</td>
<td>24.7</td>
</tr>
</tbody>
</table>

4 Conclusions and Future Work

We present a model which incorporates multiple multimodal attention mechanisms into a neural machine translation decoder. Source language and visual attention mechanisms have been well-studied in the recent literature, but our results indicate that multimodal attention appears to be more complex than simply combining two independent attention mechanisms. In particular, we hoped to find a greater improvement from adding visual features, relative to text-only models. However, the Multi30k dataset is relatively small, with a small vocabulary and simple syntactic structures (Elliott et al., 2016). Whereas SMT models can be trained effectively on such datasets, neural models usually perform best when a large amount of data is available. We believe that as the amount of data in multimodal translation datasets increases, neural models will become more competitive.

In future work we plan to study why the source language attention mechanism contributes more to the model than the visual attention. We believe that using the source language context vector $c_i$ may help when computing the image context vector $i_t$. We also plan to investigate other attention mechanisms, for instance the “hard” attention as proposed by Xu et al. (2015). Soft attention may be too diffuse in this setting, especially over the large set of image context vectors.

Acknowledgements

This research is partially supported by Science Foundation Ireland through the CNGL Programme (Grant 12/CE/12267) in the ADAPT Centre (www.adaptcentre.ie) at Dublin City University. DE was supported by the NWO Vici grant nr. 277-89-002. SF was supported by European Unions Horizon 2020 research and innovation programme under grant agreement nr. 645452.
References


Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2015. Deep captioning with multimodal recurrent neural networks (m-RNN). In ICLR.


CUNI System for WMT16 Automatic Post-Editing and Multimodal Translation Tasks

Jindřich Libovický  Jindřich Helcl
Marek Tlustý  Ondřej Bojar  Pavel Pecina

Charles University in Prague
Malostranské náměstí 25, 112 00 Prague, Czech Republic
{libovicky, helcl, tlusty, bojar, pecina}@ufal.mff.cuni.cz

Abstract

Neural sequence to sequence learning recently became a very promising paradigm in machine translation, achieving competitive results with statistical phrase-based systems. In this system description paper, we attempt to utilize several recently published methods used for neural sequential learning in order to build systems for WMT 2016 shared tasks of Automatic Post-Editing and Multimodal Machine Translation.

1 Introduction

Neural sequence to sequence models are currently used for variety of tasks in Natural Language Processing including machine translation (Sutskever et al., 2014; Bahdanau et al., 2014), text summarization (Rush et al., 2015), natural language generation (Wen et al., 2015), and others. This was enabled by the capability of recurrent neural networks to model temporal structure in data, including the long-distance dependencies in case of gated networks (Hochreiter and Schmidhuber, 1997; Cho et al., 2014).

The deep learning models’ ability to learn a dense representation of the input in the form of a real-valued vector recently allowed researchers to combine machine vision and natural language processing into tasks believed to be extremely difficult only few years ago. The distributed representations of words, sentences and images can be understood as a kind of common data type for language and images within the models. This is then used in tasks like automatic image captioning (Vinyals et al., 2015; Xu et al., 2015), visual question answering (Antol et al., 2015) or in attempts to ground lexical semantics in vision (Kiela and Clark, 2015).

In this system description paper, we bring a summary of the Recurrent Neural Network (RNN)-based system we have submitted to the automatic post-editing task and to the multimodal translation task. Section 2 describes the architecture of the networks we have used. Section 3 summarizes related work on the task of automatic post-editing of machine translation output and describes our submission to the Workshop of Machine Translation (WMT) competition. In a similar fashion, Section 4 refers to the task of multimodal translation. Conclusions and ideas for further work are given in Section 5.

2 Model Description

We use the neural translation model with attention (Bahdanau et al., 2014) and extend it to include multiple encoders, see Figure 1 for an illustration. Each input sentence enters the system simultaneously in several representations $x_i$. An encoder used for the $i$-th representation $X_i = (x_1^i, \ldots, x_k^i)$ of $k$ words, each stored as a one-hot vector $x_j^i$, is a bidirectional RNN implementing a function

$$f(X_i) = H_i = (h_1^i, \ldots, h_k^i)$$

where the states $h_j$ are concatenations of the outputs of the forward and backward networks after processing the $j$-th token in the respective order.

The initial state of the decoder is computed as a weighted combination of the encoders’ final states.

The state of the decoder is computed as a weighted combination of the encoder’s final states.

The decoder is an RNN which receives an embedding of the previously produced word as an input in every time step together with the hidden state from the previous time step. The RNN’s output is then used to compute the attention and the next word distribution.

The attention is computed over each encoder separately as described by Bahdanau et al. (2014).

The attention vector $a_t^i$ of the $i$-th encoder in the
The probability of the decoder emitting the word $y_m$ in the $j$-th step, denoted as $P(y_m|H_1, \ldots, H_i, Y_{0..m-1})$, is proportional to

$$
\exp \left( W_o s_j^l + \sum_{i=1}^{n} W_a a_i^l \right)
$$

where $H_i$ are hidden states from the $i$-th encoder and $Y_{0..m-1}$ is the already decoded target sentence (represented as matrix, one-hot vector for each produced word). Matrices $W_o$ and $W_a$ are learned parameters; $W_o$ determines the recurrent dependence on the decoder’s state and $W_a$ determines the dependence on the (attention-weighted) encoders’ states.

For image captioning, we do not use the attention model because of its high computational demands and rely on the basic model by Vinyals et al. (2015) instead. We use Gated Recurrent Units (Cho et al., 2014) and apply the dropout of 0.5 on the inputs and the outputs of the recurrent layers (Zaremba et al., 2014) and L2 regularization of $10^{-8}$ on all parameters. The decoding is done using a beam search of width 10. Both the decoders and encoders have hidden states of 500 neurons, word embeddings have the dimension of 300. The model is optimized using the Adam optimizer (Kingma and Ba, 2014) with learning rate of $10^{-3}$.

We experimented with recently published improvements of neural sequence to sequence learning: scheduled sampling (Bengio et al., 2015), noisy activation function (Gülçehre et al., 2016), linguistic coverage model (Tu et al., 2016). None of them were able to improve the systems’ performance, so we do not include them in our submissions.

Since the target language for both the task was German, we also did language dependent pre- and post-processing of the text. For the training we split the contracted prepositions and articles ($am \leftrightarrow an \text{ dem}$, $war \leftrightarrow zu \text{ der}$, . . .) and separated some pronouns from their case ending (keinem $\leftrightarrow$ kein -em, unserer $\leftrightarrow$ unser -er, . . .). We also tried splitting compound nouns into smaller units, but on the relatively small data sets we have worked with, it did not bring any improvement.
3 Automatic Post-Editing

The task of automatic post-editing (APE) aims at improving the quality of a machine translation system treated as black box. The input of an APE system is a pair of sentences – the original input sentence in the source language and the translation generated by the machine translation (MT) system. This scheme allows to use any MT system without any prior knowledge of the system itself. The goal of this task is to perform automatic corrections on the translated sentences and generate a better translation (using the source sentence as an additional source of information).

For the APE task, the organizers provided tokenized data from the IT domain (Turchi et al., 2016). The training data consist of 12,000 triplets of the source sentence, its automatic translation and a reference sentence. The reference sentences are manually post-edited automatic translations. Additional 1,000 sentences were provided for validation, and another 2,000 sentences for final evaluation. Throughout the paper, we report scores on the validation set; reference sentences for final evaluation were not released for obvious reasons.

The performance of the systems is measured using Translation Error Rate (Snover et al., 2006) from the manually post-edited sentences. We thus call the score HTER. This means that the goal of the task is more to simulate manual post-editing, rather than to reconstruct the original unknown reference sentence.

3.1 Related Work

In the previous year’s competition (Bojar et al., 2015), most of the systems were based on the phrase-base statistical machine translation (SMT) in a monolingual setting (Simard et al., 2007). There were also several rule-based post-editing systems benefiting from the fact that errors introduced by statistical and rule-based systems are of a different type (Rosa, 2014; Mohaghegh et al., 2013).

Although the use of neural sequential model is very straightforward in this case, to the best of our knowledge, there have not been experiments with RNNs for this task.

3.2 Experiments & Results

The input sentence is fed to our system in a form of multiple input sequences without explicitly telling which sentence is the source one and which one is the MT output. It is up to the network to discover their best use when producing the (single) target sequence. The initial experiments showed that the network struggles to learn that one of the source sequences is almost correct (even if it shares the vocabulary and word embeddings with the expected target sequence). Instead, the network seemed to learn to paraphrase the input.

To make the network focus more on editing of the source sentence instead of preserving the meaning of the sentences, we represented the target sentence as a minimum-length sequence of edit operations needed to turn the machine-translated sentence into the reference post-edit. We extended the vocabulary by two special tokens keep and delete and then encoded the reference as a sequence of keep, delete and insert operations with the insert operation defined by the placing the word itself. See Figure 2 for an example.

After applying the generated edit operations on the machine-translated sentences in the test phase, we perform a few rule-based orthographic fixes for punctuation. The performance of the system is given in Table 1. The system was able to slightly improve upon the baseline (keeping the translation as it is) in both the HTER and BLEU score. The system was able to deal very well with the frequent error of keeping a word from the source in the translated sentence. Although neural sequential models usually learn the basic output structure very quickly, in this case it made a lot of errors in pairing parentheses correctly. We ascribe this to the edit-operation notation which obfuscated the basic orthographic patterns in the target sentences.

<table>
<thead>
<tr>
<th>method</th>
<th>HTER</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>.2481</td>
<td>62.29</td>
</tr>
<tr>
<td>edit operations</td>
<td>.2438</td>
<td>62.70</td>
</tr>
<tr>
<td>edit operations+</td>
<td>.2436</td>
<td>62.62</td>
</tr>
</tbody>
</table>

Table 1: Results of experiments on the APE task on the validation data. The "+" sign indicates the additional regular-expression rules – the system that has been submitted.

4 Multimodal Translation

The goal of the multimodal translation task is to generate an image caption in a target language (German) given the image itself and one or more captions in the source language (English).
Recent experiments of Elliott et al. (2015) showed that including the information from the images can help disambiguate the source-language captions.

The participants were provided with the Multi30k dataset (Elliott et al., 2016) which is an extension of the Flickr30k dataset (Plummer et al., 2015). In the original dataset, 31,014 images were taken from the users collections on the image hosting service Flickr. Each of the images were given five independent crowd-sourced captions in English. For the Multi30k dataset, one of the English captions for each image was translated into German and five other independent German captions were provided. The data are split into a training set of 29,000 images, a validation set of 1,014 images and a test set with 1,000 images.

The two ways in which the image annotation were collected also lead to two sub-tasks. The first one is called Multimodal Translation and its goal is to generate a translation of an image caption to the target language given the caption in source language and the image itself. The second task is the Cross-Lingual Image Captioning. In this setting, the system is provided five captions in the source language and it should generate one caption in target language given both source-language captions and the image itself. Both tasks are evaluated using the BLEU (Papineni et al., 2002) score and METEOR score (Denkowski and Lavie, 2011). The translation task is evaluated against a single reference sentence which is the direct human translation of the source sentence. The cross-lingual captioning task is evaluated against the five reference captions in the target language created independently of the source captions.

4.1 Related Work

The state-of-the-art image caption generators use a remarkable property of the Convolutional Neural Network (CNN) models originally designed for ImageNet classification to capture the semantic features of the images. Although the images in ImageNet (Deng et al., 2009; Russakovsky et al., 2015) always contain a single object to classify, the networks manage to learn a representation that is usable in many other cases including image captioning which usually concerns multiple objects in the image and also needs to describe complex actions and spacial and temporal relations within the image.

Prior to CNN models, image classification used to be based on finding some visual primitives in the image and transcribing automatically estimated relations between the primitives. Soon after Kiros et al. (2014) showed that the CNN features could be used in a neural language model, Vinyals et al. (2015) developed a model that used an RNN decoder known from neural MT for generating captions from the image features instead of the vector encoding the source sentence. Xu et al. (2015) later even improved the model by adapting the soft alignment model (Bahdanau et al., 2014) nowadays known as the attention model. Since then, these models have become a benchmark for works trying to improve neural sequence to sequence models (Bengio et al., 2015; Guçlüşehre et al., 2016; Ranzato et al., 2015).

4.2 Phrase-Based System

For the translation task, we trained Moses SMT (Koehn et al., 2007) with additional language models based on coarse bitoken classes. We follow the approach of Stewart et al. (2014): Based on the word alignment, each target word
Table 2: Results of experiments with the multimodal translation task on the validation data. At the time of the submission, the models were not tuned as well as our final models. The first six systems are targeted for the translation task. They were trained against one reference – a German translation of one English caption. The last four systems are target to the cross-lingual captioning task. They were trained with 5 independent German captions (5 times bigger data).

<table>
<thead>
<tr>
<th>System</th>
<th>Multimodal translation BLEU</th>
<th>Multimodal translation METEOR</th>
<th>Cross-lingual captioning BLEU</th>
<th>Cross-lingual captioning METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses baseline</td>
<td>32.2</td>
<td>54.4</td>
<td>11.3</td>
<td>33.8</td>
</tr>
<tr>
<td>MM baseline</td>
<td>27.2</td>
<td>32.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tuned Moses</td>
<td>36.8</td>
<td>57.4</td>
<td>12.3</td>
<td>35.0</td>
</tr>
<tr>
<td>NMT</td>
<td>37.1</td>
<td>54.6</td>
<td>13.6</td>
<td>34.6</td>
</tr>
<tr>
<td>NMT + Moses</td>
<td>36.5</td>
<td>54.3</td>
<td>13.7</td>
<td>35.1</td>
</tr>
<tr>
<td>NMT + image</td>
<td>34.0</td>
<td>51.6</td>
<td>13.3</td>
<td>34.4</td>
</tr>
<tr>
<td>NMT + Moses + image</td>
<td>37.3</td>
<td>55.2</td>
<td>13.6</td>
<td>34.9</td>
</tr>
<tr>
<td>-- &quot; --, submitted</td>
<td>31.9</td>
<td>49.6</td>
<td>13.0</td>
<td>33.5</td>
</tr>
<tr>
<td>captioning only</td>
<td></td>
<td>9.1</td>
<td></td>
<td>25.3</td>
</tr>
<tr>
<td>5 en captions</td>
<td></td>
<td>22.7</td>
<td></td>
<td>38.5</td>
</tr>
<tr>
<td>5 en captions + image</td>
<td></td>
<td>24.6</td>
<td></td>
<td>39.3</td>
</tr>
<tr>
<td>-- &quot; --, submitted</td>
<td></td>
<td>14.0</td>
<td></td>
<td>31.6</td>
</tr>
</tbody>
</table>

Manual inspection of these three best configurations reveals almost no differences; often the outputs are identical. Comparing to the baseline (a single word-based LM), it is evident that coarse models prefer to ensure agreement and are much more likely to allow for a different word or preposition choice to satisfy the agreement.

4.3 Neural System

For the multimodal translation task, we combine the RNN encoders with image features. The image features are extracted from the 4096-dimensional penultimate layer (fc7) of the VGG-16 Imagenet network Simonyan and Zisserman (2014) before applying non-linearity. We keep the weights of the convolutional network fixed during the training. We do not use attention over the image features, so the image information is fed to the network only via the initial state.

We also try a system combination and add an encoder for the phrase-based output. The SMT encoder shares the vocabulary and word embeddings with the decoder. For the combination with SMT output, we experimented with the CopyNet architecture (Gu et al., 2016) and with encoding the sequence the way as in the APE task (see Section 3.2). Since neither of these variations seems to have any effect on the performance, we report only the results of the simple encoder combina-
Source: A group of men are loading cotton onto a truck.
Moses: Eine Gruppe von Männern ladt cotton auf einen Lkw.
MMMT: Eine Gruppe von Männern ladt etwas auf einem Lkw.
Gloss: A group of men are loading something onto a truck.
CLC: Mehrere Personen stehen an einem LKW.

Source: A man sleeping in a green room on a couch.
Reference: Ein Mann schlaft in einem grünem Raum auf einem Sofa.
Moses: Ein Mann schlaft in einem grünem Raum auf einem Sofa.
MMMT: Ein Mann schlaft in einem grünem Raum auf einer Couch.
CLC: Eine Frau schlaft auf einer Couch.

Figure 3: Sample outputs of our multimodal translation (MMMT) system and cross-lingual captioning (CLC) system in comparison with phrase-based MT and the reference. The MMMT system refers to the ‘NMT + Moses + image’ row and CLC system to the ‘5 captions + image’ row in Table 2.

4.4 Results

The results of both the tasks are given in Table 2. Our system significantly improved since the competition submission, therefore we report both the performance of the current system and of the submitted systems. Examples of the system output can be found in Figure 3.

The best performance has been achieved by the neural system that combined all available input both for the multimodal translation and cross-lingual captioning. Although, using the image as the only source of information led to poor results, adding the image information helped to improve the performance in both tasks. This supports the hypothesis that for the translation of an image caption, knowing the image can add substantial piece of information.

The system for cross-lingual captioning tended to generate very short descriptions, which were usually true statements about the images, but the sentences were often too general or missing important information. We also needed to truncate the vocabulary which brought out-of-vocabulary tokens to the system output. Unlike the translation task where the vocabulary size was around 20,000 different forms for both languages, having 5 source and 5 reference sentences increased the vocabulary size more than twice.

Similarly to the automatic postediting task, we were not able to come up with a setting where the combination with the phrase-based system would improve over the very strong Moses system with bitoken-classes language model. We can therefore hypothesize that the weakest point of the models is the weighted combination of the inputs for the initial state of the decoder. The difficulty of learning relatively big combination weighting matrices which are used just once during the model execution (unlike the recurrent connections having approximately the same number of parameters) probably over-weighted the benefits of having more information on the input. In case of system combination, more careful exploration of explicit copy mechanism as CopyNet (Gu et al., 2016) may be useful.
5 Conclusion

We applied state-of-the-art neural machine translation models to two WMT shared tasks. We showed that neural sequential models could be successfully applied to the APE task. We also showed that information from the image can significantly help while producing a translation of an image caption. Still, with the limited amount of data provided, the neural system performed comparably to a very well-tuned SMT system.

There is still a big room for improvement of the performance using model ensembles or recently introduced techniques for neural sequence to sequence learning. An extensive hyper-parameter testing could be also helpful.

Acknowledgment

We would like to thank Tomáš Musil, Milan Straka and Ondřej Dušek for discussing the problem with us and countless tips they gave us during our work.

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 645 452 (QT21) and no. 644 753 (KConect) and the Czech Science Foundation (grant n. P103/12/G084). Computational resources were provided by the CESNET LM2015042, provided under the programme “Projects of Large Research, Development, and Innovations Infrastructures.”

References


Sepp Hochreiter and Jürgen Schmidhuber. 1997.
D1.1: Semantic Translation Models


Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neu-


