Aljoscha Burchardt (DFKI), Ondřej Bojar (CUNI), Yvette Graham (DCU), Kim Harris (text&form), Qun Liu (DCU), Qingsong Ma (DCU), Katrin Marheinecke (text&form), Lucia Specia (USFD), Frédéric Blain (USFD), Inguna Skadiņa (Tilde), Mārcis Pinnis (Tilde), Marco Turchi (FBK), Vivien Macketanz (DFKI), JanThorsten Peter (RWTH), Dušan Variš (CUNI), Philip Williams (UEDIN)

Dissemination Level: Public

Final, 31st January, 2018
<table>
<thead>
<tr>
<th><strong>Grant agreement no.</strong></th>
<th>645452</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project acronym</strong></td>
<td>QT21</td>
</tr>
<tr>
<td><strong>Project full title</strong></td>
<td>Quality Translation 21</td>
</tr>
<tr>
<td><strong>Type of action</strong></td>
<td>Research and Innovation Action</td>
</tr>
<tr>
<td><strong>Coordinator</strong></td>
<td>Prof. Josef van Genabith (DFKI)</td>
</tr>
<tr>
<td><strong>Start date, duration</strong></td>
<td>1st February, 2015, 36 months</td>
</tr>
<tr>
<td><strong>Dissemination level</strong></td>
<td>Public</td>
</tr>
<tr>
<td><strong>Contractual date of delivery</strong></td>
<td>31st January, 2018</td>
</tr>
<tr>
<td><strong>Actual date of delivery</strong></td>
<td>31st January, 2018</td>
</tr>
<tr>
<td><strong>Deliverable number</strong></td>
<td>D3.5</td>
</tr>
<tr>
<td><strong>Deliverable title</strong></td>
<td>Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Report</td>
</tr>
<tr>
<td><strong>Status and version</strong></td>
<td>Final</td>
</tr>
<tr>
<td><strong>Number of pages</strong></td>
<td>200</td>
</tr>
<tr>
<td><strong>Contributing partners</strong></td>
<td>UEDIN, DFKI, RWTH, DCU, KIT, FBK, USFD, text&amp;form, TILDE</td>
</tr>
<tr>
<td><strong>WP leader</strong></td>
<td>FBK</td>
</tr>
</tbody>
</table>

**Author(s)**
Aljoscha Burchardt (DFKI), Ondřej Bojar (CUNI), Yvette Graham (DCU), Kim Harris (text&form), Qun Liu (DCU), Qingsong Ma (DCU), Katrin Marheinecke (text&form), Lucia Specia (USFD), Frédéric Blain (USFD), Inguna Skadiņa (Tilde), Mārcis Pinnis (Tilde), Marco Turchi (FBK), Vivien Macketanz (DFKI), JanThorsten Peter (RWTH), Dušan Varioš (CUNI), Philip Williams (UEDIN)

**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
stephan.busemann@dfki.de

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

© 2018, The Individual Authors

No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the copyright owner.
Quality Translation 21

D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

Contents

1 Preliminary remark 4
2 Source-driven evaluation 5
3 Target diagnostics and Error Profiles 6
4 Reference-based evaluation 9
5 Reference-free evaluation 10
6 Re-assessing QT21 MT systems at WMT17 12

References 15

A Machine Translation: Phrase-Based, Rule-Based and Neural Approaches with Linguistic Evaluation 17
B A Ling. Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines 33
C Can Out-of-the-box NMT Beat a Domain-trained Moses on Technical Data? 45
D Can Out-of-the-box NMT Beat a Domain-trained Moses on Technical Data? 51
E TQ-AutoTest – An Automated Test Suite for (Machine) Translation Quality 57
F QT21 data and the TAUS Quality Dashboard 61
G Feature-Rich NMT and SMT Post-Edited Corpora for Productivity and Evaluation Tasks with a Subset of MQM-Annotated Data 67
H MaxSD: A Neural Machine Translation Evaluation Metric Optimized by Maximizing Similarity Distance 81
I Blend: a Novel Combined MT Metric Based on Direct Assessment — CASICT-DCU submission to WMT17 Metrics Task 90
J Can machine translation systems be evaluated by the crowd alone 96
K CUNI Experiments for WMT17 Metrics Task 124
L Further Investigation into Reference Bias in Monolingual Evaluation of Machine Translation 132
M Translation Quality and Productivity: A Study on Rich Morphology Lang. 142
N Bilexical Embeddings for Quality Estimation 159
O Feature-Enriched Character-Level Convolutions for Text Regression 165
P One-parameter models for sentence-level post-editing effort estimation 172
Q Fine-grained word-level machine translation quality estimation 184
R Neural Quality Estimation 189
S Improving Evaluation of Document-level MT Quality Estimation 195
1 Preliminary remark

This Deliverable reports on a number of different activities in WP3 (mostly Task 3.2) that revolve around utilizing both human annotations and automatic assessment of translations to evaluate translations and to further the state of the art in this important area. As explained previously, the notion of "second annotation round" might be confusing as it dates back to an early idea to develop the WP1/2 engines in two cycles during the project, with evaluation of the system output by WP3 (post-editing, annotation, etc.) after each cycle. Due to the shift to neural MT and the different preferences of training data (domain vs. news), however, the plan was modified early in the project to allow for more freedom in the research directions. As foreseen in the Periodic Report D6.1, we have extended the structure of D3.3 to cover all relevant topics. This document is thus structured along the following main aspects of MT diagnosis and evaluation researched in QT21:

- Source-driven evaluation (Section 2)
- Target diagnostics and Error Profiles (Section 3)
- Reference-based evaluation (Section 4)
- Reference-free evaluation/prediction (Section 5)

In Section 6, we present an additional study in which we have applied our main evaluation method on WMT-2017 systems including systems from QT21.

Again, we cite the relevant publications in the Appendix and refer to them in the various sections. As this is the last Deliverable, we also include ongoing work that has not yet been published.

After the QT21 general meeting in Berlin, advisory board member Jörg Porsiel, head of translation at Volkswagen, asked WP3 partners to contribute to the book "Machine Translation" he was editing for the BDÜ, the professional association of interpreters and translators in Germany. The two contributions ([1], [2]) focus on quality management and bridging the gap between scientific insights and the needs of industry regarding MT.
2 Source-driven evaluation

First experiments using the test suite approach developed by DFKI in QT21 have been performed in cooperation with the QTLearn project on a test suite built specifically for the IT domain. Part of this work had been summarized already in D3.3. It has been published in an extended form in [3]. Interestingly, the different system types (RBMT, PBMT and NMT) tested show a similar overall performance in terms of the errors analysed, getting about 75% of the phenomena right. However, the distribution of errors is very different. The admittedly early neural system used in QTLearn had, e.g., difficulties with compounds and phrasal verbs (not as severe as the SMT systems, though) while the RBMT system had most difficulties with terminology. The full paper can be found in Appendix A.

An early version of the test suite built in QT21 (ca. 800 sentences per language direction EN-DE as described in D3.3) has been tested on engines from WP 1 and WP 2 in a larger experiment. This version of the test suite was used as we had results of Google Translate before the shift to NMT only for this version. This way, we could compare the project’s systems with both instances of Google. We chose a mix of different systems types and training material to get a broad range of insights and feedback. Some of the work had been sketched in D3.3 already. This test suite version contained a relatively large amount of verbal paradigms which are handled good by RBMT systems. But the experiments have shown that the best NMT systems are performing also very well this data. The work has been reported in [4] (see Appendix B).

The work on NMT performed by QT21 has lead to questions from industry as to whether it is the right time (mid 2017) to make a switch to NMT. To provide evidence, we have designed another experiment with a custom-built test suite and a system from WP 1 that is tested against a domain-optimised Moses system that was in use in production at an LSP. The experiments have shown that even the unadapted NMT system could beat the SMT system on most error categories that we have assessed. The only exceptions were terminology and tags, but this was expected as the SMT system had been optimised for both while the NMT system was trained on general domain data. The experiment is reported in [5] (see Appendix D). It was also presented orally at the tekom fair in Germany by Anne Beyer and Aljoscha Burchardt in November 2017 (see Appendix E).

Our test suite in its current, fully extended version comprises about 5000 test items for the language pair German-English in both directions. All experiments reported above were based on manual evaluations. In the last project year, we have focussed our efforts on the automation of the experimenting and testing workflow. The envisaged approach using regular expressions has shown to perform beyond our initial expectations. By the time of writing, half of the test suite has already been automated (DE-EN), and the other half is work in progress. The dashboard interface is ready and in use. The current state of the test suite and its automation is reported in Appendix E, which has been accepted for presentation at LREC 2018. The test suite has been demonstrated at the META-FORUM and has caught some interest, mostly from industry. QT21 advisory board member Jörg Porsiel from Volkswagen had invited DFKI to present the test suite at VW and it is planned to cooperate in a project in the future.
3 Target diagnostics and Error Profiles

3.1 DQF-MQM standardization process and adoption in the translation industry

The DQF-MQM standard for error typology has quickly been adopted in the translation industry (dissemination activities are summarised in Deliverable D5.4). The DQF-MQM standardization process has also made very good progress. It has become a working item at ASTM that is under constant development by a group of stakeholders representing research, industry, DGT, and others. In order to maintain and host the the metrics definition beyond the QT21 project, a W3C community group has been formed. This metrics definition will be the publicly available counterpart to the ASTM standard document, which will need to be purchased. It will include practices, benchmarks, cost calculation schemes, etc. relevant for industrial users of DQF-MQM.

DQF-MQM is also part of the TAUS DQF framework. DQF is a comprehensive set of tools, best practices, metrics, reports and data to help industry set benchmarking standards for translation quality. DQF-MQM is the part of DQF that is concerned with error analysis and annotation. DQF-MQM has been adopted both as a part of the DQF framework and in standalone use. Examples of the latter category are the current Smartling functionality to evaluate translations with DQF-MQM error categories, and the use of spreadsheet templates of which the TAUS Error typology template is just one (https://info.taus.net/dqf-mqm-error-typology-templ). The use of DQF-MQM through the DQF Framework is rapidly growing. First there is the growing number of CAT tools that have developed or are developing plugins for the DQF Framework. These CAT tools include Trados Studio, WorldServer, GlobalLink, SDL TMS, XTM, Kaleidoscope and MateCat. TAUS DQF is adopted by several companies who are using the DQF framework on a daily basis. In just a few months the DQF database has grown to over 100 million words thanks to contributors such as Dell EMC, Lionbridge, LDS Church and Tableau Software. Growth is currently accelerating. Current users are populating the database primarily with translation data. This will change as there is much interest and involvement from other parties who will be using the DQF Framework primarily to review translations. DQF-MQM provides a uniform way of approaching quality that is quantifiable and comparable among all parties in the translation industry, and therefore is embraced as one of the main paradigms for translation quality evaluation. In Appendix F we compare the QT21 approach to error annotation with the DQF standard process for reviewing and reporting, which makes use of the TAUS Dashboard.

3.2 Error profiles

The vast amount of industry data collected in QT21, the use and application of MQM, the commitment to determine the most useful input and feedback for both academic and applied research in the most congenial and efficient environment for translators has produced an unprecedented amount of data containing key indicators of MT quality, both in terms of original output as well as post-editing effort and annotation insight.

The data consisted of 20,000 - 45,000 sentences for four language pairs (EN-DE, DE-EN, EN-LV and EN-CS) and included a complete set of validated reference translations by professional translators. While the scope of the project did not allow us to perform any in-depth comparative quality analysis on the linguistic differences between the original MT output and the reference translations, initial observations indicate that industry-specific (and often mandatory) stylistic and valency discrepancies between the two are quite significant. First results have been presented at MT Summit (see Appendix G).

The primary features recorded for error profile analysis during post-editing include keystroke logging, time logging, as well as quality scores for every sentence. In addition, two identical sets of data were run through both a PBMT and an NMT system for two language pairs, and
Table 1: MQM error categories and breakdown of annotations completed to data.

MQM annotation was performed on 2000 sentences per language pair. The idea behind the collection of these key data points was to determine whether there were correlations between time/effort, perceived quality and actual quality, and to study these correlations with respect to actual errors and error patterns and other characteristics such as sentence length in the two systems.

A first analysis of the annotated data showed definite patterns in the error types annotated, with distinct patterns for PBMT and NMT output respectively (see Table 1 from [6]). For EN-DE, for example, we observed that

- the number of errors in PBMT output is almost twice as high as in NMT,
- function words represent largest error category by far in both systems,
- agreement and word order were the primary error categories for PBMT,
- mistranslation and omissions were the primary error categories for NMT,
- the proportion of mistranslations is twice as high in NMT as in SMT (PBMT).

For EN-LV we observed that (see [7])

- NMT translations are more fluent than SMT translations, especially word order errors are twice as less as in SMT outputs,
- errors related to accuracy, especially, mistranslation and omission errors, occur more often in NMT outputs,
- the word form errors, that characterize the morphological richness of Latvian, are frequent for both systems, but slightly fewer in NMT outputs.

### 3.2.1 Initial observations

Currently lacking the analytical tools to perform empirical analyses on the data points collected, we interviewed the translators who performed the EN-DE post-editing and annotation on all
of the data. Initial results indicate that there is a strong correlation between the error profiles and required effort to post-edit, and that cognitive effort plays a much more significant role in effort than previously thought. In addition, the number of errors contained in a PBMT sentence appears to influence the perceived quality of that sentence even if the amount of time required to edit is equal to that required to edit the same NMT with fewer but more cognitively challenging sentences. An example is provided below:

**Source** A *dim* check mark indicates that the condition is applied only to part of the selection.

**NMT** Ein Häkchen bedeutet, dass die Bedingung nur auf einen Teil der Auswahl angewendet wird. (Omission)

**PBMT** Eine unscharfe Häkchen gibt an, dass die Bedingung nur auf einen Teil der Auswahl angewendet wird. (Agreement, Mistranslation).

Classifying errors and error types and performing initial rough analyses on the logged data and translator interviews have shown that:

- SMT errors are repetitive, show patterns.
- NMT errors are less predictable.
- Cognitive effort is therefore higher for post-editing NMT than it is for SMT.
- As a result, current NMT output does not appear to significantly beat PBMT in PE productivity.
- Longer sentences in NMT contain less obvious errors because these systems handle distance better than SMT. The risk of errors remaining undetected is therefore greater.
- Sentence length can greatly influence productivity and quality regardless of system. One sentence that is three times as long as another took five times as long to post-edit and contained more errors than three shorter sentences.

### 3.2.2 Ongoing analyses

Three of the partners are currently developing a tool outside of the project that will allow us to filter and combine results in order to empirically assess and evaluate the results with a focus on the following:

- combine data points to demonstrate various hypothesis within and against each system, for example, cognitive effort (time, keystrokes, sentence length) and perceived effort (time, keystrokes, quality score);
- benchmark automatic scores such as BLEU, METEOR, TER against quality scores, keystrokes and time in any combination;
- demonstrate improvement approaches such as the automatic pre-editing of the source based on criteria such as error types and sentence length;
- improve automatic post-editing of the target;
- benchmark the results of proprietary user Quality Estimation (QE) tools against this data set to evaluate accuracy using various data points.

The results will be published later.
4 Reference-based evaluation

DCU proposed a novel metric for machine translation evaluation based on neural networks. In the training phase, we maximize the distance between the similarity scores of high and low-quality hypotheses (available in Appendix H). Then, the trained neural network is used to evaluate the new hypotheses in the testing phase. The proposed metric could efficiently incorporate lexical and syntactic metrics as features in the network and thus is able to capture different levels of linguistic information. Experiments showed state-of-the-art performance is achieved across several language pairs at system level and at segment level.

Furthermore, DCU developed a new metric, Blend (available in Appendix I). Blend includes syntactic similarity measures between MT output and reference translation based on shallow parsing, dependency parsing and constituent parsing. In terms of semantic similarity, Blend also incorporates named entities and semantic role labels. Since Blend incorporates a large number of different metrics, an important component of the metric is the method of combining this large range of metric scores into a single numeric score that will correlate well with human assessment for unseen translations. Experiments were carried out to investigate the best human evaluation to employ for the purpose of training Blend in this respect. The first method of human evaluation explored was Relative Ranking (available in Appendix J), where two or more translations are compared with one another and a human evaluator annotates their rank order in terms of translation quality. Secondly, Direct Assessment (DA) (available in Appendix J) was investigated as a form of human assessment to combine the metrics that comprise Blend. Results showed that when DA human evaluation is used to guide the training process this results in a vast reduction (from 445,000 sentence rankings to 4,800 sentence DA scores) in required training data for combining the syntactic and semantic features that make up Blend, while still achieving improved performance (from 0.633 to 0.641 in terms of pearson coefficient) when evaluated on WMT16 to-English language pairs.

In DCU and CUNI investigate the accuracy of human evaluation metrics by how well they correlate with human judgment for all systems participating in WMT17 News translation task. CUNI also submitted three evaluation metrics to the task (available in Appendix K). Participating translation systems included a range of PBMT, NMT, hybrid and rule-based systems from both academic institutions as well as industry. As expected metrics’ correlation with human assessment was similar to previous year’s metric shared task results, indicating that the accuracy of metrics has not changed in any substantial way since the introduction of NMT systems. Metrics therefore maintain the existing problems as compared to evaluation of PBMT systems, they do not provide a perfect substitute for human assessment and are more reliable on the system level than the sentence level. (See also Deliverable 4.3 on further details on the metrics task.)

In addition, DCU investigated prior conclusions about reference bias in monolingual human evaluation of MT. Monolingual evaluation of MT aims to simplify human assessment by requiring assessors to compare the meaning of the MT output with a reference translation, opening up the task to a much larger pool of genuinely qualified evaluators. Monolingual evaluation runs the risk, however, of bias in favour of MT systems that happen to produce translations superficially similar to the reference and, consistent with this intuition, previous QT21 research, published in (available in Appendix L), concluded monolingual assessment to be strongly biased in this respect. In (available in Appendix L), DCU carried out further investigation into reference bias via direct human assessment of MT adequacy via quality controlled crowd-sourcing. Contrary to both intuition and past conclusions, our results for this particular experiment showed no significant evidence of reference bias in the monolingual evaluation of MT.
5 Reference-free evaluation

Based on the post-edited and MQM machine translation data collected in WP3 [4], USFD proposed novel methods for word, phrase and sentence-level Quality Estimation (QE):

- A new approach to word-level quality estimation, with systems submitted to the WMT17 word and phrase-level quality estimation tasks. The approach exploits bi-lexical embeddings to eliminate the need for engineered features. It does so by learning bi-lexical operators over distributional representations of words in source-target text pairs to predict whether information encoded in the source sentence is preserved in the target sentence after translation. The experiments include using both labelled and unlabelled data and different context windows. Word-level predictions are then extrapolated for the phrase-level task, for example, by labelling an entire phrase as incorrect if at least one of its words is predicted to be incorrect [17] (see Appendix N).

- A new neural approach for sentence-level quality estimation focusing on combining engineered features and character-level information. The model uses deep parallel convolution stacks, multi-layer perceptrons and multi-task learning. It is divided in three main components: a pair of deep convolution layer stacks for the source and translated sentences which takes as input one-hot character-level representations of these sentences, a multi-layer perceptron for the engineered features, and a final multi-layer perceptron to combine all of this information. This is a general approach that can be used for many text regression tasks [18] (see Appendix O).

- An analysis of the performance of predictors of post-editing time at sentence-level based on simple, easy to interpret one-parameter models that explore general properties of the data: (a) a weighted average of measured post-editing times in a training set, where weights are an exponential function of edit distances between the new segment and those in the training data; (b) post-editing time as a linear function of the length of the segment; and (c) source and target statistical language models. These simple estimators outperform strong baselines, such as the official baseline of the quality estimation shared task (0.65 Pearson correlation with human scores versus 0.61 obtained by the official baseline in the WMT13 shared task on quality estimation), and are surprisingly competitive compared to more complex estimators, which have many more parameters and combine rich features (e.g. 0.65 Pearson correlation with human scores versus 0.68 obtained by the winning system in the WMT13 shared task on quality estimation). Simple linear combinations of estimators of types (b) and (c) do not seem to be able to improve the performance of the single best estimator, which suggests that more complex, non-linear models could indeed be beneficial when multiple indicators are used [19] (see Appendix P).

- An extensive study on the predictability of different word-level MQM errors using both neural and statistical machine translated data that has been post-edited and subsequently MQM-annotated. By mixing post-editing and error annotation, it was possible to isolate and study specific errors. Prediction models were built using a state-of-the-art neural method, which achieved the highest performance score in the word-level quality estimation shared task at WMT17 [13]. With these models we were able to investigate the nature of the different types of word-level errors in machine translation output and address the following questions: Are all of them equally difficult to predict? How do they interact? Is it better to identify them separately, i.e. taking other errors in the sentence in context? We then explored the best strategy to detect these errors using the state-of-the-art method. A number of experiments were performed to transition from employing generic models built on large amounts of training data to using small, error-specific models (see draft paper submission in Appendix Q).
• A new light-weight neural approach for sentence-level quality estimation that performs as well as the state-of-the-art method at a much lower training cost. In the last two years, neural models have been proposed for quality estimation that outperform other approaches by a large margin. However, those methods are rather costly: They still tend to rely on feature extraction, use complex architectures or, most importantly, require extensive pre-training. In addition, quality estimation models have been designed for outputs of statistical machine translation, despite the predominance of the rather more popular neural machine translation paradigm. We investigate state-of-the-art quality estimation methods on neural machine translation output and show that they still outperform others, but face new challenges. We then propose a low-cost neural quality estimation approach that yields results comparable to those of state-of-the-art methods relying on no feature engineering and on reduced datasets, without any pre-training (see draft paper submission in Appendix \[R\]).

When it comes to document-level QE, meaningful conclusions about the relative performance of NLP systems are only possible if the gold standard employed in a given evaluation is both valid and reliable. In \[20\] (available in Appendix \[S\]), DCU explored the validity of human annotations currently employed in the evaluation of document-level quality estimation for machine translation. The degree to which quality estimation system rankings are dependent on weights employed in the construction of the gold standard were demonstrated, before investigating direct human assessment as a valid alternative. Experiments showed direct assessment scores for documents to be highly reliable, achieving a correlation of above 0.9 in self-replication experiments.
6 Re-assessing QT21 MT systems at WMT17

In this section we analyse a different type of data: News domain texts used in the WMT17 shared task. The goal is to evaluate translations for this data produced by state-of-the-art systems officially submitted to the translation shared task from two perspectives with respect to: (i) the post-editing effort involved in fixing such translations using statistics from post-editing such as time and keystroke logging, and (ii) the specific errors made by the systems. This evaluation methodology is significantly different from the one used by WMT, i.e., the direct assessment (DA), where (non-professional) human annotators judge each system translation according to its adequacy based on a comparison with a reference translation only, i.e. monolingual evaluation. For our evaluation we follow exactly the same procedure and guidelines used for the rest of the data collected within WP3: Professional translators post-edit the translations and subsequently annotate the errors of a subset of the post-edited data whose original MT output was given an average quality score using MQM.

The analysed data is part of the official WMT17 test sets for three language pairs: EN-DE, EN-LV, EN-CS. For the post-editing effort analysis, a sample of 1,200 source sentences per language pair was selected, and translations from the top three MT systems submitted by QT21 partners were used. Not all WMT17 participating systems were used because the cost of post-editing would go beyond what was available for this exercise. The data for each language pair was split across three translators such that each translator would see and post-edit one machine translation for a given source sentence and post-edit an equal number of translations per MT system. The statistics collected were post-editing time, edit distance between the raw MT and final post-edited version (i.e. HTER), keystrokes (split across 10 types of keys), and perceived post-editing effort in \{1,4\}, where 1 indicates a perfect translation, and 4 a translation that is not worth editing.

In Table 2 we summarise the post-editing statistics for the data, and contrast it with the relative position of each system in the official WMT17 system rankings according to direct assessment. We refer the reader to [13] for more details on direct assessment. As evidenced, the post-editing and direct assessment statistics differ in most cases. Even though a direct comparison is not possible since the post-edited data is only a sample of the data used for direct assessment, the sample is taken from the same distribution, and therefore we can expect similar observations (on average) for different translations. Some interesting observations:

- For EN-LV, the two measurements (DA and post-editing) agree on the best system: The best system (Sys3) is ranked first according to DA and is also the system with the shortest average post-editing time, lowest average number of keys, best post-editing effort score and smallest HTER. However, while DA clusters the two other systems together, they seem to differ significantly when it comes to post-editing statistics, with Sys2 performing consistently worse according to all metrics (but better according to DA-related metrics).

- For EN-DE and EN-CS, while DA ranks all systems as the same, there seems to be a clear trend for best system. In EN-DE, Sys3 is the best according to all post-editing statistics as well as raw DA scores, with Sys1 coming second. In EN-CS, Sys3 is clearly the best according to post-editing statistics, but the worst according to raw DA.

For the MQM error annotation, a sample of 200 sentences with at least some editing done and with a perceived post-editing effort score of ’2’ (i.e., very good translations, which could be post-edited quickly) were selected for each language pair and MT system. In Table 3 we show the counts of different types of errors observed. In this case, even though all MQM annotated sentences were selected in a biased way, such as to have ’very good quality’, we can attempt to compare the total count of errors per system against the DA-based ranking/scores and post-editing statistics: a higher number of errors could indicate systems that make more mistakes per segment, even in segments considered to be of good quality. The counts are comparable across systems since the same number of sentences was selected per system (200). In addition, we can look at the error distributions for different MT systems to understand whether supposedly better
Table 2: Post-editing statistics (average across 1,200 sentences) versus official DA scores and rankings (computed for a superset of the data including more MT systems). The DA-based ranking shown here is computed from the standardized mean DA scores, with systems that are not significantly different to each other grouped in clusters. Raw DA scores (Ave %) as well as standardised DA scores ($z$ scores) – from which the DA-based ranking is derived – are also shown.

<table>
<thead>
<tr>
<th></th>
<th>EN-DE</th>
<th>EN-LV</th>
<th>EN-CS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sys1</td>
<td>Sys2</td>
<td>Sys3</td>
</tr>
<tr>
<td>DA Ave % scores</td>
<td>66.7</td>
<td>66.0</td>
<td>69.8</td>
</tr>
<tr>
<td>DA Ave $z$ scores</td>
<td>0.022</td>
<td>-0.003</td>
<td>0.139</td>
</tr>
<tr>
<td>DA-based ranking</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Post-editing time</td>
<td>58.08</td>
<td>61.40</td>
<td>49.66</td>
</tr>
<tr>
<td>Post-editing keystrokes</td>
<td>76.18</td>
<td>78.82</td>
<td>65.44</td>
</tr>
<tr>
<td>Post-editing effort score</td>
<td>2.39</td>
<td>2.46</td>
<td>2.26</td>
</tr>
<tr>
<td>Post-editing distance (HTER)</td>
<td>23.21</td>
<td>25.11</td>
<td>20.56</td>
</tr>
</tbody>
</table>

systems make more or less of certain types of errors, in sentences judged as having comparable quality in terms of post-editing effort.

As shown in Table 3 for EN-DE all three MT systems make approximately the same total number of errors (all categories). The best system according to post-editing statistics (Sys3) makes more mistranslation errors and fewer grammatical errors, which one could argue are harder to fix. For EN-LV, the best system according to DA and post-editing statistics (Sys3) makes slightly fewer errors overall, but again makes substantially higher number of mistranslation errors but much fewer word form errors (especially agreement and tense/aspect/mood). The worse system according to DA and post-editing statistics (Sys2) makes substantially more errors in most categories, especially spelling errors. Finally, for EN-CS the best system according post-editing statistics (Sys3) makes fewer errors overall, and also much fewer word form errors due to incorrect tense/aspect/mood. The worse system (Sys1) makes substantially more errors, particularly omission and tense/aspect/mood errors.

We think it is too early to draw final conclusions, but our results show clearly that different human-based measures and metrics assess different aspects of translation quality. It is important that one is aware of this when designing an evaluation setup for a given project or experiment.
### Table 3: Counts or errors per category in the 200 MQM-annotated translations by each of three MT systems (Sys1, Sys2, Sys3) per language pair.

<table>
<thead>
<tr>
<th></th>
<th>EN–DE</th>
<th>EN–LV</th>
<th>EN–CS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sys1</td>
<td>Sys2</td>
<td>Sys3</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Addition</td>
<td>5</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>Mistranslation</td>
<td>159</td>
<td>152</td>
<td>180</td>
</tr>
<tr>
<td>Omission</td>
<td>20</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>Untranslated</td>
<td>23</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Fluency</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Grammar</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Function words</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Extraneous</td>
<td>12</td>
<td>18</td>
<td>12</td>
</tr>
<tr>
<td>Incorrect</td>
<td>43</td>
<td>60</td>
<td>44</td>
</tr>
<tr>
<td>Missing</td>
<td>33</td>
<td>37</td>
<td>43</td>
</tr>
<tr>
<td>Word form</td>
<td>20</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>Part of speech</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Agreement</td>
<td>4</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Tense/aspect/mood</td>
<td>20</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>Word order</td>
<td>19</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>Spelling</td>
<td>5</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>Typography</td>
<td>13</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Unintelligible</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Terminology</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>All categories</td>
<td>385</td>
<td>379</td>
<td>372</td>
</tr>
</tbody>
</table>

D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles
References


A Machine Translation: Phrase-Based, Rule-Based and Neural Approaches with Linguistic Evaluation

Vivien Macketanz¹, Eleftherios Avramidis¹, Aljoscha Burchardt¹, Jindrich Helcl², Ankit Srivastava¹

¹German Research Center for Artificial Intelligence (DFKI), Language Technology Lab, Berlin, Germany
²Institute of Formal and Applied Linguistics, Charles University, Czech Republic

E-mails: vivien.macketanz@dfki.de  eleftherios.avramidis@dfki.de  aljoscha.burchardt@dfki.de  helcl@ufal.mff.cuni.cz  Ankit.Srivastava@dfki.de

Abstract: In this article we present a novel linguistically driven evaluation method and apply it to the main approaches of Machine Translation (Rule-based, Phrase-based, Neural) to gain insights into their strengths and weaknesses in much more detail than provided by current evaluation schemes. Translating between two languages requires substantial modelling of knowledge about the two languages, about translation, and about the world. Using English-German IT-domain translation as a case-study, we also enhance the Phrase-based system by exploiting parallel treebanks for syntax-aware phrase extraction and by interfacing with Linked Open Data (LOD) for extracting named entity translations in a post decoding framework.

Keywords: Machine translation, Parallel treebanks, Entity linking, Manual evaluation, Neural approaches.

1. Introduction to deep Machine translation and evaluation

With the recent appearance of neural approaches to Machine Translation (MT), we are dealing with three main MT paradigms: Rule-Based MT systems (RBMT), “classical” phrase-based Statistical MT systems (SMT) and Neural MT systems (NMT), the latest type of statistical systems. Translating between two languages requires substantial modelling of knowledge about the two languages, about translation, and about the world. Interestingly enough, little effort has been spent in the past on understanding what exactly MT systems learn, or to ask a simpler question – what aspects of language they can deal with and what remains challenging.

Unfortunately, today’s automatic measures for MT quality are not able to detect and model these aspects of translation in a detailed and analytical way. As one consequence, particular differences in the translations of different systems or system variants that may or may not constitute improvements remain undetected. Therefore,
we have argued for an evaluation approach that extends the current MT evaluation practice by steps where language experts inspect systems outputs [1]. We have started to use this extended evaluation approach in our contribution to the WMT2016 IT task [2] and presented it also at the Workshop on Deep Language Processing for Quality Machine Translation (DeepLP4QMT) in Varna, Bulgaria. In this contribution, we will provide a much extended description of our novel evaluation method driven by linguistic phenomena assembled in a test suite.

While test suites are a well-known tool that has often been used in Natural Language Processing (NLP), e.g., to test the performance of a parser, they are employed in MT only to a minor degree. One of the reasons might be that the complexity of languages makes it difficult to evaluate the MT output and draw conclusions from the findings. Nevertheless, in narrow domains there seems to be interest in detecting differences between systems and within the development of one system, e.g., in terms of pronouns [3] or verb-particle constructions [4]. A related fertile area of research is the series of shared tasks on cross-lingual pronoun prediction wherein similar to our linguistically-driven evaluation the discourse phenomenon (pronouns) is evaluated on competing MT systems using a “test suite” of lemmatised target-language human-authored translations [5].

In this paper we want to show to what extent a linguistically-driven evaluation may grant interesting insights into the nature of different MT systems and how these observations may help to improve the systems. In order to achieve this aim, we present a domain-specific as well as a domain-independent analysis.

Machine translation like other language processing tasks is confronted with the Zipf’ian distribution of relevant phenomena. Although surface-data-driven systems have enlarged the head considerably over the last years, the tail still remains a challenge. Many approaches have therefore tried to include various forms of linguistic knowledge in order to systematically address chunks of the tail [6].

One goal of this paper is to show how we can extend the classical phrase-based SMT systems in this direction. Adding to previous work [2], we will report more in-depth on “deeper”, more knowledge-driven ingredients of our work, namely (i) exploiting parallel treebanks for syntax-aware phrase extraction in SMT, and (ii) using Linked Open Data (LOD) for extracting named entity translations as a post-decoding module. Both parallel treebanks and LOD have been integrated in SMT systems previously. Syntactically annotated corpora have been used directly in syntax-based models [7-9] as well as indirectly as an augmentation to the non-linguistic phrase pairs [10-12]. In this paper, we follow the latter approach by extracting linguistically motivated phrase pairs from aligned and parsed corpora and appending them to the standard phrase-based SMT models. We extended the aforementioned works (primarily focusing on parliamentary proceedings) to new domains (IT-domain). There have also been several attempts to exploit linked data (resources stored on the web and connected via web links) into translating nouns and named entities in SMT systems [13, 14]. We implement a similar approach and enrich our phrase-based SMT system with translations from semantically linked knowledge bases.
2. Method

2.1. Baseline Machine translation systems

The extensions and evaluations we describe below start from three baseline systems:

The **phrase-based SMT** baseline is a domain-enhanced version of several state-of-the-art phrase-based systems, as indicated in the Shared task of Machine translation in WMT [15]. As the best system UEDIN-SYNTAX [16] included several components that were not openly available, we proceeded with adopting several settings from the next best system UEDIN [17], also given the fact that the difference of their ranking position is minimal (0.587 vs 0.614 BLEU score for English-German which was not statistically significant as a difference). The generic parallel training data (Europarl [18], News Commentary, MultiUN [19], Commoncrawl [20]) are augmented with domain-specific data from the IT domain (Libreoffice, Ubuntu, Chromium Browser [21]). The monolingual target side of the above corpora, along with the WMT News Corpus, is used for training one language model per corpus, whereas all of these intermediate language models are interpolated on in-domain data to form the final model used within the phrase-based decoding. In this paper, we describe two enhancements to the phrase-based SMT baseline, namely syntax-aware phrase extraction and linked-data-aware post-processing in Sections 3 and 4 respectively. In the examples we refer to this system as “SMT”.

The **rule-based** baseline is Lucy [22], a system that has shown state-of-the-art performance in many shared tasks. In this method, translation occurs in three phases, namely analysis, transfer, and generation. All three phases consist of hand-written linguistic rules that can capture the structural and semantic differences between German and other languages. Additionally, manual inspection has shown that it provides better handling of complex grammatical phenomena, such as long distance dependencies, particularly when translating into German.

Our **neural MT algorithm** represents the state of the art. It follows the description of [23]. The input sequence is processed using a bidirectional RNN encoder with Gated Recurrent Units (GRU) [24] into a sequence of hidden states. The final backward state of the encoder is then projected and used as the initial state of the decoder. Again, our decoder is composed of an RNN with GRU units. In each step, the decoder takes its hidden state and the attention vector (a weighted sum of the hidden states of the encoder, computed separately in each decoding step), and produces the next output word.

In addition to the attention model, we use byte pair encoding [25] in the preprocessing step. This ensures that there are no out-of-vocabulary words in the corpus and, at the same time, enables open-vocabulary decoding.

We trained our model on the same data as the phrase-based SMT baseline system and used the first 1000 segments of the QTLeepr corpus ([http://metashare.metanet4u.eu/go2/qtleapcorpus](http://metashare.metanet4u.eu/go2/qtleapcorpus)) for validation during training. In the experiments, the sentence length was limited to 50 tokens. The size of the hidden state of the encoder was 300 units, and the size of the hidden state of the decoder was 256 units. Both source and target word embedding vectors had 300
dimensions. For training, a batch size of 64 sentences was used. We used dropout and L2 for regularization.

Our model was implemented using Neural Monkey [26], a sequence to sequence learning toolkit built on top of the Tensorflow framework [27]. In the examples we refer to this system as “neural”.

2.2. Syntax-aware phrase extraction

Herein we define a linguistic enhancement to the phrase-based SMT baseline system described in Section 2. Under standard configuration such as in the baseline phrase-based SMT system, phrase pairs are extracted from parallel (sentence-aligned) corpora by obtaining word alignment in both directions and using heuristics such as the Grow-Diag-Final (GDF) algorithm [28].

The phrase pairs in the baseline system are not linguistically motivated which in turn leads to a number of errors in translation such as missing verbs. We extract linguistically motivated phrase pairs by obtaining phrase structure parse trees for both the source and target languages (on the same data as the baseline system) using monolingual constituency structure parsers such as the Berkeley Parser [29], and then aligning the subtrees using a statistical tree aligner [30]. These phrase pairs (illustrated with an example in Fig. 1) are then merged with the phrase pairs extracted in the baseline SMT system into one translation model. Thus we are merely using syntax to constrain the phrase boundaries and enabling SMT decoder to pick syntax-aware phrases, thereby ensuring noun phrases and verb phrases remain cohesive.

Through experimentation detailed in [31], we have discovered that non-linguistic phrase-based models (baseline phrase-based SMT) have a long tail (of coverage) and syntax-aware phrases underperform, if not concatenated with non-linguistic phrase pairs. We observed the syntax-aware system scored 0.8 BLEU points over the baseline system. Note that this system is referred to as the “SMT-syntax” system hereafter in the evaluations.

2.3. Named entity translation using linked data

In this section, we describe another enhancement to the baseline system: named entity translation. Named entities are terms (usually nouns like people names, places, organizations, locations or technical terms) which have a fixed (consistent) translation. SMT systems often translate them inconsistently or are unable to translate them on account of the named entity being absent in the models (unknown words).
One technique to address this deficiency is to integrate the SMT system with a Named Entity Recognition (NER) system, i.e., annotate all words and phrases in the source language which are identified as named entities. These named entities are then linked with a bilingual dictionary to retrieve translations in the target language which are then inserted into the translation in a post-decoding process.

For the dictionary in our experiments, we exploit multilingual terms semantically linked with each other in the form of freely available linguistic linked data on the web such as DBpedia (http://wiki.dbpedia.org) to identify named entities in our dataset in the same vein of [13]. These entities and their linked translations are then integrated with the translations of the baseline system such that the translations from DBpedia overwrite the baseline system translations. A step-by-step procedure for translating named entities in this manner is detailed in [32].

Note that although many unknown words are correctly identified and translated, DBpedia is a user-generated dictionary sourced from Wikipedia and is prone to contain errors or a different term altogether (“Microsoft Paint” versus “MS Paint”) which may reflect poorly in automatic evaluation metrics. This is another motivation to exercise the deep manual evaluation on enhancements to the baseline phrase-based SMT system. Hereafter “linked data” is used to refer to this MT system in the linguistic evaluations.

3. Manual linguistic evaluation

The evaluation of our systems is comprised of a deep manual analysis, performed by a professional German linguist (Following the general practice in industry, only one trained person, in this case the linguist, does the quality assurance). The goal of the manual evaluation was to validate the systems’ capabilities of specific linguistic phenomena. Apart from gaining insights into the nature of the errors, this method can also provide guidance for setting priorities for future extensions and improvements of the systems.

The manual evaluation has been performed on a variety of MT systems so far, the interested reader is referred to [2, 33].

3.1. Manual evaluation procedure

For the human-based analysis the following procedure was found to be a good practice: In a first step, the linguist browses through the outputs of the different systems and detects errors related to linguistic phenomena that seem prevalent and systematic. Additionally, we have consulted professional translators that provided us with a list of possible (machine) translation errors in the technical domain. With this approach we make sure that we do not miss any important linguistic categories that might lead to errors. To this end, we use the domain corpora of the WMT 2016 IT-translation task, namely the QTLeap corpus. The result of this first step is a short list of phenomena that require closer inspection.

Note that we understand “linguistic phenomenon” in a pragmatic sense covering a wide range of issues that impact translation quality. This can include not only...
common morpho-syntactic and semantic categories, but also formatting issues, issues of style, etc.

In order for this manual evaluation to be transferable to other contexts and domains as well, we are currently creating an expansive test suite (English <> German) that will be published elsewhere, containing a wide range of various linguistic phenomena (cf. Section 5.3). By selecting only the categories that are needed in a given context or domain, this test suite can serve as a basis for evaluation in various settings.

As it would be too time-consuming to perform a deep analysis of the complete corpus, 100 source segments that contain the respective linguistic phenomenon are randomly selected. Based on the source sentences, all the instances (by “instance” we refer to each occurrence of the phenomenon, e.g., each verb, term, etc.) of the respective phenomenon are counted in the 100 selected target segments of each system. Consequently, the occurrences of correctly translated phenomena in the system outputs are counted. The percentage of correctly translated phenomena is calculated by dividing the overall number of correctly translated instances by the overall number of instances in the source sentences. As one segment may consist of several sentences and contain several phenomenon instances, overall instance numbers can be greater than 100.

Additionally, certain key rules are followed in the evaluation process: First of all, the translation of the linguistic phenomena does not have to be equivalent to the reference translation, as there may be several correct translations. Furthermore, if a linguistic phenomenon is realized in a different structure that correctly translates the meaning, the output is counted as a correctly translated instance, cf. Example 1, in which the compound (F11 key) can either be translated as a compound like in the reference translation (F11-Taste) or as a (stylistically slightly dispreferred) noun modifier-construction like in the MT output (Taste F11) (The fact that the MT output produces the unnecessary verb “angezeigt” is being ignored here, as the focus always lies on only one phenomenon at a time).

**Example 1.**

Source:  *Try pressing the F11 key.*

Syntax:  *Drücken Sie die Taste F11 angezeigt.*

Reference:  *Betätigen Sie einfach die Taste F11.*

Note that the reference in this case introduces a spurious adverb (*einfach – simply*). This is one of several issues that we detected in the given corpus. It can be affiliated to the fact that – as has frequently been observed – human reference translations are sometimes not of perfect quality, depending on the circumstances of their creation.

The linguistic phenomena we identified as particularly prone to translation errors in the given corpus include imperatives, compounds, quotations marks, menu item separators (“>”), missing verbs, phrasal verb and terminology (as the segments were from the technical helpdesk domain). All these phenomena were analyzed separately, which means that the correctness of phenomena occurring within other phenomena (e.g., phrasal verbs within imperative constructions) is ignored when analyzing the latter.
The central idea of the test suite evaluation approach is to focus on certain phenomena and aspects of translation at a given time. For the test item, the criterion is not whether the whole phrase has been translated correctly. It depends on the test item what needs to be present in the translation to count as a correct treatment of the test item at hand.

Below, we will amplify how we treated the evaluation of the various phenomena in detail, as the different categories required individual regulations in addition to the universal key rules mentioned above.

Imperatives. Imperatives were classified as correct when the translation contained the proper imperative verb form and formal address pronoun (Sie). It is not required that the translation of the verb itself is correct, as the only important requirement was that the imperative construction is realized (correctly). As mentioned above, missing elements of phrasal verbs are irrelevant for the correctness of the construction.

The following example shows a segment that contains two imperatives (slide and Tap), of which only the second is translated correctly by the SMT system (Tippen Sie), whereas the first imperative is mistranslated as the noun Folie (because slide is a homonym that can mean both). The neural system leaves out the verbs completely. The reference translation contains the two correct translations (gleiten Sie and Tippen Sie).

Example 2.
Source: On the Home screen, quickly slide left to the Applications list. Tap Settings > About > More info.

SMT: Über die innere Bildschirm, schnell Folie links nach der Anwendungsliste auswählen. Tippen Sie Einstellungen > About > Mehr.


Compounds. A compound is counted as correctly translated when its components either form a German compound, which means that they have to be realized in one word, or when the components form a morphological construction that expresses the compound (e.g., adjective + noun, or genitive construction).

Example 3 contains three compounds (cartridge contact – Patronenkontakte, ink cartridges – Tintenpatronen and connectivity issues – Verbindungsprobleme). The RBMT system correctly translates all three compounds (Patronenkontakte, Tintenpatronen, Konnektivitätsfragen (the latter being a literal translation instead of the correct term, a fact which is being ignored in this case as the focus lies on the compound itself). The SMT-syntax (in the examples referred to as “syntax”) system correctly translates the first two compounds (Patronen-Kontakte (the variant with the hyphen is not common but not incorrect either) and Tintenpatronen), while it mistranslates the third compound (Anbindungsmöglichkeiten Fragen).
Example 3.
Source: Try to remove it and blow into the cartridge contacts as well as the printer’s. If they are recycled ink cartridges, there could be connectivity issues.

RBMT: Versuch, um es zu beseitigen und in die Patronenkontakte ebenso wie den Drucker zu blasen. Wenn sie wiederverwertete Tintenpatronen sind, könnte es Konnektivitätsfragen geben.

Syntax: Versuchen, um es zu entfernen und Schlag in die Patronenkontakte sowie die Drucker. Wenn sie wiederaufbereitet werden Tintenpatronen, es könnte Anbindungsmöglichkeiten Fragen.

Reference: Versuchen Sie, es zu entfernen und blasen in die Patronenkontakte als auch die des Druckers. Wenn es recycelte Tintenpatronen sind, könnte es Verbindungsprobleme sein.

Quotation marks. The quotation marks need to be placed around the right word in order to be counted as correct. They are not counted pairwise but separate as it may be the case that only one of a pair is placed correctly while the second one is missing or placed somewhere else. When there are more instances of quotation marks in the MT output than in the source, every redundant quotation mark is subtracted from the overall count of the respective segment, as it is the case in Example 4.

The source sentence in Example 4 comprises four quotation marks, but the RBMT system produces an output with five quotation marks. The first one of them is placed correctly (before (Advanced) – (Fortgeschritten)) while the second one is misplaced (after an instead of Privacy – Privatsphäre). Furthermore, the other two quotation marks from the reference around Clear browsing data are placed correctly around Klare Browsingdaten but the system added an additional quotation mark after Klare. Hence, even though the MT system achieves three correct instances, subtracting the redundant quotation mark results in two correct instances. The SMT system on the other hand places the right amount of quotation marks at the right positions.

Example 4.
Source: […] Touch “(Advanced) Privacy”. Select “Clear browsing data”.

RBMT: […] Fassen Sie „(Fortgeschritten) Privatsphäre an“. Auserlesene „Klare“ Browsingdaten“.

SMT: […] Touch „(Advanced) Datenschutz“. Wählen Sie „Browserdaten löschen“.

Reference: […] Berühren „(Erweitert) Datenschutz“. Wählen Sie „Browserdaten löschen“.

Menu item separators. The menu item separator “<>“ is counted in the same way as the quotation marks: The placement between two words needs to be correct in order for the menu item separator to be counted as correctly translated. Furthermore, the same rule concerning additional separators holds, meaning that those will be subtracted from the segment count.
Example 5 demonstrates the incorrect and correct translation of the menu item separator “>”: The source sentence contains two separators. Even though the RBMT system places the two separators in its output between the right words, it adds hyphens before and after the separators, converting the three words around the separators into one long compound. The linked data system represents the correct placement of the separators.

**Example 5.**

Source: *Go to Settings > General > Code Blocking.*

RBMT: *Gehen Sie zu Einstellungen > General > Code Blockierung.*

Linked data: *Gehen Sie zu Einstellungen > Allgemein > Code Blocking.*

Reference: *Gehen Sie auf Einstellungen > Allgemein > Codesperre.*

**Verbs.** For the translation of a verb in order to be counted as correct it is important that the verb is present in the MT output. The verb needs to be translated correctly or at least partly correctly as for example incomplete phrasal verbs are counted as correct. Every occurring verb form is counted separately. The conjugation does not need to be correct and verbs realized as nominalizations are also counted as correct. As has been said above, we allow ourselves a certain freedom what we call a linguistic phenomenon as our goal is not to create a linguistic theory, therefore verbs in fixed commands are not counted as they rather belong to terminology. Furthermore, it needs to be taken account of the fact that English progressive constructions (consisting of two verb forms) do not exist in German and are translated into a single verb which means that those constructions should be counted as one instance instead of two in the source sentence.

The source sentence in Example 6 contains four instances of verbs (*have*, *go*, *choose* and *are* *programming*) as the progressive construction *are* *programming* counts as one instance. The SMT system leaves out the verb *go* – *gehen* and mistranslates the progressive construction as verb + noun (*sind* *Programmierung*) which is a frequently occurring error. The RBMT system does not produce either of those errors as it correctly translates all four verbs (*müssen*, *geben*, *wählen* and *programmieren*). Note that the conjugation of the verb *wählen* is incorrect (cf. reference *auswählen*) but as mentioned above this translation counts as correctly translated (see below for the case of *wählen* vs. *auswählen*).

**Example 6.**

Source: *[...]* You have to *go* to the Language menu and there *choose* the language in which you *are* *programming*.

SMT: *[...]* Sie *haben*, um das Language Menü und *wählen* Sie die Sprache, in der Sie *sind* *Programmierung*.

RBMT: *[...]* Sie *müssen* zum Sprach-Menü *gehen* und es *wählt* die Sprache, in der Sie *programmieren*.

Reference: *[...]* Sie *müssen* ins Sprachen Menü *gehen* und die Sprache *auswählen*, in der Sie *programmieren*.

**Phrasal verbs.** German phrasal verbs have the characteristic that their prefixes move to the end of the sentence in certain constructions. The moved prefix is prone to getting lost in a machine translation or not moving to the end of the sentence but instead staying in its initial position. Therefore, only translations that contain the verb...
as well as its prefix (in the expected position) are counted as correct. Nevertheless, the evaluation of this phenomenon is not always easy as there are often cases where the English verb can be translated with a phrasal verb or a regular verb, which means that if a regular verb is present it needs to be counted as a correctly translated phrasal verb. Moreover, there are phrasal verbs that are acceptable with and without their suffix (e.g., *auswählen* vs. *wählen*). Hence, the analysis of the translation of the phrasal verbs needs to be treated with caution.

The verb *depend* in the source sentence in Example 7 translates into German as *abhängen*. In the given sentence, the prefix *ab* moves to the end of the sentence, as is it the case in the reference and the SMT-syntactic system. In the baseline SMT translation on the other hand the prefix stays in its initial position which is incorrect.

**Example 7.**

<table>
<thead>
<tr>
<th>Source</th>
<th>SMT</th>
<th>Syntax</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>It depends.</em> [...]</td>
<td><em>Es hängt davon ab.</em> [...]</td>
<td><em>Das hängt davon ab.</em> [...]</td>
<td><em>Es hängt davon ab.</em> [...]</td>
</tr>
</tbody>
</table>

**Terminology.** In order to be counted as correct, a translation of a term either needs to match the reference or the translation needs to be found in Microsoft’s Language Portal for Terminology ([https://www.microsoft.com/Language/en-US/Search.aspx](https://www.microsoft.com/Language/en-US/Search.aspx)). Commands consisting of more than one word (e.g., *Save as…*) are counted as one single term. Compounds on the other hand are counted as separate terms (e.g., *router page* is counted as two instances). Moreover, proper terms also belong to terminology. Case sensitivity needs to be taken into account.

In Example 8 the source sentence contains the three terms *desktop*, *right-click* and *icons* that should be translated into *Desktop*, *klicken Sie mit rechten Maustaste* and *Symbole* in German, as can be seen in the reference. While the SMT system correctly translates *desktop* – *Desktop* and *icons* – *Symbole*, it leaves out the verb and the pronoun (klicken Sie) in *right-click* – *klicken Sie mit der rechten Maustaste*, resulting in two correct instances. The linked data system correctly translated *icons* – *Symbole* and also leaves out the verb and pronoun in the second term. Additionally, it translates *desktop* as *Schreibtisch* – which is not an incorrect translation in general, but is incorrect in this technical domain.

**Example 8.**

<table>
<thead>
<tr>
<th>Source</th>
<th>SMT</th>
<th>Linked data</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>On the desktop, right-click in the area without icons</em> [...]</td>
<td><em>Auf dem Desktop, mit der rechten Maustaste auf dem Gebiet ohne Symbole</em> [...]</td>
<td><em>Auf dem Schreibtisch, mit der rechten Maustaste auf dem Gebiet ohne Symbole</em> [...]</td>
<td><em>Auf dem Desktop klicken Sie mit der rechten Maustaste in den Bereich ohne Symbole</em> [...]</td>
</tr>
</tbody>
</table>

3.2. Manual Evaluation results

For the seven linguistic categories depicted in the previous section, 657 source segments were extracted for the human-based analysis (for the category of phrasal verbs only 57 instead of 100 segments could be found in the given corpus, leading to...
a total of 657 instead of 700). As described above, each source segment contains at least one instance of the respective phenomenon, in many cases several instances could be found within one segment in this analysis, resulting in 2104 phenomena overall (Table 1).

As it can be seen in Table 1, the overall average performance of the systems is very similar for all systems. The SMT, RBMT and neural system slightly outperform the other systems with a 0.95 confidence level on the average performance. This is an interesting observation as the performance on the linguistic phenomena is quite diverse: While a shallow evaluation would render the systems more or less identical, this view makes it possible to identify and study their strengths and weaknesses in detail. Note that none of the systems was optimised to perform particularly well on these phenomena, although it is expected that the RBMT system already contained hand-written rules to handle linguistic phenomena.

Table 1. Translation accuracy on manually evaluated sentences focusing on particular phenomena. Boldface indicates best system on each phenomenon (row) with 0.95 confidence level.

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>#</th>
<th>SMT</th>
<th>RBMT</th>
<th>SMT-syntax</th>
<th>Linked data</th>
<th>Neural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imperatives</td>
<td>247</td>
<td>68%</td>
<td>79%</td>
<td>68%</td>
<td>68%</td>
<td>74%</td>
</tr>
<tr>
<td>Compounds</td>
<td>219</td>
<td>55%</td>
<td>87%</td>
<td>55%</td>
<td>56%</td>
<td>51%</td>
</tr>
<tr>
<td>“&gt;”-separators</td>
<td>148</td>
<td>99%</td>
<td>39%</td>
<td>97%</td>
<td>97%</td>
<td>93%</td>
</tr>
<tr>
<td>Quotation marks</td>
<td>431</td>
<td>97%</td>
<td>94%</td>
<td>93%</td>
<td>94%</td>
<td>95%</td>
</tr>
<tr>
<td>Verbs</td>
<td>505</td>
<td>85%</td>
<td>93%</td>
<td>81%</td>
<td>85%</td>
<td>90%</td>
</tr>
<tr>
<td>Phrasal verbs</td>
<td>89</td>
<td>21%</td>
<td>67%</td>
<td>7%</td>
<td>11%</td>
<td>38%</td>
</tr>
<tr>
<td>Terminology</td>
<td>465</td>
<td>63%</td>
<td>50%</td>
<td>53%</td>
<td>51%</td>
<td>55%</td>
</tr>
<tr>
<td>Sum</td>
<td>2104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>76%</td>
<td>76%</td>
<td>71%</td>
<td>72%</td>
<td>75%</td>
</tr>
</tbody>
</table>

While the baseline SMT system outperforms the other systems on terminology and (except for the neural) on quotation marks, all three SMT systems outperform the RBMT system on the “>”-separators, but only the baseline is significantly better than the neural in this category. The RBMT system shows a complementary performance compared to the SMT baseline system, as it outperforms all other systems on compounds, verbs and phrasal verbs, and outperforms the three SMT systems on imperatives.

It can be stated that the RBMT system shows the tendency to perform better on the morpho-syntactic linguistic categories (i.e., imperatives, compounds, verbs and phrasal verbs), while the baseline SMT systems seem to be able to handle the remaining categories better (namely the “>”-separators, quotation marks and terminology). The tendency on the performance regarding these categories is similar for the other two SMT systems (SMT-syntax, linked data) but generally less pronounced.
Concerning the neural system, the individual categories indicate that it performs very close to the SMT system in overall, but it improves significantly on it concerning verbs and phrasal verbs. Furthermore, it is remarkable that the neural system is the only system that reaches 90% accuracy or more on three categories. Nevertheless, our neural system is rather premature and there may still be implementation issues. Particularly the fact that compounds are not properly formed despite the byte pair encoding indicates that further work needs to focus on the performance and integration of this module.

The generally lower scores on phrasal verbs and terminology for all systems might be an indication that at the present time these are the categories (at least of those categories we inspected) causing the most difficulties for all systems – regardless of the nature of the system. Thus, future work might in a first step focus on tackling these problems.

Lastly, it is interesting to mention that there were cases in which the translation of the MT system was found to be better than the reference translation, as can be seen in Example (9) in which the reference contains the English spelling of the term email. The correct German spelling can be seen in the SMT-syntax output (E-Mail).

Example 9.

Source: Send an email to [...].
Neural: Senden Sie eine E-Mail an [...].
Reference: Senden Sie eine email an [...].

3.3. Evaluation of test suite data

In addition to the evaluation of our systems on the seven domain-specific categories, we also evaluated the systems on a small-scale generic test suite by creating 100 test sentences of 50 general linguistic categories (two sentences per category). These 50 linguistic categories can be condensed to fourteen super-categories.

The evaluation process was conducted the same way as in the domain-specific analysis: The correctly translated phenomena per category were counted and thus the overall sum of correctly translated phenomena was divided by the overall number of instances in the phenomenon.

Even though we are aware that the analysis on such few instances per category is not necessarily representative, the evaluation of this data still provides interesting insights into the distinct nature of the systems. Table 2 shows the behaviour of the systems on the different super-categories.

The best-performing system on this data selection is the RBMT system, as it shows an average percentage of correct translations more than twice of the SMT, SMT-syntax and linked data systems. While the latter three systems have very similar average scores ranging 28-31%, the neural system has the second-highest average score, namely 48%.

The three SMT systems do not only have similar overall average scores but also behave similarly regarding various phenomena: In six of the fourteen super-categories, the baseline SMT, SMT-syntax and linked data system correctly translate the same percentage of test sentences (on false friends, function words, composition, Named Entity (NE) & terminology, negation and punctuation). On four of these
super-categories, all three systems reach 50% or more, the SMT baseline and SMT-syntax additionally have 50% or more on two categories.

The neural system reaches 50% or more on eight of the fourteen categories while the RBMT system shows this property on eleven systems.

Table 2. Translation accuracy on test suite sentences focusing on particular phenomena. Boldface indicates best system on each phenomenon (row) with a 0.95 confidence level when significant.

<table>
<thead>
<tr>
<th>Super-category</th>
<th>#</th>
<th>SMT</th>
<th>RBMT</th>
<th>SMT-syntax</th>
<th>Linked data</th>
<th>Neural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambiguity</td>
<td>2</td>
<td>50%</td>
<td>50%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Coordination &amp; ellipsis</td>
<td>8</td>
<td>13%</td>
<td>13%</td>
<td>0%</td>
<td>0%</td>
<td>13%</td>
</tr>
<tr>
<td>False friends</td>
<td>2</td>
<td>100%</td>
<td>50%</td>
<td>100%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Function word</td>
<td>4</td>
<td>50%</td>
<td>75%</td>
<td>50%</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>LDD &amp; interrogative</td>
<td>16</td>
<td>25%</td>
<td>69%</td>
<td>19%</td>
<td>25%</td>
<td>63%</td>
</tr>
<tr>
<td>MWE</td>
<td>10</td>
<td>40%</td>
<td>40%</td>
<td>50%</td>
<td>50%</td>
<td>10%</td>
</tr>
<tr>
<td>Composition</td>
<td>2</td>
<td>0%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>NE &amp; terminology</td>
<td>6</td>
<td>50%</td>
<td>67%</td>
<td>50%</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>Negation</td>
<td>2</td>
<td>50%</td>
<td>100%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Non-verbal agreement</td>
<td>8</td>
<td>50%</td>
<td>88%</td>
<td>38%</td>
<td>38%</td>
<td>25%</td>
</tr>
<tr>
<td>Punctuation</td>
<td>2</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>50%</td>
</tr>
<tr>
<td>Subordination</td>
<td>10</td>
<td>20%</td>
<td>90%</td>
<td>30%</td>
<td>30%</td>
<td>60%</td>
</tr>
<tr>
<td>Verb tense/mood/asp.</td>
<td>18</td>
<td>33%</td>
<td>89%</td>
<td>17%</td>
<td>11%</td>
<td>78%</td>
</tr>
<tr>
<td>Verb valency</td>
<td>10</td>
<td>10%</td>
<td>80%</td>
<td>30%</td>
<td>30%</td>
<td>50%</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>31%</td>
<td>69%</td>
<td>29%</td>
<td>28%</td>
<td>48%</td>
</tr>
</tbody>
</table>

Even though we know that this small-scale study is not fully representative, we calculated the statistical significance of the best system per phenomenon. We found the RBMT and neural system to be the best systems on the function words, Long Distance Dependency (LDD) & interrogative and verb tense/mood/aspect. The SMT-syntax and the linked data system are outperforming the neural system on the MultiWord Expressions (MWE), but not the baseline SMT or the RBMT. The RBMT is additionally the best system on composition and non-verbal agreement. Furthermore, the RBMT is better than the SMT-syntax and linked data system on subordination and verb valency. In these two categories, the neural additionally outperforms the SMT system. It is also worth noting that out of two samples
containing negation, only the RBMT system translated both of them correctly, whereas all of the statistical systems missed one.

The general conclusions that can be drawn from this small test suite evaluation are that the RBMT seems to handle the given linguistic phenomena better than the other systems. Moreover, the neural system is not as good as the RBMT system but still better than the SMT systems.

Among the SMT systems, the baseline SMT system treats the phenomena a little bit better than the other two SMT systems, just like in the domain-specific analysis. In addition to that, the RBMT system is in both analyses one of the best systems/the best system.

4. Conclusion and outlook

In this paper we have described several ways of making machine translation more linguistically aware. We have attempted to introduce linguistically aware phrases in the models as well as show improvements in the translation of named entities by linking with semantic web resources such as the DBpedia. Our detailed evaluation of relevant linguistic phenomena has shown that the performance of the MT systems differs considerably with respect to these phenomena while their overall performance in terms of errors made on these phenomena is very much the same. While the extended systems had previously shown performance improvements in automatic tests on larger corpora, we could not find such indications in the selected test items. However, the systems were not optimized for performance on the test suite. In this sense, this approach can really be seen as a “stress test”. Moreover, the manual evaluation of SMT-syntax and linked data systems highlight the limitations inherent in such approaches dependent on external tools with their own set of errors. For example, the SMT-syntax system is sensitive to errors in the respective language parsers as well as the statistical tree aligner employed to extract the linguistically motivate phrase pairs. The linked data system obtains its translations from user-generated knowledge bases and is also limited by the performance of the Named Entity Recognition system employed to identify the named entities.

Interestingly enough, the more general evaluation shows first indications that the neural system is capable of learning several aspects of the language that are coded in the rules of the RBMT in a better way than the phrase-based SMT systems. They certainly lack abstraction (and generalization) in this respect. We are convinced that this test-suite based approach will lead to more insights in the future and will become important, e.g., in the area of machine teaching for neural MT.

Given this detailed method and results, it is now possible to select/improve systems with respect to a given task (an extension of this work for the purpose of the WMT16 Shared Task is presented in [2]). For example, if there is a post-editor involved, one would focus on fixing issues that are hard to post-edit. If the goal is to provide information to end users, one would focus on those issues that affect readability most. This prioritization would not be possible when using today’s automatic measures. One obvious way for improving statistical systems would be to...
create targeted training material focusing on the relevant aspects such as imperatives starting from the test items.

In order to adapt the evaluation for other language pairs, it might be helpful to draw inspiration from the evaluation done in the context of this paper, but it would also include extensive manual work due to this approach being language-dependant. Furthermore, the choice of the phenomena is a subjective decision, which means that many more/different categories could be investigated, as for instance lexical choice, modal verbs, etc.

Adaptation of the evaluation to a different task is a manual step involving human expertise. Once the community (including industry) has come up with a set of test suites for certain tasks/requirement, it will be easier to put together tests for new tasks from these sources.

Acknowledgements: This article has received support from the EC’s Horizon 2020 research and innovation programme under Grant agreements No 645452 (QT21), from FP7 (2007-2013) under Grant agreement No 610516: “QTLeap: Quality Translation by Deep Language Engineering Approaches”, and by Project Digitale Kreationstechnologien (DKT), supported by the German Federal Ministry of Education and Research (BMBF). “Unternehmen Region”, instrument “Wachstumskern-Potenzial” (No 03WKP45). We also thank the two anonymous reviewers for their valuable comments.

References


A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines

Aljoscha Burchardt, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, Philip Williams

German Research Center for Artificial Intelligence (DFKI)
RWTH Aachen University
University of Edinburgh

Abstract
In this paper, we report an analysis of the strengths and weaknesses of several Machine Translation (MT) engines implementing the three most widely used paradigms. The analysis is based on a manually built test suite that comprises a large range of linguistic phenomena. Two main observations are on the one hand the striking improvement of a commercial online system when turning from a phrase-based to a neural engine and on the other hand that the successful translations of neural MT systems sometimes bear resemblance with the translations of a rule-based MT system.

1. Introduction
Test suites are a familiar tool in NLP in areas such as grammar checking, where one may wish to ensure that a parser is able to analyse certain sentences correctly or test the parser after changes to see if it still behaves in the expected way. In contrast to a “real-life” corpus the input in a test suite may well be made-up or edited to isolate and illustrate issues.

Apart from several singular attempts (King and Falkedal, 1990; Isahara, 1995; Koh et al., 2001, etc.) broadly-defined test suites have not generally been used in MT research. One of the reasons for this might be the fear that the performance of statistical MT systems depends so much on the particular input data, parameter settings, etc., that final conclusions about the errors they make, particularly about the different
reasons (e.g., length of n-grams, missing training examples), are difficult to obtain. A related concern is that statistical MT systems are designed to maximise scores on test corpora that are comparable to the training/tuning corpora and that it is therefore unreliable to test these systems in different settings. While these concerns may hold for systems trained on very narrowly-defined domains, genres, and topics (such as biomedical patent abstracts), in fact many systems are trained on large amounts of data covering mixed sources and are expected to generalize to some degree.

A last reason might be that “correct” MT output cannot be specified in the same way as the output of other language processing tasks like parsing or fact extraction where the expected results can be more or less clearly defined. Due to the variation of language, ambiguity, etc., checking and evaluating MT output can be almost as difficult as the translation itself. Still, people have tried to automatically classify errors comparing MT output to reference translations or post-edited MT output using tools like Hjerson (Popovic, 2011).

In narrow domains there seems to be interest in detecting differences between systems and within the development of one system, e.g., in terms of verb-particle constructions (Schottmüller and Nivre, 2014) or pronouns (Guillou and Hardmeier, 2016). Bentivogli et al. (2016) performed a comparison of neural- with phrase-based MT systems on IWSLT data using a coarse-grained error typology. Neural systems have been found to make fewer morphological, lexical and word-order errors.

Below, we present a pioneering effort to address translation barriers in a systematic fashion. We are convinced that testing of system performance on error classes leads to insights that can guide future research and improvements of systems. By using test suites, MT developers will be able to see how their systems perform compared to scenarios that are likely to lead to failure and can take corrective action.

This paper is structured as follows: After the general introduction (Section 1), Section 2 will briefly introduce the test suite we have used in the experiments reported in Section 3. Section 4 concludes the paper.

2. The Test Suite

The experiments reported below are based on a test suite for MT Quality we are currently building for the language pair English – German in the QT21 project. The test suite itself will be described in more detail in a future publication. In brief, it contains segments selected from various parallel corpora and drawn from other sources such as grammatical resources, e.g., the TSNLP Grammar Test Suite (Lehmann et al., 1996) and online lists of typical translation errors.

Each test sentence is annotated with the phenomenon category and the phenomenon it represents. An example showing these fields can be seen in Table 1 with the first column containing the source segment and the second and third column containing the phenomenon category and the phenomenon, respectively. The fourth column shows the translation given by the old Google Translate system and the last column
contains a post-edit of the MT output that is created by making as few changes as possible. In our latest version of the test suite, we have a collection of about 5,000 segments per language direction that are classified in about 15 categories (most of them similar in both language directions) and about 120 phenomena (many of them similar but also some differing, as they are language-specific). Depending on the nature of the phenomenon, each is represented by at least 20 test segments in order to guarantee for a balanced test set. The categories cover a wide range of different grammatical aspects that might or might not lead to translation difficulties for a MT system. Currently, we are still in the process of optimising our test segments and working on an automatic solution for the evaluation.

<table>
<thead>
<tr>
<th>Source</th>
<th>Phenomenon Category</th>
<th>Phenomenon</th>
<th>Target (raw)</th>
<th>Target (edited)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena machte sich früh vom Acker.</td>
<td>MWE</td>
<td>Idiom</td>
<td>Lena [left the field early].</td>
<td>Lena left early.</td>
</tr>
<tr>
<td>Lisa hat Lasagne gemacht, sie ist schon im Ofen.</td>
<td>Non-verbal agreement</td>
<td>Coreference</td>
<td>Lisa has made lasagne, [she] is already in the oven.</td>
<td>Lisa has made lasagna, it is already in the oven.</td>
</tr>
<tr>
<td>Ich habe der Frau das Buch gegeben.</td>
<td>Verb tense/aspect/mood</td>
<td>Ditransitive - perfect</td>
<td>I [have] the woman of the Book.</td>
<td>I have given the woman the book.</td>
</tr>
</tbody>
</table>

Table 1. Example test suite entries German—English (simplified for display purposes).

For the experiments presented here, we have used a preliminary version of our test suite (ca. 800 items per language direction, to a large extent verb paradigms) to include the changes of Google Translate which has recently been switched from a phrase-based to neural approach according to the companies’ publications. There are more than 100 different linguistic phenomena that we investigated in this version of the test suite in each language direction. In this preliminary version, the number of instances reported in the experiments below strongly varies among the categories (as well as between the languages).

3. Evaluating PBMT, NMT, and RBMT Engines and an Online System

3.1. System Description

We have evaluated several engines from leading machine translation research groups and a commercial rule-based system on the basis of the very same test suite version to be able to compare performance with the leading online system that has recently switched to a neural model. We included a number of different NMT systems with different properties and levels of sophistication to shed light on how these
new types of systems perform on the different kinds of phenomena. Below, we will briefly describe the systems.


**O-NMT** New version of Google Translate (web interface, Nov. 2016).

**OS-PBMT** Open-source phrase-based system that primarily uses a default configuration to serve as a baseline. This includes a 5-gram modified Kneser-Ney language model, nkcls and MGiza for alignment, GDF A phrase extraction with a maximum phrase length of five, msd-bidi-fe lexical reordering, and the Moses decoder (Koehn et al., 2007). The WMT’16 data was Moses-tokenized and normalized, truecased, and deduplicated.

**DFKI-NMT** Barebone neural system from DFKI. The MT engine is based on the encoder-decoder neural architecture with attention. The model was trained on the respective parallel WMT’16 data.

**ED-NMT** Neural system from U Edinburgh. This MT engine is the top-ranked system that was submitted to the WMT’16 news translation task (Sennrich et al., 2016). The system was built using the Nematus toolkit. Among other features, it uses byte-pair encoding (BPE) to split the vocabulary into subword units, uses additional parallel data generated by back-translation, uses an ensemble of four epochs (of the same training run), and uses a reversed right-to-left model to rescore n-best output.

**RWTH-NMT** NMT-system from RWTH (only used for German – English experiments). This system is equal to the ensemble out of 8 NMT systems optimized on TEDx used in the (Peter et al., 2016) campaign. The eight networks used make use of subwords units and are finetuned to perform well on the IWSLT 2016 MSLT German to English task.

**RBMT** Commercial rule-based system Lucy (Alonso and Thurmair, 2003).

### 3.2. Evaluation Procedure

In order to evaluate a system’s performance on the categories in the test suite, we concentrate solely on the phenomenon in the respective sentence and disregard other errors. This means that we have to determine whether a translation error is linked to the phenomenon under examination or if it is independent from the phenomenon. If the former is the case, the segment will be validated as incorrect. If, however, the error in the translation can not be traced back to the phenomenon, the segment will be counted as correct.

Currently, the system outputs are being automatically compared to a “reference translation” which is, in fact, a post-edit of the O-PBMT output as those were the very first translations to be generated and evaluated when we started building the test suite (see description of the test suite in Section 2 and Table 1). In a second step,
all the translations that do not match the “reference” are manually evaluated by a professional linguist since the translations might be very different from the O-PBMT post-edit but nevertheless correct. As this is a very time-consuming process, we are currently working on automating this evaluation process by providing regular expressions for various possible translation outputs – naturally, only focusing on the phenomenon under investigation.

We refrain from creating an independent reference as we think that generating the regular expressions that focus solely on the phenomena instead is the more sophisticated solution in this context. As a consequence, we cannot compute automatic scores like BLEU. We do not see this as a disadvantage as with the test suite we want to focus rather on gaining insights about the nature of translations than on how well translations match a certain reference.

3.3. Results German – English

Table 2 shows the results for the translations from German to English from the different systems on the categories. The second column in the table (“#”) contains the number of instances per category. As the distribution of examples per category in this old version of our test suite was very unbalanced with some categories having only very few examples, some more categories we tested were excluded from the analysis we present here.

Before we discuss the results, we want to point out that the selection of phenomena and the number of instances used here is not representative of their occurrence in
corpora. Consequently, it can not be our goal to find out which of the systems is the globally “best” or winning system. Our goal is to check and illustrate the strengths and weaknesses of system (types) with respect to the range of phenomena we cover with this version of the test suite. Using this evaluation approach, researchers and system developers ideally can form hypotheses about the reasons why certain errors happen (systematically) and can come up with a prioritised strategy for improving the systems. Our ultimate goal is to represent all phenomena relevant for translation in our test suite.

Coming to the analysis, it is first of all striking how much better the neural version of Google Translate (O-NMT) is as compared to its previous phrase-based version (O-PBMT). Interestingly, the O-NMT and the RBMT – two very different approaches – are the best-performing systems on average, achieving almost the same amount of correct translations on average, i.e., 73%, resp. 75%, but looking at the scores of the categories reveals that the performance of the two systems regarding the categories is in fact very diverse. While the O-NMT system is the most-frequent best-performing system per phenomenon, as it is best on composition, function words, long distance dependency (LDD) & interrogative, multi-word expressions (MWE), subordination and verb valency, the RBMT is only the best system on ambiguity\(^2\) and verb tense/aspect/mood. The high number of instances of the latter category leads to the high average score of the RBMT system, as verb paradigms are part of the linguistic information RBMT systems are based on.

The OS-PMBT reaches the lowest average score, but it is nevertheless the best-performing system on named entities (NE) & terminology. The DFKI-NMT system reaches a higher average score than the PBMT system (four percentage points more). The RWTH-NMT is (along with the O-NMT) the best-performing system on function words. On average it reaches 63% of correct translations. The ED-NMT outrules (also along with the O-NMT) the other systems on composition and verb valency and reaches 56% correct translations on average.

In order to see if we find some interesting correlations that might serve as a preview for more extensive analyses with a more solid and balanced amount of test segments in the future, we have calculated Pearson’s coefficient over the phenomenon counts (being aware that we are dealing with very small numbers here). As the correlations for the direction English – German were higher and for space reasons, we will show the numbers only for the other direction in the following Subsection to give an indication about possible future work.

One general impression that will also be supported by the examples below is that NMT seems to learn some capabilities that the RBMT system has. It may lead to the speculation that NMT indeed learns something like the rules of the language. This, however, needs more intensive investigation. Another interesting observation is that

\(^2\)The good performance of RBMT on ambiguity can be explained by the very small number of items and it is more or less accidental that the preferred readings were the ones the RBMT has coded in its lexicon.
The RWTH-NMT system has a lower overall correlation with the other NMT systems. This might be because it has also been trained and optimised on transcripts of spoken language as opposed to the other systems trained solely on written language.

The following examples depict interesting findings from the analysis and comparison of the different systems. When a system created a correct output (on the respective category), the system’s name is marked in boldface.

(1) **Source:** Warum hörte Herr Muschler mit dem Streichen auf?  
**Reference:** Why did Mr. Muschler stop painting?  
**O-PBMT:** Why heard Mr Muschler on with the strike?  
**O-NMT:** Why did Mr. Muschler stop the strike?  
**RBMT:** Why did Mr Muschler stop with the strike?  
**OS-PBMT:** Why was Mr Muschler by scrapping on?  
**DFKI-NMT:** Why did Mr Muschler listen to the rich?  
**RWTH-NMT:** Why did Mr. Muschler listen to the stroke?  
**ED-NMT:** Why did Mr. Muschler stop with the stump?

Example (1) contains a phrasal verb and belongs to the category composition. German phrasal verbs have the characteristics that their prefix might be separated from the verb and move to the end of the sentence in certain constructions, as it has happened in example (1) with the prefix *auf* being separated from the rest of the verb *hören*. The verb *aufhören* means *to stop*, but the verb *hören* without the prefix simply means *to listen*. Thus, phrasal verbs might pose translations barriers in MT when the system translates the verb separately not taking into account the prefix at the end of the sentence. The output of the O-PBMT, DFKI-NMT and RWTH-NMT indicates that this might have happened. The O-NMT, RBMT and the ED-NMT correctly translate the verb which could mean that more context (and thus, including the prefix *auf* at the end of the sentences) was taken into account for the generation of the output.

(2) **Source:** Warum macht der Tourist drei Fotos?  
**Reference:** Why does the tourist take three fotos?  
**O-PBMT:** Why does the tourist three fotos?  
**O-NMT:** Why does the tourist make three fotos?  
**RBMT:** Why does the tourist make three fotos?  
**OS-PBMT:** Why does the tourist three fotos?  
**DFKI-NMT:** Why does the tourist make three fotos?  
**RWTH-NMT:** Why is the tourist taking three fotos?  
**ED-NMT:** Why does the tourist make three fotos?

One of the phenomena in the category LDD & interrogative is wh-movement. It is for example involved in wh-questions, like in the sentence in (2). A wh-question in English is usually built with an auxiliary verb and a full verb, e.g., wh-word + *to*...
have/to be/to do + full verb. In German on the other hand, an auxiliary verb is not necessarily needed. This fact might lead to translation difficulties, as can be seen in (2), where the O-PBMT and the OS-PBMT treat the verb *does* as a full verb instead of an auxiliary verb. All the other systems translate the question with two verbs, however, except for the RWTH-NMT, they all mistranslate *ein Foto machen* as *to make a foto* (literal translation) instead of *to take a foto*. Nevertheless, these translations count as correct, since they do contain an auxiliary verb + a full verb.

(3) **Source:** Die Arbeiter müssten in den sauren Apfel beißen.
**Reference:** The workers would have to bite the bullet.
**O-PBMT:** The workers would have to bite the bullet.
**O-NMT:** The workers would have to bite into the acid apple.
**RBMT:** The workers would have to bite in the acid apple.
**OS-PBMT:** The workers would have to bite the bullet.
**DFKI-NMT:** Workers would have to bite in the acid apple.
**RWTH-NMT:** The workers would have to bite into the clean apple.
**ED-NMT:** The workers would have to bite in the acidic apple.

Idioms are an interesting phenomenon within the category MWE. The meaning of an idiom in one language can not be transferred to another language by simply translating the separate words, as the meaning of these multi-word units goes beyond the meaning of the separate words. As a consequence, idioms have to be transferred to another language as a whole. For German <> English it is often the case that an idiom in one language can be transferred to another idiom in the other language. This is also the case in example (3). The German idiom *in den sauren Apfel beißen* can be translated as *to bite the bullet*. Only the two PBMT system correctly translate this idiom, the other systems all give a literal translation - with the RWTH-NMT translating *sauren* as *clean* instead of *acid(ic)* like the other systems, probably not knowing the word *sauren* and instead translating the similar word *sauberen*. This is one example where a phrase-based approach has a real advantage (if the phrase was in the training data).

(4) **Source:** Wie kann ich die Farbe, mit der ich arbeite, ändern?
**Reference:** How can I change the color I am working with?
**O-PBMT:** How can I change the color *with which I work* to change?
**O-NMT:** How can I change the color *with which I work*?
**RBMT:** How can I change the color *with which I work*?
**OS-PBMT:** How can I change the color, *with whom I work*, change?
**DFKI-NMT:** How can I change the color *I work with*?
**RWTH-NMT:** How can I change the color *I work with*?
**ED-NMT:** How can I change the color *I work with*?
The sentence in (4) contains a relative clause which belongs to the category subordination. Relative clauses in English can, but do not have to contain a relative pronoun. The outputs in (4) show both properties. The O-PBMT and the OS-PBMT double the verb change, the remaining systems correctly translate the relative clause.

(5)

Source: Ich hätte nicht lesen gedurft.
Reference: I would not have been allowed to read.
O-PBMT: I would not have been allowed to read.
O-NMT: I should not have read.
RBMT: I would not have been allowed to read.
OS-PBMT: I would not have read gedurft.
DFKI-NMT: I would not have been able to read.
RWTH-NMT: I wouldn’t have read.
ED-NMT: I wouldn’t have read.

Verb paradigms (verb tense/aspect/mood) make up about one third of the whole test suite. Example (5) shows a sentence with a negated modal verb, in the tense pluperfect subjunctive II. This is a quite complex construction, thus it is not surprising that only few systems correctly translate the sentence. As might be expected, one of them is the RBMT system. The second one is the O-PBMT. The neural version of this system on the other hand does not correctly produce the output.

3.4. Results English – German

The results for the English – German translations can be found in Table 3. For this language direction, only five systems were available instead of seven like for the other direction. As in the analysis for the other language direction, we excluded the categories that had too few instances from the table. Nevertheless, similarities between the categories of both language directions can be found.

As in the German – English translations, the RBMT system performs best of all systems on average, reaching 83%. It performs best of all systems on verb tense/aspect/mood and verb valency. The second-best system is – just like in the other language direction but with a greater distance (seven percentage points less on average, namely 76%) – the O-NMT. The O-NMT shows quite contrasting results on the different categories, compared to RBMT: it outrules (most of) the other systems on the remaining categories, i.e., on coordination & ellipsis, LDD & interrogative, MWE, NE & terminology, special verb types and subordination.

The third-best system on average is the ED-NMT system. It reaches an average of 61% correct translations. The other remaining NMT system, the barebone DFKI-NMT system, reaches 11 percentage points less on average than the ED-NMT, for it reaches 50%. But it outrules the other systems on subordination along with O-NMT. The system with the lowest average score is the previous version of Google Translate,
Table 3. Results of English – German translations. Boldface indicates best system(s) on each category (row).

<table>
<thead>
<tr>
<th>Category</th>
<th>O-PBMT</th>
<th>O-NMT</th>
<th>RBMT</th>
<th>DFKI-NMT</th>
<th>ED-NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordination &amp; ellipsis</td>
<td>17</td>
<td>6%</td>
<td>47%</td>
<td>29%</td>
<td>24%</td>
</tr>
<tr>
<td>LDD &amp; interrogative</td>
<td>70</td>
<td>19%</td>
<td>61%</td>
<td>54%</td>
<td>41%</td>
</tr>
<tr>
<td>MWE</td>
<td>42</td>
<td>21%</td>
<td>29%</td>
<td>19%</td>
<td>21%</td>
</tr>
<tr>
<td>NE &amp; terminology</td>
<td>20</td>
<td>25%</td>
<td>80%</td>
<td>40%</td>
<td>45%</td>
</tr>
<tr>
<td>Special verb types</td>
<td>14</td>
<td>14%</td>
<td>86%</td>
<td>79%</td>
<td>29%</td>
</tr>
<tr>
<td>Subordination</td>
<td>35</td>
<td>11%</td>
<td>71%</td>
<td>54%</td>
<td>71%</td>
</tr>
<tr>
<td>Verb tense/aspect/mood</td>
<td>600</td>
<td>41%</td>
<td>82%</td>
<td>96%</td>
<td>53%</td>
</tr>
<tr>
<td>Verb valency</td>
<td>22</td>
<td>36%</td>
<td>59%</td>
<td>68%</td>
<td>64%</td>
</tr>
<tr>
<td>Sum</td>
<td>820</td>
<td>287</td>
<td>622</td>
<td>679</td>
<td>410</td>
</tr>
<tr>
<td>Average</td>
<td>35%</td>
<td>76%</td>
<td>83%</td>
<td>50%</td>
<td>61%</td>
</tr>
</tbody>
</table>

Table 4. Overall correlation of English – German systems

<table>
<thead>
<tr>
<th>Correlations</th>
<th>O-PBMT</th>
<th>O-NMT</th>
<th>RBMT</th>
<th>DFKI-NMT</th>
<th>ED-NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>O-PBMT</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O-NMT</td>
<td>0.34</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBMT</td>
<td>0.39</td>
<td>0.55</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFKI-NMT</td>
<td>0.28</td>
<td>0.36</td>
<td>0.36</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>ED-NMT</td>
<td>0.30</td>
<td>0.33</td>
<td>0.43</td>
<td>0.55</td>
<td>1.00</td>
</tr>
</tbody>
</table>

namely the O-PBMT. With 35% on average, it reaches less than half of the score of the O-NMT.

The results of the calculation of the Pearson’s coefficient can be found in Table 4. Only categories with more than 25 observations had their correlation analysed. For the interpretation, we used a rule-of-thumb mentioned in the literature³.

In the overall correlation, RBMT has a moderate correlation with O-NMT, which might be traced back to the fact that these are the two systems that correctly translate most of the test segments, compared to the other systems. The two neural systems, DFKI-NMT and ED-NMT, also have moderate correlations. All the other systems have weak correlation with each other.

Again, for the small and unbalanced numbers of samples, we do not want to put too much emphasis on the observations regarding correlations. This type of analysis might, however, become more informative in future work.

4. Conclusions and Outlook

While the selection of test items/categories and even more the selection of examples we discussed provides a selective view on the performance of the system, we are convinced that this type of quantitative and qualitative evaluation provides valuable insights and ideas for improvement of the systems, e.g., by adding linguistic knowledge in one way or another. Two main observations we want to repeat here is the striking improvement of the commercial online system when turning from a phrase-based to a neural engine. A second observation is that the successful translations of some NMT systems often bear resemblance with the translations of the RBMT system. Hybrid combinations or pipelines where RBMT systems generate training material for NMT systems seem a promising future research direction to us.

While the extracted examples above give very interesting insights on the systems’ performances on the categories, these are only more or less random spot tests. However, taking a close look at the separate phenomena at a larger scale and in more detail will lead to more general, systematic observations. This is what we aim to do with our current version of the test suite which is therefore much more extensive and systematic and therefore also allows for more general observations and more quantitative statements in future experiments.

Our ultimate goal is to automate the test suite testing. To this end, we are currently working on a method that is using regular expressions for automatically checking the output of engines on the test suite. The idea is to manually provide positive and negative tokens for each test item that can range from expected words in case of disambiguation up to, verbs and their prefixes with wild cards in between up to complete sentences in the case of verb paradigms.

Acknowledgements

This research is supported by the EC’s Horizon 2020 research and innovation programme under grant agreements no. 645452 (QT21).

Bibliography


Guillou, Liane and Christian Hardmeier. PROTEST: A Test Suite for Evaluating Pronouns in Machine Translation. In Chair), Nicoletta Calzolari (Conference, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, Proceedings of the Tenth Interna-
Quality Translation 21
D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

PBML 108 JUNE 2017


Address for correspondence:
Aljoscha Burchardt
aljoscha.burchardt@dfki.de
German Research Center for Artificial Intelligence (DFKI)
Language Technology Lab, Alt-Moabit 91c, 10559 Berlin, Germany

170
Can Out-of-the-box NMT Beat a Domain-trained Moses on Technical Data?
Anne Beyer∗, Vivien Macketanz∆, Aljoscha Burchardt∆ and Philip Williams⋄
∗ beo Gesellschaft für Sprachen und Technologie mbH
Ruppmannstrae 33b, 70565 Stuttgart, Germany
anne.beyer@beo-doc.de
∆ German Research Center for Artificial Intelligence (DFKI)
Language Technology Lab, Alt-Moabit 91c, 10559 Berlin, Germany
firstName.lastName@dfki.de
⋄ University of Edinburgh
School of Informatics, 10 Crichton Street, Edinburgh, EH8 9AB, UK
pwillia4@inf.ed.ac.uk

Abstract
In the last year, we have seen a lot of evidence about the superiority of neural machine translation approaches (NMT) over phrase-based statistical approaches (PBMT). This trend has shown for the general domain at public competitions such as the WMT challenges as well as in the obvious quality increase in online translation services that have changed their technology. In this paper, we take the perspective of an LSP. The questions we want to answer with this study is if now is already the time to invest in the new technology. To answer this question, we have collected evidence as to whether an existing state-of-the-art NMT system for the general domain can already compete with a domain-trained and optimised Moses (PBMT) system or if it is maybe already better. As it is well known that automatic quality measures are not reliable for comparing the performance of different system types, we have performed a detailed manual evaluation based on a text suite of domain segments.

1 Introduction
In the last year, we have seen a lot of evidence about the superiority of neural machine translation approaches (NMT) over phrase-based statistical approaches (PBMT). This trend has shown for the general domain at public competitions such as the WMT challenges (Bojar et al., 2016) as well as the obvious quality increase in online translation services that have changed their technology.1

When it comes to particular domains in the context of commercial translation services, the interest in NMT is huge, but we are not aware of systematic public studies about the performance of NMT in comparison to PBMT. While bigger companies are already in the process of changing their technology, smaller language service providers (LSP) have limited resources in their day-to-day-business both in terms of humans and compute power for undertaking the necessary experiments. For researchers, it is still difficult to obtain suitable training data in order to assess the potential of the new technology on in-domain data.

The background for this study was simply the question of an LSP if now is already the time to invest in the new technology. To answer this question, we wanted to collect evidence as to whether an existing state-of-the-art NMT system for the general domain can already compete with a domain-trained and optimised Moses (PBMT) system or if the former can maybe even outperform the latter already.

As we did not want to rely solely on automatic measures, we have performed a manual evaluation based on a phenomenon-driven test-suite, a method we have applied for evaluations in the technical domain before, e.g., in (Avramidis et al., 2016).

2 Experiment
2.1 Data
The customer data used in this study came from translations of catalogues for technical tools. Our

dataset consisted of translation tasks from German into British English assigned to beo over a course of two months. Overall, the set contained around 5,000 segments.

2.2 Phrase-based Statistical MT System

The PBMT system used is based on Moses (Koehn et al., 2007) and was adapted to integrate MT into the translation workflow at beo.

As training data we used the customer’s translation memory (TM) and terminology, which yielded a total of 337,600 segments. Formatting tags were removed from the data and it was tokenized and lower cased. As we translate from German, compounds were also split on the source side in order to reduce data sparseness in terms of unknown words. A 3-gram language model was built using IRSTLM (Federico et al., 2008).

The training procedure follows the baseline Moses setup, but the model was not tuned further, as no tuning setup was found yet which improved the system’s performance over the baseline, according to an internal evaluation with our translators. This is similar to what we found for other customer set-ups. It could be due to the fact that the training-data and the translations are very similar, as we only used in-domain data for training. We have not yet tried to add more out-of-domain data because this did not improve the usefulness of systems trained for other customers, but might look into that at a later point as well. As we are only concerned with the application of MT for post-editing, the quality requirements are different from other tasks such as quality evaluation and we rely more on post-editor feedback that automated quality scores.

For the translation, we used the M4Loc integration tools, a wrapper for Moses which extracts formatting tags before the translation and inserts them into the target afterwards according to the word alignment (Hudk and Ruopp, 2011). Furthermore, we ran a few test rounds on the customer data together with our translators and created a set of hand-crafted rules based on regular expressions which are applied after the MT to fix certain errors (e.g., with casing or spaces).

2.3 Neural MT System

The neural system that was used in this study was built by the University of Edinburgh. This MT engine is the top-ranked system that was submitted to the WMT ’16 news translation task (Sennrich et al., 2016). The system was built using the Nematus toolkit.4

As training data, only the official WMT task data was used – this system did not have access to the customer-specific data during training. The data was tokenized and truecased, and tokens on both the English and German sides were split into subword units using byte-pair encoding (BPE), a frequency-based method that aims to improve the handling of rare words.

The full training configuration and scripts for this system have been publicly released.3

2.4 Manual Evaluation Procedure

For the manual evaluation process, two professional (computational) linguists went through the data and identified recurring linguistic phenomena that are characteristic for this domain-specific data.6 In a second step, all the phenomena detected were narrowed down to the most prominent ones, namely formal address, genitive, modal construction, negation, passive voice, predicate adjective, prepositional phrase, terminology and tagging. Thereafter, 100 segments per phenomenon were extracted, resulting in a total of 900 segments. For each segment, the total occurrences of the respective phenomenon were counted. Then, the total occurrences of the phenomena in the MT outputs were counted. Consequential, translation accuracy was calculated by dividing the number of occurrences in the MT output by the total number of occurrences in the segments.

When evaluating the correctness of the translations, the focus lies solely on the respective phenomenon under consideration, other errors are ignored. For a translated phenomenon to be counted as correct, it does not necessarily exactly have to match the reference, but it can also be realized in a different linguistic construction expressing the same semantic meaning, e.g., a passive construction that is translated in active construction will

---

4https://github.com/rsennrich/nematus
5https://github.com/rsennrich/wmt16-scripts
6These “linguistic phenomena” are understood in a pragmatic sense and include a wide range of issues that can influence the translation quality.
have less components but if the meaning is translated correctly, the counting should be adjusted to the instances in the source accordingly.

3 Evaluation Results

Due to the repetitive nature of the customer data, some of the segments in our dataset were already part of the TM or very similar to segments in the TM and therefore part of the training data for the Moses system. In order not to distort the results too much, those segments where Moses exactly matched the reference translations were omitted from the automatic evaluation. For the manual evaluation, we did not exclude those segments.

3.1 Automatic Evaluation Results

Even though BLEU is not intended to be used in order to compare different MT systems, this is a practice that is performed quite often. In order to show how much different translation quality evaluation methods can vary, we also carried out an evaluation on BLEU and METEOR, cf. Table 1. For calculating the automatic score, all tags were removed from the segments and the reference, furthermore all numbers were replaced by “10” because there were cases in which the reference involved different tags/numbers than the segments.

<table>
<thead>
<tr>
<th></th>
<th>NMT</th>
<th>Moses</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>23.68</td>
<td>47.98</td>
</tr>
<tr>
<td>METEOR</td>
<td>28.46</td>
<td>38.26</td>
</tr>
</tbody>
</table>

Table 1: BLEU and METEOR scores.

As described above, the automatic evaluation has a clear bias towards Moses. This is amplified by the fact that the references were derived from post-edits of the Moses output. These segments are thus naturally more similar to the Moses output than to the completely independent NMT output. Despite removing the segments for which the translation by Moses exactly matched the reference, both BLEU and METEOR show distinctly better scores for Moses compared to the NMT system. Taking into account the manual evaluation, though, gives a different picture.

3.2 Manual Evaluation Results and Examples

Table 2 shows the results of the manual evaluation on segment-level. For the 900 segments extracted, 1,453 phenomena could be found altogether, as there was often more than one occurrence of the phenomenon per segment. Phenomena like terminology occur more frequently than phenomena like negation that rarely appear more than once within one segment. Percentage values in boldface indicate that the systems is significantly better on the respective phenomenon with a 0.95 confidence level.

<table>
<thead>
<tr>
<th>Phenomena</th>
<th>NMT %</th>
<th>Moses %</th>
</tr>
</thead>
<tbody>
<tr>
<td>formal address</td>
<td>90%</td>
<td>86%</td>
</tr>
<tr>
<td>genitive</td>
<td>92%</td>
<td>68%</td>
</tr>
<tr>
<td>modal construction</td>
<td>94%</td>
<td>75%</td>
</tr>
<tr>
<td>negation</td>
<td>93%</td>
<td>86%</td>
</tr>
<tr>
<td>passive voice</td>
<td>83%</td>
<td>40%</td>
</tr>
<tr>
<td>predicate adjective</td>
<td>81%</td>
<td>75%</td>
</tr>
<tr>
<td>prepositional phrase</td>
<td>81%</td>
<td>75%</td>
</tr>
<tr>
<td>terminology</td>
<td>35%</td>
<td>68%</td>
</tr>
<tr>
<td>tagging</td>
<td>83%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 2: Manual evaluation translation accuracy focusing on particular phenomena.

The NMT system outperforms Moses on three categories: genitive, modal construction and passive voice. Moses on the other hand outperforms NMT on terminology and tagging – which is not surprising as terminology was part of the TM and tagging was handled by an extra module. For the remaining phenomena, the systems show no statistical significantly variance. Additionally, the NMT system also outperforms Moses on the overall average. Nevertheless, it is important to keep in mind that the values of the manual evaluation only give insights on certain phenomena and do not necessarily represent the systems’ overall performance but can rather be interpreted as revealing a tendency. Interestingly, the tendency the manual evaluation displays is counter to that of the automatic scores shown in Table 1. This can be traced back to the training material for Moses which included the customer’s translation memory and terminology which has a high influence on the BLEU and METEOR scores. The manual evaluation results on the other hand imply that even if a translation deviates substantially from a given reference it can...
still be correct, a fact that is not taken into account in the automatic scores.

The following examples depict interesting findings from the analysis and comparison of the two systems. The relevant component of the sentence is underlined. When a system created a correct output for the respective phenomenon, the system name is marked in boldface.

(1) Source: Schweißbänder erhöhen wesentlich den Tragekomfort eines Helmes.
Ref.: Sweatbands significantly increase the wearing comfort of a helmet.
NMT: Welding belts significantly increase the wearing comfort of a helmet.
Moses: Welding belts significantly increase the wearing comfort of a helmet.

Example (1) contains the genitive eines Helmes that should correctly be translated as of a helmet. As can be seen, the NMT correctly translates the genitive while Moses leaves Helmes untranslated which makes it hard to tell whether it correctly translates the genitive. This was a systematic problem for Moses, as Moses left unknown words un-translated. The NMT system on the other hand often generated sentences that were grammatical and contained “only” mistranslated unknown words rather than untranslated unknown words. As a result, syntactic features like the genitive in example (1) can be maintained.

(2) Source: Dazu kann das Board werkzeuglos gedreht und wieder eingehängt werden.
Ref.: The board can be turned and re-attached.
NMT: The board can be rotated and remounted.
Moses: To do this, the board can be rotated and back.

Example (2) includes a modal verb construction. A modal verb is always followed by at least one other verb. In the construction above, the modal verb kann is followed by the two verbs gedreht and eingehängt as well as the verb werden. Those verbs form a processual passive construction. In order to count as correctly translated, the English MT outputs should also exhibit four verbs, as the construction is formed the same way in English. While the NMT system correctly translated all four verbs, Moses leaves out one verb. Note that the fact that both systems do not translate werkzeug-
los (without using tools in the reference) can be ignored in this evaluation as the focus lies exclusively on the phenomenon of modal verb constructions.

(3) Source: Die Panoramascheibe mit integriertem Seitenschutz sorgt für eine optimale Augenraumabdeckung.
Ref.: The panoramic lens with integral side protection ensures optimum coverage of the eye area.
NMT: The panorama disc with integrated side protection ensures optimal eye room cover.
Moses: The panoramic lens with integral side protection ensures optimum Augenraumabdeckung.

The third example given here is taken from the terminology category. Additionally, it contains tagging which can be ignored in this case. The source sentence contains three terms: Panoramascheibe, Seitenschutz and Augenraumabdeckung which should be translated as panoramic lens, side protection and coverage of the eye area, respectively. The NMT system only correctly translates side protection while mistranslating the other two terms, giving literal translations. Moses correctly translates two of the three terms, leaving Augenraumabdeckung untranslated. Nevertheless, at first glance the NMT output looks “better” because it does not leave words untranslated. When taking a closer look though, this assumption does not hold.

As Moses benefits in terms of knowing a subset of the terminology, we considered it reasonable to also analyze segments without terminology in order to draw some more general conclusions about the comparison between the two systems, independent of the domain. For this purpose, 90 segments without domain-specific terminology were extracted from the data set. These segments comprise 30 short (< 40 characters), 30 medium-length (40 - 79 characters) and 30 long (> 79 characters) items. Two annotators were asked to evaluate these segments individually, rating them on a scale from 1 - 3, with 1 = perfect translation, 2 = small errors, content still understandable, and 3 = unintelligible. The mean values of the two annotators can be found in Table 3. While the NMT's
performance is judged better for the longer segments. Moses’ performance is judged better for short and medium-length segments. Nevertheless, conducting a t-test showed that the differences in the mean values are not statistically significant. Yet, it should be kept in mind at this point that we did not expect the differences to be statistically significant as the population of segments examined was very small. We interpret the scores solely as a tendency.

Below, we will discuss an example from this category:

(4) Source: Neben den Bedingungen zur Aufstellung und Inbetriebnahme wird eine Vielzahl von technischen und gesetzlichen Anforderungen an das Lager selbst gestellt, um z. B. wassergefährdende Flüssigkeiten, Säuren und Laugen oder auch entzündbare Flüssigkeiten gesetzeskonform aufzubewahren und zu lagern.

Ref.: In addition to the conditions for erection and commissioning there are a wide variety of technical and legal requirements on the storage location itself, relating for instance to water-polluting liquids, acids and alkalis or also flammable liquids, which must be kept safe and stored in accordance with regulations.

NMT: In addition to the conditions for installation and commissioning, a wide range of technical and legal requirements will be placed on the warehouse itself in order to maintain and store, for example, water-hazardous liquids, acids and foliage, or even flammable liquids.

Moses: In addition to the conditions for erection and commissioning is a wide variety of technical and legal requirements of the stored even, e.g. for water-polluting liquids, acids and alkalis or flammable liquids legally compliant aufzubewahren and zu lagern.

Example (4) belongs to the long segments, having 293 characters. While there were long segments that consisted of several sentences, this segment comprises only one sentence. It contains an infinitive clause that reaches from the conjunction um to the verb zu lagern. While in German, objects are located between the conjunction and the last verb, in English the conjunction in order to is immediately followed by the verb in the infinitive with the objects being located behind the verb. The NMT system successfully manages to resolve this construction, placing the verbs at the right position while Moses not only leaves the verbs at the end of the sentence but also leaves one verb untranslated. This example depicts our finding that NMT can handle long sentences better than Moses.

At the same time, this sentence also highlights difficulties that can arise, e.g., for post-editing, by the fact that the NMT system substitutes unknown words in the source with similar words in order to be able to translate them. While in some cases this might work out well, there are other cases where it does not, as in example (4) above: The word Laugen (alkalis) was treated as the word Laub which means foliage, resulting in a rather curious translation. For post-editing this means that in order to detect erroneous translations it is crucial to check the NMT output very thoroughly because mistranslations might be harder to find than in a system output that contains untranslated words.

### 4 Conclusion and Outlook

From the viewpoint of the linguistic phenomena we have studied in our experiment, the answer to the question in the title of this paper would probably be a sentence beginning with “Yes, but . . .”. The reason for the restriction is that the two categories NMT can not yet handle as good as Moses are of high importance in the language business: tags and terminology.

Still, sooner rather than later there will be tag-handling components for NMT systems and the issues with terminology will probably vanish once the NMT is trained on customer domain data. So, from the analytic perspective we took here, NMT could indeed become a valid alternative to PBMT.
The purpose of this study was to determine if now is already the time for LSPs to start investing in NMT. Our comparison showed that even an out-of-the-box system can perform quite reasonably, although it was not trained on the specific data. Our next step will be to look into the OpenNMT system and to compare models trained on the same dataset. Here, we will also take a closer look at other important factors, such as the time and effort needed for setting up such a system, the different training and decoding times and the impact of different kinds of errors on the post-editing effort.

For this purpose, we plan to also perform productivity tests with post editors to get a second, less phenomenon-driven comparison between the systems. In this course, we may also re-calculate automatic scores using post-edits as reference translations to rule out the Moses bias we have clearly observed in the figures we have presented here. For scenarios without post-editing, it would also be interesting to repeat task-based evaluations like the one we present in (Gaudio et al., 2016).

Another follow-up study that could be conducted might focus on a comparison of systems which are more similar with regard to their setup of the training data. In doing so, it would be interesting to investigate whether, for instance, an NMT system’s BLEU and METEOR scores might get closer to those of an SMT system, and if the bias towards the NMT system in the manual evaluation scores persists or even increases.

Acknowledgement

This article has received support from the EC’s Horizon 2020 research and innovation programme under grant agreements no. 645452 (QT21). We thank the anonymous reviewers for their valuable feedback.

References


D Can Out-of-the-box NMT Beat a Domain-trained Moses on Technical Data?

LT04: Where do matters stand on commercial MT after the advent of neural technology?

Anne Beyer, Aljoscha Burchardt

LT04: Where did we stand?

Let’s try!

Where did we stand?

Integrated SMT into translation process

The advent of neural technology

The ideas date back to the 1940s
Advances in computational power gave rise to a re-discovery
Several breakthroughs in different areas (Image Classification, Game Playing, ...)

Gathered some experience with SMT

www.smt.org/moses

Persuaded translators to give it a try
Along comes NMT

Lots of media attention for NMT

Open-source toolkits (Nematus, OpenNMT)
Adoption of NMT in the industry (Google, Facebook, booking.com, ...)
(Preliminary) NMT solutions from MT providers (Systran, Lionbridge, KantanMT, ...)
How useful is it for LSPs (yet)?

Test suite approach

Test suites are a familiar tool in NLP in areas such as grammar development.
Idea: Use test suites in MT development.
By test suite, we refer to a selected set of source-target pairs that reflects interesting or difficult case (MWEs, long-distance, negation, terminology, etc.).
In contrast to a "real-life" corpus with reference translations, the input in a test suite may well be made-up or edited to isolate and illustrate issues.

Test suite approach

Goal: get quantitative and qualitative insights (system X gets all 20 imperatives right, but only 50% of the negations)
Compare different types of systems more reliably
Testing can be local/partial, based on phenomena of interest

Exemplary test suite entries

De-En

<table>
<thead>
<tr>
<th>Source</th>
<th>Category</th>
<th>Phenomenon</th>
<th>Target (raw)</th>
<th>Target (edited)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena machte sich früh vom Acker.</td>
<td>MWE</td>
<td>Idiom</td>
<td>Lena [+leG the field] early.</td>
<td>Lena leG early.</td>
</tr>
<tr>
<td>Lisa hat das Lasagne gemacht, sie ist schon im Ofen.</td>
<td>Non-verbal agreement</td>
<td>Coreference</td>
<td>Lisa has made lasagna, [she] is already in the oven.</td>
<td>Lisa has made lasagna, it is already in the oven.</td>
</tr>
<tr>
<td>Ich habe der Frau das Buch gegeben.</td>
<td>Verb tense/aspect/mood</td>
<td>Ditransitive - perfect</td>
<td>I [have] the woman of the Book.</td>
<td>I have given the woman the book.</td>
</tr>
</tbody>
</table>
Test suite experiment – evaluation procedure

- So far: manual checking
- One phenomenon at a time, e.g.:
  - For ambiguity: Do I find the right sense, no matter what I find in the rest of the sentence?
  - For a prefix verb: Do I find both parts?
  - For an English question: Do I see the Wh-Word and two verbs?
  - For a verb paradigm “X has given Y to Z”: Is the sentence complete and correct?

Count results

<table>
<thead>
<tr>
<th></th>
<th>O-PBMT</th>
<th>O-NMT</th>
<th>RBMT</th>
<th>OS-PBMT</th>
<th>DFKI-NMT</th>
<th>RWTH-NMT</th>
<th>ED-NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambiguity</td>
<td>17</td>
<td>58</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Composition</td>
<td>11</td>
<td>27</td>
<td>55</td>
<td>27</td>
<td>55</td>
<td>27</td>
<td>55</td>
</tr>
<tr>
<td>Coordination &amp; ellipsis</td>
<td>11</td>
<td>25</td>
<td>100</td>
<td>38</td>
<td>63</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>False friends</td>
<td>1</td>
<td>85</td>
<td>20</td>
<td>20</td>
<td>45</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>For a prefix verb</td>
<td>151</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>For an English question</td>
<td>53</td>
<td>25</td>
<td>74</td>
<td>55</td>
<td>55</td>
<td>74</td>
<td>55</td>
</tr>
<tr>
<td>For a verb paradigm “X has given Y to Z”</td>
<td>15</td>
<td>20</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Negation</td>
<td>1</td>
<td>85</td>
<td>20</td>
<td>20</td>
<td>45</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>Subordination</td>
<td>10</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Verb tense/aspect</td>
<td>58</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>True friends</td>
<td>5</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Average</td>
<td>46</td>
<td>73</td>
<td>74</td>
<td>43</td>
<td>47</td>
<td>63</td>
<td>56</td>
</tr>
</tbody>
</table>

Test suite experiment – results (De-En)

(1) Source: Er hat einen Kater, weil er sehr tierlieb ist.
Reference: He has a cat because he is very fond of animals.
O-PBMT: He has a hangover because he is very fond of animals.
O-NMT: He has a cat because he is very fond of animals.
RBMT: He has a tomcat because it is very animal-dear.
OS-PBMT: He has a hangover because it is an encounter.
DFKI-NMT: He has a kater because he is very animal.
RWTH-NMT: He has a hangover because he’s very animal.
ED-NMT: He has a hangover because he is very animal-loving.
Test suite experiment – examples: phrasal verb

(2) Source: Warum haben die Mäuse mit den Streifen gekämpft?
Reference: Why did the mice fight over the stripes?

O-NMT: Warum haben die Mäuse mit den Streifen gekämpft?
ED-NMT: Warum haben die Mäuse mit den Streifen gekämpft?
DS-NMT: Why did the mice fight over the stripes?
ED-NMT: Why did the mice fight over the stripes?

Test suite experiment – examples: negation

(4) Source: Ich glaube, dass es auch nicht die amerikanische Position unterstützt.
Reference: I think that it does not support the American position either.

O-PBMT: [...] ich glaube, dass es auch nicht die amerikanische Position unterstützt.
O-NMT: [...] ich glaube, dass es auch nicht die amerikanische Position unterstützt.
PO-NMT: [...] ich glaube, dass es auch nicht die amerikanische Position unterstützt.
ED-NMT: [...] ich glaube, dass es auch nicht die amerikanische Position unterstützt.

Test suite experiment – examples: relative clause

(6) Source: Warum hörte Herr Muschler mit dem Streichen auf?
Reference: Why did Mr. Muschler stop painting?

O-PBMT: [... ] why did Mr. Muschler stop painting?
O-NMT: [... ] why did Mr. Muschler stop painting?
PO-NMT: [... ] why did Mr. Muschler stop painting?
ED-NMT: [... ] why did Mr. Muschler stop painting?

Test suite experiment – examples: MWE

(7) Source: Die Arbeiter müssen in den sauren Apfel beißen.
Reference: The workers would have to bite into the acid apple.

O-NMT: The workers would have to bite into the acid apple.
ED-NMT: The workers would have to bite into the acid apple.
D3.5: Quality Estimation Metrics and Analysis of 2\textsuperscript{nd} Annot. Round and Error Profiles

Automatic Evaluation

Evaluation with customer data

- Selection of frequent phenomena from customer data
- Moses-based SMT system trained in-house
- NMT from University Edinburgh (winning system in WMT'16)

Example: genitive nouns

Source: Schweißbänder erhöhen wesentlich den Tragekomfort eines Helmes.
Ref.: Sweatbands significantly increase the wearing comfort of a helmet.
NMT: Welding tapes significantly increase the comfort of a helmet.
Moses: Welding belts significantly increase the wearing comfort of a Helmes.

Example: modal verbs

Source: Dam kann das Board werkzeug-los gedreht und wieder eingehängt werden.
Ref.: The board can be turned and re-attached without using tools.
NMT: The board can be rotated and re-mounted.
Moses: To do this, the board can be rotated and back.

Automatic evaluation scores

<table>
<thead>
<tr>
<th></th>
<th>NMT</th>
<th>Moses</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>23.68</td>
<td>47.06</td>
</tr>
<tr>
<td>METEOR</td>
<td>20.46</td>
<td>38.20</td>
</tr>
</tbody>
</table>

Evaluation with customer data

<table>
<thead>
<tr>
<th></th>
<th>NMT</th>
<th>Moses</th>
</tr>
</thead>
<tbody>
<tr>
<td>formal address</td>
<td>138</td>
<td>90%</td>
</tr>
<tr>
<td>genitives</td>
<td>111</td>
<td>92%</td>
</tr>
<tr>
<td>modal construction</td>
<td>290</td>
<td>94%</td>
</tr>
<tr>
<td>negation</td>
<td>100</td>
<td>90%</td>
</tr>
<tr>
<td>passive voice</td>
<td>290</td>
<td>93%</td>
</tr>
<tr>
<td>predicate adjective</td>
<td>232</td>
<td>93%</td>
</tr>
<tr>
<td>propositional phrase</td>
<td>104</td>
<td>91%</td>
</tr>
<tr>
<td>terminology</td>
<td>330</td>
<td>95%</td>
</tr>
<tr>
<td>tagging</td>
<td>1.85</td>
<td>95%</td>
</tr>
<tr>
<td>a</td>
<td>15.5</td>
<td>89%</td>
</tr>
</tbody>
</table>
Useful for post-editing?

Small scale human evaluation

- 90 segments without specific terminology
- Divided into short, medium and long
- Selection: MT1, MT2, both, none

<table>
<thead>
<tr>
<th>Reference dependent</th>
<th>Reference independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>medium</td>
</tr>
<tr>
<td>NMT better</td>
<td>4</td>
</tr>
<tr>
<td>NMT worse</td>
<td>6</td>
</tr>
<tr>
<td>equal</td>
<td>good</td>
</tr>
</tbody>
</table>

Example: long segments

Source: Neben den Bedingungen zur Aufstellung und Inbetriebnahme wird eine Vielzahl von technischen und gesetzlichen Anforderungen an das Lager selbst gestellt, um z. B. wassergefährdende Flüssigkeiten, Säuren und Lagen oder auch entzündbare Flüssigkeiten gesetzskonform aufzubewahren und zu lagern.

NMT: In addition to the conditions for installation and commissioning, a wide range of technical and legal requirements will be placed on the warehouse itself in order to maintain and store, for example, water-hazardous liquids, acids and foliage, or even flammable liquids.

Lessons learnt – as an LSP

- NMT is not flawless either but can capture information beyond the scope of SMT
- Keep an eye on NMT developments!
  - Terminology
  - Tag-handling
  - Domain adaption
  - ...
- Prepare post-editors for different error types!!!

Thank you!

Your opinion is important to us! Please tell us what you thought of the lecture. We look forward to your feedback on smartphone or tablet under http://lt04.honestly.de or scan the QR code:
The feedback tool will be available even after the conference!
TQ-AutoTest – An Automated Test Suite for (Machine) Translation Quality

Vivien Macketanz, Renlong Ai and Aljoscha Burchardt
DFKI – Language Technology Lab
Berlin, Germany
{firstname.lastname}@dfki.de

Abstract

In several areas of NLP evaluation, test suites have been used to analyse the strengths and weaknesses of systems. Today, Machine Translation (MT) quality is usually assessed by shallow automatic comparisons of MT outputs with reference corpora resulting in a number. Especially the trend towards neural MT has renewed peoples’ interest in better and more analytical diagnostic methods for MT quality. In this paper we presented TQ-AutoTest, a novel framework that supports a linguistic evaluation of (machine) translations using test suites. Our current test suites comprise about 5000 handcrafted test items for the language pair German–English. The framework supports the creation of tests and the semi-automatic evaluation of the MT results using regular expressions. The expressions help to classify the results as correct, incorrect or as requiring a manual check. The approach can easily be extended to other NLP tasks where test suites can be used such as evaluating (one-shot) dialogue systems.

Keywords: Machine Translation, Quality Evaluation, Test Suites

1. Introduction and Background

In several areas of NLP evaluation, test suites have been used to analyse the strengths and weaknesses of systems. In contrast to “real-life” gold standard corpora, test suites can contain made-up or edited input-output pairs to isolate interesting or difficult phenomena.

In Machine Translation (MT) research, broadly-defined test suites have not been used apart from several singular attempts (King and Falkedal, 1990; Ishahara, 1995; Koh et al., 2001, etc.). One of the reasons for this might be the fear that the performance of statistical MT systems depends so much on the particular input data, parameter settings, etc., that relevant conclusions about the errors they make are difficult to obtain. Another concern is that “correct” MT output cannot be specified in the same way as the output of other language processing tasks like parsing or fact extraction where the expected results can be more or less clearly defined. Due to the variation of language, ambiguity, etc., checking and evaluating MT output can be almost as difficult as the translation itself.

Today, MT quality is usually assessed by shallow automatic comparisons of MT outputs with reference corpora resulting in a number. In narrow domains, however, researchers have started to explore the differences between systems and between the development stages of one system in more linguistic detail. Especially the trend towards neural MT has renewed peoples’ interest in better and more analytical diagnostic methods for MT quality. Recent work based on specific test suites includes the study of verb–particle constructions (Schottmüller and Nivre, 2014), pronouns (Guil- lou and Hardmeier, 2016) or structural divergences (Isabelle et al., 2017). (Bentivogli et al., 2016) performed a comparison of neural- with phrase-based MT systems on IWSLT data using a coarse-grained error typology where neural systems have been found to make fewer morphological, lexical and word-order errors.

Using our own test suites, we have performed several comparative studies of different MT systems both in the general domain (Burchardt et al., 2017) and in the technical domain (Beyer et al., 2017). When presenting this work, one of the most (obvious) criticism we got was the huge amount of manual effort that was involved in the evaluation procedure. In this paper we will present the novel TQ-AutoTest framework that supports the evaluation procedure. Finally, in Section 4 we will conclude and give an outlook on future work.

2. Test Suites for English–German

We have built a test suite for a fine-grained evaluation of MT quality for the language pair English – German. In brief, it contains segments selected from various parallel corpora and drawn from other sources such as grammatical resources, e.g., the TSNLP Grammar Test Suite (Lehmann et al., 1996) and online lists of typical translation errors. Each test sentence is annotated with a phenomenon category and the phenomenon it represents. An example showing these fields can be seen in Table 1 with the first column containing the source segment and the second and third column containing the phenomenon category and the phenomenon, respectively. The fourth column shows an example machine translation1 and the last column contains a post-edit of the MT output that is created by making as few changes as possible.

In our latest version of the test suite, we have a collection of about 5,000 segments per language direction that are classified in about 15 categories (most of them similar in both language directions) and about 120 phenomena (many of...
them similar but also some differing, as they are language-specific). Depending on the nature of the phenomenon, each is represented by at least 20 test segments in order to guarantee for a balanced test set. The categories cover a wide range of different grammatical aspects that might or might not lead to translation difficulties for a MT system.

2.1. Manual Evaluation Procedure
In order to evaluate a system’s performance on the categories in the test suite, we concentrate solely on the phenomenon in the respective sentence and disregard other errors. This means that we have to determine whether a translation error is linked to the phenomenon under examination or if it is independent from the phenomenon. If the former is the case, the segment will be validated as incorrect. If, however, the error in the translation can not be traced back to the phenomenon, the segment will be counted as correct.

When conducting the manual evaluation, the system outputs were automatically being compared to a “reference translation”, which is, in fact, the post-edit of the Google Translate output, as those were the very first translations to be generated and evaluated when we started building the test suite. In a second step, all the translations that do not match the “reference” were manually evaluated by a professional linguist since the translations might be very different from the Google post-edit but nevertheless correct. This is also the reason why we refrained from creating an independent reference. As a consequence, we cannot compute automatic scores like BLEU. We do not see this as a disadvantage as with the test suite we want to focus rather on gaining insights about the nature of translations than on how well translations match a certain reference.

Nevertheless, this manual evaluation is a very time-consuming process, so we decided to come up with a semi-automatic solution, i.e., the TQ-AutoTest.

3. The TQ-AutoTest Framework
With the test suite growing bigger over time, we decided to implement a framework that facilitates the evaluation procedure by automating the analysis. Therefore, we built the TQ-AutoTest. In order to include as many correct translations options as possible, the TQ-AutoTest is based on regular expressions (cf. Section 3.1.). Currently, the automation is almost fully completed for the language direction German – English and we are working on expanding and completing the other language direction. We plan to have both language directions finalized by the end of the project QT21, thus, January 2018.

Presently, the TQ-AutoTest exhibits the following features (described in further detail in the following Section): data preparation; upload report; view report; compare engines; regular expression evaluation; expand, edit and query database (cf. Figures 2 and 3).

With these functions, the TQ-AutoTest can be used for different purposes: You can not only test a system’s performance with regard to the linguistic phenomena but also compare the performance of different systems/system types or track changes within one system’s performance. By doing so, you can test the system(s) either on all phenomena, or a just selection of the phenomena. To prevent overfitting or cheating, we will not publish the test items. Before sending them to colleagues who want their engines tested, we use a mechanism for scrambling the test segments with a large amount of “distractor” segments.

3.1. Regular Expressions
The foundation of the evaluation with the TQ-AutoTest are regular expressions. With the help of these patterns, we try to cover as many correct translations as possible. In-line with our manual evaluation procedure briefly described in Section 2.1., the regular expressions only focus on the part of the segment that is under investigation, i.e., the respective phenomenon. Since all other mistranslations that can not be related to the phenomenon are ignored, it is not necessary for the regular expressions to cover the whole sentence.

The process of creating the regular expressions was thus very complex and elaborate. Considering that once the corpus is completed it can be used over and over again, we are convinced it is worthwhile investing the time and effort to create the regular expressions. We did not only create positive regular expressions with which the MT output can be evaluated as correct, but in some cases also negative regular expressions with which the MT output is evaluated as incorrect.

The German source sentence in example (1) contains a lexical ambiguity: The German word Mann can either mean man or husband. In combination with a possessive pronoun (in this case ihr - her), Mann always refers to husband. Output 1 - 3 are exemplary MT outputs. As can be seen, only output 1 matches the regular expression. The regular expression also allows translations that include the words spouse, hubby or hubbies. Output 2 on the other hand matches the negative regular expression and thus would be evaluated as incorrect. Output 3 does not match any of the regular expressions and therefore would be reconsidered in a follow-up manual check (cf. Section 3.2.). A screenshot of a positive match with a regular expression in the TQ-AutoTest can be seen in Figure 1.

Figure 1: Screenshot: Positive Match with RegEx.

In addition to the regular expressions, we also implemented a feature for positive and negative tokens. With this feature, a whole sentence (i.e., a MT output) can be added to the database to be matched against in subsequent evaluations. This feature is very convenient for phenomena or segments that are more complex. As a consequence, the database is constantly expanding and covers and increasing amount of possible MT outputs.
### 3.2. Workflow

An exemplary workflow in the TQ-AutoTest looks as follows (not all steps must necessarily be realized):

**Data Preparation** An absolute or relative number of sentences from all categories/a selection of categories, resp. from all phenomena/a selection of phenomena is selected randomly (cf. Figure 2) and then scrambled with a random selection of the distractors, whereby the scramble factor can be selected manually. The resulting data is generated in a text file with an ID.

This text file can then be used for running the translations, e.g., on different types of MT systems, say a phrase-based and a neural MT system, or on different version of one system, e.g., before and after some expected improvement.

**Upload Report** Once the translations are generated (the order of the sentences must be maintained), a text file with the outputs can be uploaded. With the upload, information about the engine (e.g., Google), the type of engine (e.g., NMT) and further comments must/can be entered, cf. Figure 3.

The test sentences are then automatically unscrambled from the distractors and the sentences are evaluated based on the database of regular expressions and tokens.

**View Reports** In this tab, all reports that have been generated can be viewed and edited. Edited means in this case that sentences that did not match any of the regular expressions or tokens need to be double-checked manually. Correct outputs are shaded in green, incorrect outputs in red and outputs that need to be determined are shaded in yellow. If desired, the evaluation of the manual checking can be added to the database ("Apply Tokens", if it should not be added "Skip Tokens"), cf. Figure 4.

Furthermore, a statistic about the amount of correct/incorrect/tbd translations is automatically generated. This statistic contains tables as well as graphs, both of which can be exported.

**Compare Engines** This function allows for a comparison of different MT systems/system types that generated translations for the same (sub)set of sentences. Hereby, the absolute and relative numbers of correct/incorrect/tbd translations per systems are calculated on the phenomena, categories and on average, and are displayed in tables as well as graphs, both of which can be exported as well. An exported graph with three exemplary MT systems can be found in Figure 5. The comparison shows the relative number of correct translation per systems on the categories.

We have started first test that will be reported in the full paper.
3.3. Implementation
The web interface is implemented using Play Framework, which is open-source, reactive, flexible, and provides TypeScript so that both research and commercial requirements are supported. The front-end uses bootstrap library to ensure compatibility across browsers and platforms, and the back-end is implemented with Scala. Both templates and test results are stored in Mysql database. The software can be downloaded at https://gitlab.com/QT21/QT21-Resources/tree/master/Tools/TQ-AutoTest. Note that the test items themselves are not part of the available resources.

4. Conclusion and Outlook
In this paper we have presented TQ-AutoTest, a framework that supports the analytical evaluation of (machine) translations using test suites. Our current test suites comprise about 5000 test items for the language pair German-English in both directions. The framework supports the creation of tests and the evaluation of the translation results using regular expressions. The expressions classify the results as correct, incorrect or requiring a manual check. After finalizing the regular expressions, we will conduct more tests of the tool. Looking into the future, the approach allows for a number of extensions. Obvious possibilities are more languages and including also domain-specific test suites. Both will be manual work, but given the experience and example we have created will hopefully speed up the process.

It also imaginable to extend the approach to other NLP applications such as dialogue (Chatbots). We have a concrete request by an industry partner to explore the possibility of evaluating meeting translations that we are currently pursuing.

5. Acknowledgements
Will be added for camera ready.

6. Bibliographical References


Figure 5: Screenshot: Compared Engines.
F  QT21 data and the TAUS Quality Dashboard

QT21 and the TAUS Quality Dashboard

Report

Introduction to the TAUS Quality Dashboard

DQF-MQM is one of the pillars of the TAUS Quality Dashboard and the DQF framework. The DQF Framework is implemented by a growing range of CAT tools as a plugin with which it can communicate with the TAUS DQF database.

The data sent to the DQF database concern translation data for production statistics, and review data for translation quality statistics. The review data can be divided into corrections and error annotations. Corrections are improvements made by a reviewer on the work of a (typically human) translator, with the purpose of reviewing the job of the translator, and at the same time to make sure the final publication of the translation has the intended quality level.

In this scenario, error annotations have the purpose of reviewing the translator and to give feedback. In fact, these tasks of correction and error annotation are often combined in a single review task. During error annotation a reviewer can select segments or parts of segments, and attach error annotations to this selected text, that indicate an error category and an error severity.

The categories for the error annotation tasks and the severities are defined by the DQF-MQM model. The DQF Framework user however is free to focus on a subset of the DQF-MQM error typologies. As in the case of the QT21 annotations, errors can be annotated for only the main categories, instead of having all the more granular secondary levels.

The CAT tool plugins provide both the interface to perform the above tasks (in as far as the CAT tool itself does not provide this interface), and the communication with the DQF API, which is connected to the DQF database. Through a single API, the DQF Framework ensures a uniform way of measuring, and makes it possible to compare translation and review data, even though they originate from different CAT tools, workflows and machine translation engines.

Buyers, translators, reviewers and other translation providers all working on the same project can see the statistics about their projects. The main front end for showing statistics is the TAUS Quality Dashboard. As the DQF Framework is growing in functionality, a data connector is added. This data connector provides users with an API and returns statistics in JSON format, so that new charts and custom dashboards can be programmed with these returned JSON formatted data for their input.

Currently the TAUS Quality Dashboard is the main place where users can see results of correction tasks and error annotations. The Quality Dashboard has charts on multiple levels. The first is on project level, which shows statistics for a single project. Then it offers levels in which numbers are
aggregated on user, company, and industry level. This makes it possible to benchmark the user’s own data or the company’s data against the data aggregated over the complete database (‘industry’ level).

**A Typical translation workflow in the Quality Dashboard**

As explained in the previous paragraph, the TAUS Quality Dashboard is developed for, and used in translation production workflows. In these workflows, the first concern is translation, and review tasks are only needed whenever there is a special demand for high quality translation such that extra review rounds are justified. Other workflows in which review tasks might be needed are pilot projects, where estimations of the quality of resources (human translators, or automated translation sources) are requested.

The general and typical workflow here is that a translation task is created, in which a text is (partially) pre-translated by machine translation, translation memory, or a combination of both, and this pre-translation is edited by a human translator. The TAUS Quality Dashboard shows statistics on this part of this type of translation job, but these statistics are mainly productivity numbers that measure time for translation and are related to the volume of the translated text.

After the translation phase it is possible to have a review phase. This review can consist of either correction, or error annotation, or a combination of both. Correction in this context is the process in which the translation is evaluated and corrected if needed in order to get a better translation as the end result. Error annotation is the evaluation of the translation, but instead of correcting the translation, words and segments in the translation are annotated with errors. Whenever an error occurs, the annotation characterizes the error in terms of the DQF/MQM typology, and in severity (‘neutral’, ‘minor’, ‘major’, and ‘critical’).

This typical workflow is different from the primary objectives of the work performed in QT21, which is evaluating the quality of machine translation output for five language pairs in two domains. Because of this, each segment in the QT21 research project has been corrected, which is an improbable outcome for review projects on the Quality Dashboard. Also 4000 segments per language pair were manually annotated by professional translators, and this is just as improbable for the Quality Dashboard.

**Using QD calculation on the QT21 dataset**

In order to showcase the dashboard’s profiling capacities, we have taken the data sets presented in QT21 Deliverable D3.4 (and presented at MT Summit in Specia et al. 2017) and imported them into the dashboard.

The charts on the TAUS Quality Dashboard for project related statistics are divided in three categories: productivity, correction and error annotation.

The **charts on corrections** currently show these statistics:
The fraction of the segments in the total translation that are corrected (disregarding the number of corrections),

The number of corrections on each 100 characters of translated text. This is called the 'correction density'. It takes the number of corrections according to the Levenshtein Distance and divides this by the total of characters in the 'target' text multiplied by 100.

By the nature of the QT21 research data, only the second of these charts has informative value (the first of these charts is much less relevant given almost all segments were post-edited).

The Correction Density chart, just like any of the other charts on the Quality Dashboard, has many ways of filtering the data in the chart. For example, data might be filtered for certain vendors within the project, or certain language combinations in the project. Likewise, the data in the chart can be grouped, which means that the chart shows multiple columns, each filtered by one of the values in the grouping property. The following chart is the Correction Density in the QT21 data set, where it is grouped by the language combination (and SMT versus NMT in the combination English to Latvian and English to German).

The charts on error annotations on the Quality Dashboard currently show these statistics:

- The fraction of the segments that has one or more error annotations;
- Number of error annotations per 1000 (translated) words, called 'Error Density'. This takes the number of target words multiplied by 1000 as denominator, and the number of error annotations as the numerator.
- Number of weighted errors per 1000 (translated) words, called 'Weighted Error Density'. This is the same calculation as the previous, but errors can be given different weights, based on the severity of the annotated error. This weight is applied with controls on the Quality Dashboard.
- Number of errors for each severity

Here also only the second chart has informative value.

The following charts show the Error Density in the research data. In order to fit the QT21 research data, the mapping defined in QT21 Deliverable D3.1 was applied, because of the higher granularity of the research data. This means that some of the error categories were mapped to their 'parent' category. The following error categories were mapped to the 'Grammar' error category:

- Function words,
- Extraneous
- Incorrect
- Missing
- Word form
- Agreement
- Part of speech
- Tense/aspect/mood
- Word order

‘Unintelligible’ was mapped to ‘Fluency’.

The data in the following chart are grouped by a combination of language combination and reviewer. With four language combinations (including the SMT and NMT variances for English to Latvian), and two reviewers, this generates eight columns.

The Quality Dashboard also has the possibility to filter of group by error category as well.

In the following charts, a grouping by error category is applied, while a filter on both language combination and reviewer is applied.

Filtered on English to Czech, Reviewer 1, grouped by error category:

Filtered on English to Czech, Reviewer 2, grouped by error category:
Filtered on German to English, Reviewer 2, grouped by error category:

Filtered on English to German, SMT, Reviewer 1, grouped by error category:

Filtered on English to German, SMT, Reviewer 2, grouped by error category:

Filtered on English to German, NMT, Reviewer 1, grouped by error category:

Filtered on English to German, NMT, Reviewer 2, grouped by error category:
Feature-Rich NMT and SMT Post-Edited Corpora for Productivity and Evaluation Tasks with a Subset of MQM-Annotated Data

Kim Harris, Lucia Specia, Katrin Marheinecke, Aljoscha Burchardt, MT Summit, September 2017 (Nagoya)

Paper Overview

Background

• Why QT21? – Project goals
• Data collection
• Resources and processes

Results

• SMT analysis
• NMT analysis
• Comparing SMT and NMT output

Potential data uses

• Uses
• Challenges

Conclusions / Future work

Katrin Marheinecke – MT Summit 2017 – Nagoya (Japan)
Situation at the beginning of 2015

- Language coverage is a key obstacle impeding the free flow of people, information and trade in the European Digital Single Market.

- Many languages that are not well enough supported by current technologies show common traits:
  - morphologically complex,
  - with free and diverse word order,
  - not enough training resources and/or processing tools available.

Why QT21?

Goals of QT21

- To substantially improve statistical and machine-learning based translation models for challenging languages and under-resourced scenarios,
- To improve evaluation and continuous learning from mistakes, guided by a systematic analysis of quality barriers, informed by human translators.
- To continuously measure progress, and to provide a platform for sharing and collaboration the project revolves around a series of Shared Tasks (co-organized with WMT).
Data collection

Data collection in QT21 is an unprecedented large-scale collaboration between MT R&D and translation experts.

It contains:
- 20,000–45,000 sentences of industry-generated content in the IT and life sciences domains.
- Post-edited and annotated industry data for four morphologically rich language pairs (EN-DE, EN-CS, DE-EN, EN-LV), performed by professional translators.
- A subset of “almost perfect” sentences with MQM error annotations for further analysis and profiling of recurring error patterns.
- A combination of automatic time and keystroke logging, and quality assessment by the translators.

The type of data is significant because...

- Industry data is generally not available to the public for reasons of IPR, thus restricting the type of data used by research to domains such as news.
- Industry data is efficiently written according to guidelines and style guides, using standardized terminology and grammatical constructs for reasons of clarity and consistency and cost (unlike journalistic or creative texts).
- Therefore allows patterns in MT output to be detected and analyzed based on a variety of criteria.
- Both research and industry can use the data for quality estimation (QE) and automated post-editing purposes.
Unique resource selection phase

Selection process
• Professional bilingual translators with post-editing experience
• Bilingual post-editing and annotation (as opposed to monolingual evaluation)

Training of resources
• Training sessions (web, phone or Skype or on site) for all translators
• Wiki accessible to all participants containing pertinent information
• For the annotation task, a 20-minute training video was prepared
• Training materials are publicly available at http://qt21-wiki.dfki.de/index.php?title=Main_Page

2-step workflow (for technical reasons)
• Post-editing
• Error annotation

Post-Editing Process

While post-editing the following info was logged
• Editing time: time spent translating or editing a unit.
• Keystrokes: number and types of keys pressed during the PE.
• Actual edits: words inserted, deleted, moved, etc.
• Evaluation: quality score based on a pre-defined criterion.

After every post-edit the following question appeared:
• How good was the raw MT?
  ➢ 1. Perfect or near perfect (typographical errors only).
  ➢ 2. Very good, could be post-edited quickly.
  ➢ 3. Poor, required significant post-editing.
  ➢ 4. Very poor, required retranslation.
The objective of the annotation process was to determine whether or not specific error categories and patterns existed in the MT output and whether or not these could be used to systematically improve the results.

- After post-editing, 2000 sentences (per language pair and system) ranked as very good were selected randomly for error annotation step
- Translators/annotators were asked to flag and classify all errors that had been changed during PE step
- Annotation had to be performed according to MQM error categories
- 200 sentences were annotated by a second annotator to establish where divergence/agreement was present and why (WIP)

Tools and methods used

- PET Tool
  - Open-source Java-based tool for post-editing and assessing translations while gathering statistics on several effort indicators

- Translate 5 annotation environment
  - Open-source online tool for QA and error annotation.

- Multidimensional Quality Metrics (MQM)
  - Annotation based on industry-wide accepted error categorization standard
Decision Tree for MQM

Decision Tree for MQM
Post-Editing: Findings and Results

**Average post-edit time and number of keystrokes**

<table>
<thead>
<tr>
<th>Lang.</th>
<th>Avg. PE time</th>
<th>Avg. keystrokes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PBMT</td>
<td>NMT</td>
</tr>
<tr>
<td>DE–EN</td>
<td>36±45</td>
<td>–</td>
</tr>
<tr>
<td>EN–DE</td>
<td>46±39</td>
<td>32±37</td>
</tr>
<tr>
<td>EN–LV</td>
<td>23±28</td>
<td>36±39</td>
</tr>
<tr>
<td>EN–CS</td>
<td>42±35</td>
<td>–</td>
</tr>
</tbody>
</table>

EN–DE: Greater correction effort required for SMT, but disproportionate to the amount of time required for sentences with fewer errors according to first analyses

EN–LV: SMT system better trained, complexity of language and lack of training data indicative of results

---

**Annotation: MQM errors per language pair**

<table>
<thead>
<tr>
<th>Error type</th>
<th>DE–EN</th>
<th>EN–DE</th>
<th>EN–LV</th>
<th>EN–CS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PBMT</td>
<td>NMT</td>
<td>PBMT</td>
<td>NMT</td>
</tr>
<tr>
<td>Accuracy</td>
<td>3</td>
<td>0</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>Additive</td>
<td>59</td>
<td>52</td>
<td>156</td>
<td>158</td>
</tr>
<tr>
<td>Mistranslation</td>
<td>437</td>
<td>862</td>
<td>532</td>
<td>952</td>
</tr>
<tr>
<td>Omission</td>
<td>576</td>
<td>980</td>
<td>355</td>
<td>966</td>
</tr>
<tr>
<td>Untranslated</td>
<td>278</td>
<td>102</td>
<td>24</td>
<td>79</td>
</tr>
<tr>
<td>Fluency</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>233</td>
</tr>
<tr>
<td>Grammar</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Function word</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Extraene</td>
<td>302</td>
<td>525</td>
<td>245</td>
<td>49</td>
</tr>
<tr>
<td>Incorrect</td>
<td>139</td>
<td>806</td>
<td>449</td>
<td>56</td>
</tr>
<tr>
<td>Missing</td>
<td>362</td>
<td>779</td>
<td>231</td>
<td>88</td>
</tr>
<tr>
<td>Word form</td>
<td>0</td>
<td>94</td>
<td>267</td>
<td>280</td>
</tr>
<tr>
<td>Part of speech</td>
<td>20</td>
<td>126</td>
<td>152</td>
<td>38</td>
</tr>
<tr>
<td>Agreement</td>
<td>18</td>
<td>506</td>
<td>97</td>
<td>419</td>
</tr>
<tr>
<td>Time: speed</td>
<td>63</td>
<td>104</td>
<td>51</td>
<td>60</td>
</tr>
<tr>
<td>Word order</td>
<td>218</td>
<td>868</td>
<td>309</td>
<td>136</td>
</tr>
<tr>
<td>Spelling</td>
<td>118</td>
<td>126</td>
<td>152</td>
<td>324</td>
</tr>
<tr>
<td>Typography</td>
<td>282</td>
<td>555</td>
<td>249</td>
<td>823</td>
</tr>
<tr>
<td>Untranslatable</td>
<td>0</td>
<td>55</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Terminology</td>
<td>27</td>
<td>82</td>
<td>139</td>
<td>34</td>
</tr>
</tbody>
</table>

**EN–DE**

- Function words largest error category by far
- Agreement and word order major error categories for SMT
- Mistranslation and omissions major error categories for NMT

**Percentage of mistranslations twice as high in NMT as SMT**
Collected data points

The recording of this information during the post-editing phase allows for specific features and novel combinations of features to be used for a variety of research- and user-oriented purposes

- Determining the actual post-editing effort by translators based on time and keystrokes
- Comparing these results to the perceived level of quality of the post-edited sentence
- Establishing correlations between certain characteristics such as sentence length and post-edit time, or post-edit time and human quality evaluation

Comparison SMT and NMT (EN-DE)

Flexible "pretty print" environment displays comparative views of systems as well as a sort function according to error type

SMT:

<table>
<thead>
<tr>
<th>Error Type</th>
<th>SMT Example</th>
<th>SMT Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Type</td>
<td>SMT Example 1</td>
<td>SMT Example 2</td>
</tr>
<tr>
<td>Error Type</td>
<td>SMT Example 3</td>
<td>SMT Example 4</td>
</tr>
</tbody>
</table>

NMT:

<table>
<thead>
<tr>
<th>Error Type</th>
<th>NMT Example</th>
<th>NMT Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Type</td>
<td>NMT Example 1</td>
<td>NMT Example 2</td>
</tr>
<tr>
<td>Error Type</td>
<td>NMT Example 3</td>
<td>NMT Example 4</td>
</tr>
</tbody>
</table>

Katrin Marheinecke – MT Summit 2017 – Nagoya (Japan)
### SMT vs NMT

**SMT**
- Significant fluency issues
- BUT leaves unknown words untranslated, systematically omits specific sentence structures
  - Easier to recognize error patterns
- Perception pessimistic
  - More errors but less time required to fix them

**NMT**
- Excellent fluency results
- BUT translates words incorrectly on character basis and omits random words
  - Increased cognitive effort because errors are less obvious
- Perception optimistic
  - Fewer errors despite more (or just as much) time required to fix them

*Predictability of errors reduces cognitive effort despite higher number.*

*Fluency is a hindrance when errors are unpredictable.*

---

### SMT vs. NMT

<table>
<thead>
<tr>
<th>Source</th>
<th>A dim check mark indicates that the condition is applied only to part of the selection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMT</td>
<td>Ein ... Häkchen bedeutet, dass die Bedingung nur auf einen Teil der Auswahl angewendet wird. (omission)</td>
</tr>
<tr>
<td>SMT</td>
<td>Eine unscharfe Häkchen gibt an, dass die Bedingung nur auf einen Teil der Auswahl angewendet wird. (mistranslation)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>The export engine looks for logical partitions where a clear line of sight can be placed between objects or groups of objects.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMT</td>
<td>Die Export-Engine sucht nach logischen ..., wo eine klare Sichtlinie zwischen Objekten oder Objektgruppen platziert werden kann. (omission)</td>
</tr>
<tr>
<td>SMT</td>
<td>Die Export-Engine sucht logische Partitionen, in denen eine klare Sichtweite zwischen Objekten oder Gruppen von Objekten platziert werden können. (agreement due to distance)</td>
</tr>
</tbody>
</table>
SMT vs. NMT errors

Source: You can also perform a more complicated table sort based on the contents of two columns.
NMT: Sie können auch eine komplexere Tabelle anhand des Inhalts zweier Spalten durchführen. (omission/mistranslation)
SMT: Sie können auch eine kompliziertere Tabelle sortieren basierend auf den Inhalt von zwei Spalten durchführen. (mistranslation/POS)

Source: You can change owners for a draft that has been sent for review, and you can also remove any draft from the draft review process.
NMT: Sie können die Inhaber eines Entwurfs ändern, der zur Überprüfung gesendet wurde, und Sie können auch jeden Entwurf aus der Entwurfskonsole entfernen. (fluent mistranslation)
SMT: Sie können den Inhaber für einen Entwurf ändern, die zur Überprüfung gesendet wurde, und Sie können auch einen beliebigen Entwurf aus dem Entwurfsüberprüfungsprozess zu entfernen. (agreement)

Typical NMT errors

Random omissions, mistranslations while maintaining fluency

Source: If the website has no administrator, click Yes when a dialog box asks whether you want to become the website administrator.
NMT: Wenn die Website keinen Administrator hat, klicken Sie auf "Ja," wenn Sie in einem Dialogfeld gefragt werden, ob Sie den Website-Administrator bestimmen möchten. (fluent mistranslation)

Source: Alternate glyphs for OpenType or Asian fonts such as Tekton Pro MM appear in the Glyphs panel.
NMT: Alternative Glyphen für OpenType-Schriftarten oder asiatische Schriftarten wie "Karikon Pro MM" werden im Glyphenbedienfeld angezeigt. (random mistranslation)
Quality Translation 21
D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

1/10/18

Katrin Marheinecke – MT Summit 2017 – Nagoya (Japan)

Typical SMT errors

Source
Converting all layers in the FreeHand document to a single flattened layer in the document.

SMT
Konvertiert alle Ebenen des FreeHand-Dokuments in einer einzelnen Ebene im Dokument reduziert. (incorrect function word)

Source
These files are encoded as UTF-8 or ASCII, which is a subset of UTF-8.

SMT
Diese Dateien werden als UTF-8 oder ASCII, bei der es sich um eine Untergruppe von UTF-8 kodiert. (mistranslation, but acceptable synonym)

Potential data uses

Combine data points to demonstrate various hypothesis within and against each system, for example:

• cognitive effort (time, keystrokes, sentence length)
• perceived effort (time, keystrokes, quality score)

Benchmark automatic scores such as BLEU, METEOR, TER against quality scores, keystrokes and time in any combination

Demonstrate improvement approaches such as:

• Automatic pre-editing of source
  ➢ Based on error types
  ➢ Based on sentence length
• Authoring guidelines for source
• Automatic post-editing of target

Benchmark results of proprietary user QE tools against this data set to evaluate accuracy using various data points

Katrin Marheinecke – MT Summit 2017 – Nagoya (Japan)
Challenges

Currently no ability to apply empirical methods to data analysis

• Means of flexibly combining various data points to establish research paths for both users and the research community seriously lacking
• In-depth analysis of systematic SMT errors for progressive improvement in remedial actions such as APE must be done in order to use these results
• Unpredictability of NMT output still influences effort (speed, cognitive) with no accountability for this in automatic scores
• Systematic post-editing guidelines for NMT difficult to sufficiently generalize

Findings & Conclusions

Classifying and analyzing errors and error types has shown that:
• SMT errors are repetitive
• NMT errors are less predictable

Sentence length can greatly influence productivity and quality

• One sentence that is three times as long as another took five times as long to post-edit and contained more errors than three shorter sentences.

Current NMT output does not significantly beat SMT in PE productivity

• Cognitive effort higher, requiring more time to analyze, even for shorter sentences, when system type is known
• Longer sentences in NMT contain less obvious errors because they handle distance better than SMT
  ➢ Risk of missing errors greater as a result [the more information the post-editor is given about the system, the better the PE results].
Future work

- Additional annotations for each language pair in each system
  - Higher accuracy for error categories
  - Study recurring error types for automatic pre- and post-editing
  - Inter-annotator analysis for improved annotation

Empirical analysis of the data collected
- Develop/find tool that can combine data points and extract data for certain experiments, tests and benchmarking purposes
- Continue in-depth analysis of effort in both system types
- Establish patterns in relation to effort for further analysis

Apply results to other industry data
- Find users interested in piloting benchmarking activities with the data

Integrate results in other projects involving
- QE, APE and pre-editing

References

- Translate 5 Online Editor: https://www.translate5.net/
Thank you!

katrin_marheinecke@textform.com
kim_harris@textform.com
lspecia@sheffield.ac.uk
aljoscha.burchardt@dfki.de
MaxSD: A Neural Machine Translation Evaluation Metric Optimized by Maximizing Similarity Distance

Qingsong Ma1,2 Fandong Meng1,2 Daqi Zheng1,2 Mingxuan Wang1,2 Yvette Graham3 Wenbin Jiang1,2 Qun Liu1,3

1 Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
2 University of Chinese Academy of Sciences
3 ADAPT Centre, School of Computing, Dublin City University, Ireland

{maqingsong, mengfandong, zhengdaqi, wangmingxuan, jiangwenbin}@ict.ac.cn qun.liu@dcu.ie graham.yvette@gmail.com

Abstract. We propose a novel metric for machine translation evaluation based on neural networks. In the training phase, we maximize the distance between the similarity scores of high and low-quality hypotheses. Then, the trained neural network is used to evaluate the new hypotheses in the testing phase. The proposed metric can efficiently incorporate lexical and syntactic metrics as features in the network and thus is able to capture different levels of linguistic information. Experiments on WMT-14 show state-of-the-art performance is achieved in two out of five language pairs on the system-level and one on the segment-level. Comparative results are also achieved in the remaining language pairs.

Keywords: machine translation evaluation, neural networks, similarity distance, maximization

1 Introduction

With the development of machine translation (MT), MT evaluation (MTE) has received increasing attention. Traditional lexical-based metrics such as BLEU [8], Meteor [3], and TERp [11] take n-grams, synonyms, stems, word order, and phrases into account. However, metrics based on lexical and syntactic information are insufficient to evaluate the quality of the hypotheses, due to mismatch errors caused by limited synonyms and references.

Recently, semantic-based metrics have become more feasible with the help of deep learning. This paper presents an effective metric based on neural networks, i.e. Bidirectional Long Short Term Memory (Bi-LSTM) network [7, 10] for MTE. To capture the inner connection between hypotheses and references, we also explore the effect of an enhanced Bidirectional Combined LSTM (BiC-LSTM) network, which takes the concatenation of the hypothesis and the reference as
Generally, the goal of the framework is to predict quality scores of hypotheses, which requires references and hypotheses together with quality scores as training examples. However, the difficulty of obtaining hypotheses with quality scores leads to the insufficiency of training examples. For instance, ReVal [6] devotes extra effort to compute quality scores of hypotheses, producing less than 15 thousand training examples from the human judgement file of WMT-13 [1], and subsequently requires extra resources to enlarge the training set. As the amount of training examples is crucial to network performance, we design a new objective during the training process, which maximizes the distance between two similarity scores: one between the reference \textit{ref} and a high-quality hypothesis \textit{posh}, and the other one between \textit{ref} and a low-quality one \textit{negh}. Thus, two hypotheses, as well as the reference comprise a training example, which allows us to extract adequate training examples from WMT human judgements. Furthermore, for testing, the network takes only one hypothesis and one reference as an input, then outputs an evaluation score of the hypothesis. Compared with Guzmán et al. (2015), our metric significantly reduces complexity in this respect, as we can evaluate with a single hypothesis, while they require a pairwise setting. Experiments on WMT-14 show that state-of-the-art performance is achieved in two out of five language pairs on the system-level and one on the segment-level, comparative results are obtained for remaining language pairs.

Fig. 1: The overall architecture of the maxSD model. Bi(C)-LSTM means either Bi-LSTM or BiC-LSTM network. Bi-LSTM network takes the left side of '/' as input, while BiC-LSTM the right one. The Bi-LSTM or BiC-LSTM network produces the representation of each input, which are used to compute \textit{simPn} and \textit{simNn}. \textit{simP} and \textit{simN} are computed by incorporating 5 metric scores, namely \textit{spr} and \textit{snr} respectively. The objective of the architecture is to maximize the distance between \textit{simP} and \textit{simN} are.
2 Learning Task

The goal of the training process in our neural network is to maximize the distance of the similarity score between \( \text{ref} \) and \( \text{posh} \), and the other one between \( \text{ref} \) and \( \text{negh} \). In the testing process, we evaluate the quality of \( \text{hyp} \) given \( \text{ref} \) by computing the similarity score between them.

Thus, the input of our neural network is a tuple, marked as \((\text{ref}, \text{posh}, \text{negh})\). The loss function of the neural network is formulated as follows:

\[
J_{\theta} = -\sum_{n} \max(0, \text{sim}_P - \text{sim}_N) \tag{1}
\]

where \( \text{sim}_P \) is the similarity score between \( \text{ref} \) and \( \text{posh} \), and \( \text{sim}_N \) is that between \( \text{ref} \) and \( \text{negh} \). A more detailed computation is illustrated below.

3 MaxSD Model: Maximizing Similarity Distance Model

3.1 MaxSD Model

In order to learn the similarity scores, \( \text{sim}_P \) and \( \text{sim}_N \), we build a maxSD model. We explore two versions of MaxSD model, the performance of two LSMT networks, namely Bi-LSTM and BiC-LSTM. As showed in Figure 1, we first obtain the continuous space representations of \( \text{ref} \), \( \text{posh} \), and \( \text{negh} \) through the Bi-LSTM and BiC-LSTM networks, respectively. Then, the representations are fed into a feed-forward neural network as inputs to obtain neural network(NN)-based similarity scores, which are computed as below:

\[
\text{sim}_{Pn} = \sigma(W \cdot \sigma(W[\text{ref}_r, \text{posh}_r] + b)) \tag{2}
\]
\[
\text{sim}_{Nn} = \sigma(W \cdot \sigma(W[\text{ref}_r, \text{negh}_r] + b)) \tag{3}
\]

where \( \text{posh}_r \) denotes the representation of \( \text{posh} \), and \( \text{negh}_r \) of \( \text{negh} \). \( \text{sim}_{Pn} \) refers to the NN-based similarity score of \( \text{posh} \), while \( \text{sim}_{Nn} \) of \( \text{negh} \) given \( \text{ref}_r \). \( \text{sim}_{Pn} \) and \( \text{sim}_{Nn} \) share the same parameter weights \( W, V \) and the bias term \( b \).

Incorporating other metrics Next, we further optimize our model by incorporating lexical and syntactic metrics as features (in terms of metric scores), namely BLEU, NIST, METEOR, TERp and DPMF [13]. The concatenation of these 5 metric scores and the NN-based similarity scores are fed into a feed-forward layer, whose output shows the final similarity scores, \( \text{sim}_P \) and \( \text{sim}_N \) (mentioned in formula (1)).

\[
\text{sim}_P = \sigma(W_s[\text{sim}_{Pn}, s_{pr}] + b_s) \tag{4}
\]
\[
\text{sim}_N = \sigma(W_s[\text{sim}_{Nn}, s_{nr}] + b_s) \tag{5}
\]

where \( W_s \) is the parameter weight and \( b_s \) is a bias term. \( s_{nr} \) refers to the concatenated 5 metric scores of \( \text{neg} \), while \( s_{pr} \) that of \( \text{pos} \).
The testing phase During the testing phase, given a hypothesis hyp and a corresponding reference ref, the similarity score between them is computed as follows.

Firstly, the NN-based similarity score:
\[
\text{sim}(\text{ref}, \text{hyp}) = \sigma (W \cdot \sigma (V \cdot \text{ref}_{r} + \text{hyp}_{r}) + b)
\]
where ref, denotes the representation of ref, and hyp, of hyp. W and V are parameter weights, and b is the bias term. All W, V and b are the same with that in the training phase. Then, the final similarity score
\[
\text{sim} = \sigma (W_{s} \cdot \text{sim}(\text{ref}, \text{hyp}), s_{r} + b_{s})
\]
where W_s, b_s are the same with that in the training phase. s_r refers to the concatenated 5 metric scores of hyp given ref.

3.2 Bi-LSTM and BiC-LSTM Networks
We use Bi-LSTM and BiC-LSTM networks separately to produce the continuous space representations of ref, posh and negh, which are denoted as ref_r, posh_r and negh_r.

![Bi-LSTM Network](image)

**Bi-LSTM Network** Bi-LSTM networks have been employed to substantially improve performance in several NLP tasks. As illustrated in Figure 1, Bi-LSTM network consists of two parallel layers, a forward and a backward layer, propagating in two directions. These two layers enable the network to capture both past and future features for a given timestep. The two representation sequences produced by each layer are concatenated at each timestep, followed by mean pooling which outputs the representation of the sentence.
BiC-LSTM Network In order to capture inner connection between two sentences, we further propose an enhanced BiC-LSTM network (as illustrated in Figure 2), which takes the concatenation of the two sentences as input. The output is the representation of the second sentence. For instance, if the input of the forward layer is the concatenation of hyp and ref, denoted by \([hyp, ref]\), and that of the backward layer is the concatenation of reversals of both hyp and ref, then the network produces the representation of ref.

Fig. 3: The BiC-LSTM network. \(s_1\) denotes a sentence with length of \(m\), while \(s_2\) the other with that of \(n\). \(s_1\) and \(s_2\) are concatenated to go through the BiC-LSTM network, producing the representation of the second sentence \(s_2\), which contains the inner connection between \(s_1\) and \(s_2\).

4 Experiments and Results

4.1 Datasets

Experiments are conducted on the WMT metric shared task. Each training example is a tuple \((ref, posh, negh)\), extracted from the human judgement file of WMT-13, of which each line contains 5 human ranks of 5 randomly chosen hypotheses of a specific segment.

For duplicated tuples, we only retain one of them. There are also two tuples with opposite positions of posh and negh due to the inconsistent ranks between two annotators [2], in which case we remove the tuple appearing less often. Hence, we clean the training with respect to inconsistency and redundancy. In all, we obtain 285908 tuples for training. Evaluation is conducted on WMT-14 for other languages into English.
Table 1: Segment-Level Kendall’s tau correlations on WMT-14.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>cs-en</th>
<th>de-en</th>
<th>fr-en</th>
<th>ru-en</th>
<th>hi-en</th>
<th>PAvg</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>.218</td>
<td>.266</td>
<td>.376</td>
<td>.263</td>
<td>.299</td>
<td>.285</td>
</tr>
<tr>
<td>NIST</td>
<td>.231</td>
<td>.295</td>
<td>.382</td>
<td>.285</td>
<td>.312</td>
<td>.300</td>
</tr>
<tr>
<td>TERp-A</td>
<td>.345</td>
<td>.445</td>
<td>.589</td>
<td>.401</td>
<td>.407</td>
<td>.438</td>
</tr>
<tr>
<td>METEOR</td>
<td>.282</td>
<td>.344</td>
<td>.486</td>
<td>.333</td>
<td>.407</td>
<td>.395</td>
</tr>
<tr>
<td>DPMF</td>
<td>.283</td>
<td>.382</td>
<td>.494</td>
<td>.342</td>
<td>.426</td>
<td>.354</td>
</tr>
<tr>
<td>maxSD-1</td>
<td>.312</td>
<td>.353</td>
<td>.429</td>
<td>.342</td>
<td>.444</td>
<td>.376</td>
</tr>
<tr>
<td>maxSD-2</td>
<td>.310</td>
<td>.353</td>
<td>.431</td>
<td>.342</td>
<td>.440</td>
<td>.375</td>
</tr>
<tr>
<td>DISCOTK-PARTY-TUNED</td>
<td>.328</td>
<td>.380</td>
<td>.433</td>
<td>.355</td>
<td>.434</td>
<td>.386</td>
</tr>
<tr>
<td>BEER</td>
<td>.284</td>
<td>.347</td>
<td>.417</td>
<td>.331</td>
<td>.438</td>
<td>.362</td>
</tr>
</tbody>
</table>

Table 2: System-Level correlations on WMT-14.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>cs-en</th>
<th>de-en</th>
<th>fr-en</th>
<th>ru-en</th>
<th>hi-en</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>.963</td>
<td>.830</td>
<td>.961</td>
<td>.784</td>
<td>.928</td>
<td>.893</td>
</tr>
<tr>
<td>NIST</td>
<td>.949</td>
<td>.803</td>
<td>.964</td>
<td>.796</td>
<td>.667</td>
<td>.836</td>
</tr>
<tr>
<td>TERp-A</td>
<td>.863</td>
<td>.909</td>
<td>.976</td>
<td>.815</td>
<td>.438</td>
<td>.800</td>
</tr>
<tr>
<td>METEOR</td>
<td>.988</td>
<td>.925</td>
<td>.975</td>
<td>.807</td>
<td>.454</td>
<td>.829</td>
</tr>
<tr>
<td>DPMF</td>
<td>.999</td>
<td>.920</td>
<td>.967</td>
<td>.812</td>
<td>.882</td>
<td>.920</td>
</tr>
<tr>
<td>maxSD-1</td>
<td>.945</td>
<td>.922</td>
<td>.977</td>
<td>.827</td>
<td>.978</td>
<td>.930</td>
</tr>
<tr>
<td>maxSD-2</td>
<td>.948</td>
<td>.919</td>
<td>.977</td>
<td>.825</td>
<td>.979</td>
<td>.930</td>
</tr>
<tr>
<td>DISCOTK-PARTY-TUNED</td>
<td>.975</td>
<td>.942</td>
<td>.977</td>
<td>.870</td>
<td>.996</td>
<td>.944</td>
</tr>
<tr>
<td>LAYERED</td>
<td>.941</td>
<td>.883</td>
<td>.975</td>
<td>.864</td>
<td>.976</td>
<td>.927</td>
</tr>
</tbody>
</table>

4.2 Setups

Sentences with lengths exceeding 100 words are filtered out. The 300-dimensional glove word vectors [9] are used as the word embedding. The parameter weights are initialized by sampling from a normal distribution. We train for 10 epochs using adadelta. Our mini-batch size is 16, and dropout is used as suggested by [14]. The average of segment-level scores is the system-level score.

4.3 Results

We present two versions of our metric, namely maxSD-1 and maxSD-2 based on Bi-LSTM and Bi-LSTM networks respectively. We compare our metric with the best two in WMT-14, DISCOTK-PARTY-TUNED and BEER [12] on segment-level, and DISCOTK-PARTY-TUNED and LAYERED [4] on system-level respectively. Additionally, the other incorporated metrics are also listed in Table 1 and 2 for comparison. Scores in bold indicate best scores overall and those in bold italic show best scores achieved by our metric. Results in Tables 1 and 2 show that two versions of our metric outperform all other metrics, except DISCOTK-PARTY-TUNED, in all five directions both at the segment-
Fig. 4: Significance test results for differences in dependent correlation with human judgement (Williams test) for all competing pairs of metrics. A green cell denotes a significant win for the metric in a given row over the metric in a given column at $p < 0.05$. “PDF” in the figure corresponds to “DPMF” mentioned above.

and system-level. And our metrics are slightly behind the top-performing metric DISCOTK-PARTY-TUNED, which combines 17 different metrics requiring external resources and tuning efforts. However, for ‘hi-en’, we yield better results than DISCOTK-PARTY-TUNED, achieving the state-of-the-art results, with Kendall tau of 0.444 on the segment level and Pearson correlation of 0.979 on the system level. It is also worthy noting that maxSD-2 achieves the best performance in two (‘hi-en’ and ‘fr-en’) out of five directions at the system-level, and maxSD-1 the best in one direction at the segment-level. One interesting finding is that the enhanced maxSD-2 does not outperform maxSD-1. We suspect that the long length of the concatenated sentence affects the performance of BiC-LSTM network. As recommended by [5], significance tests for differences in dependent
correlation with human assessment were carried out for all competing metrics. Results of significance tests are shown in Figure 4.

5 Conclusion

Our proposed metric based on neural networks effectively achieves the state-of-the-art performance in two out of five language pairs on system-level and one on segment-level, and achieve comparative results for the remaining language pairs.

6 Acknowledgements

This work is supported by National Natural Science Foundation of P. R. China under Grant No. 61379086, European Union Horizon 2020 research and innovation programme under grant agreement 645452 (QT21), and the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie ) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

References

MaxSD: A Neural MTE Metric Optimized by Maximizing SD


I Blend: a Novel Combined MT Metric Based on Direct Assessment
— CASICT-DCU submission to WMT17 Metrics Task

Qingsong Ma1 Yvette Graham2 Shugen Wang1 Qun Liu2,1
1Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, University of Chinese Academy of Sciences
2 ADAPT Centre, School of Computing, Dublin City University
maqingsong@ict.ac.cn, graham.yvette@gmail.com
wangshugen@ict.ac.cn, qun.liu@dcu.ie

Abstract

Existing metrics to evaluate the quality of Machine Translation hypotheses take different perspectives into account. DPM-Fcomb, a metric combining the merits of a range of metrics, achieved the best performance for evaluation of to-English language pairs in the previous two years of WMT Metrics Shared Tasks. This year, we submit a novel combined metric, Blend, to WMT17 Metrics task. Compared to DPM-Fcomb, Blend includes the following adaptations: i) We use DA human evaluation to guide the training process with a vast reduction in required training data, while still achieving improved performance when evaluated on WMT16 to-English language pairs; ii) We carry out experiments to explore the contribution of metrics incorporated in Blend, in order to find a trade-off between performance and efficiency.

1 Introduction

Automatic machine translation evaluation (AMTE) has received much attention in recent years, with the aim of providing quick and stable measurements of the performance of machine translation (MT) systems. Various metrics for AMTE have been proposed and most operate via computation of the similarity between the MT hypothesis and the reference translation. However, different metrics focus on different perspectives in terms of measuring similarity. For lexical based metrics, BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) count n-gram co-occurrence, Meteor (Denkowski and Lavie, 2014) and GTM (Melamed et al., 2003) catch different kinds of matches, ROUGE (Lin and Och, 2004) captures common subsequences, WER (Nießen et al., 2000), PER (Tillmann et al., 1997) and TER (Snover et al., 2009) compute the post-editing distance between the hypothesis and the reference translation. Syntactic based metrics mainly use shallow syntactic structures (Chan and Ng, 2008; Zhu et al., 2010), dependency tree structures or constituent tree structures (Owczarzak et al., 2007; Liu and Gildea, 2005). Semantic measures (Lo et al., 2012) and discourse similarity based metrics (Guzmán et al., 2014) have also been proposed.

Different metrics evaluate similarity between hypotheses and reference translations from various perspectives, each of which has pros and cons. One straightforward and effective method to take advantage of the merits of existing metrics is to combine quality scores assigned by these metrics, like DPM-Fcomb (Yu et al., 2015a).

In WMT15 and WMT16 Metrics tasks, DPM-Fcomb was the best metric on average for to-English language pairs (Stanojević et al., 2015; Bojar et al., 2016). DPM-Fcomb incorporates lexical, syntactic and semantic based metrics, using ranking SVM1 to train parameters of each metric score and achieves a high correlation with human evaluation. Human evaluations in terms of relative ranking (RR) accumulated in WMT Metrics tasks are adopted to generate training data and to guide the training process. Human relative ranking is carried out by ranking the quality of 5 MT hypotheses of the same source segment from 1 to 5 via comparison with the reference translation.

1http://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html

Copenhagen, Denmark, September 7-11, 2017. ©2017 Association for Computational Linguistics

Page 90 of 200
Therefore, human RR only provides relative differences in quality of a given 5 hypotheses rather than the overall absolute quality of hypotheses. Besides, the low inter-annotator agreement level in RR (Callison-Burch et al., 2007) has been a long-lasting issue in MT human evaluation. The ability and the reliability of RR raise our concern whether the capability of the model trained with RR as the golden standard may be limited.

Fortunately, a new emerged evaluation approach, direct assessment (DA) (Graham et al., 2013), has been proven more reliable for evaluation of metrics and was recently adopted as the official human evaluation in WMT17. DA produces absolute quality scores of hypotheses, by measuring to what extend the hypothesis adequately expresses the meaning of the reference translation, through a 1-100 continuous rating scale that facilitates reliable quality control of crowd-sourcing. Large numbers of repeat human assessments per translation are standardized and then combined into a mean score as the final quality score of the MT hypothesis.

The recent development in human evaluation of MT motivates us to propose a new combined metric, named as Blend2, by adopting DA, as opposed to RR, to guide the training process indicating that a more reliable gold standard can lead to more reliable results even with less training data. Furthermore, we explore the contribution of metrics incorporated in Blend, aiming at finding a trade-off between performance and efficiency of Blend.

What follows is a brief review of DPMFcomb, before a description of Blend formulation is provided in Section 2, followed by experiments and results in Section 3, before the conclusions in section 4.

2 Metrics

2.1 Review of DPMFcomb

DPMFcomb utilizes human relative ranking data to train a combined metric that produces quality scores for MT hypotheses. In the training process, metrics are incorporated as features in the form of metric scores attributed to the same hypotheses, with relative ranks as the gold standard to guide SVM-rank to learn parameters for features. When testing, the predicted ranking scores produced by DPMFcomb reflect the quality of hypotheses. DPMFcomb allows the combination of the advantages of a set of arbitrary metrics resulting in a metric with a high correlation with human assessment. DPMFcomb includes default metrics provided by Asiya MT evaluation toolkit (Giménez and Márquez, 2010), as well as three other metrics, namely ENTF (Yu et al., 2015c), REDp (Yu et al., 2014) and DPMF (Yu et al., 2015b). Over the past two years of WMT metrics tasks, DPMFcomb has achieved the best performance for evaluation of MT of to-English language pairs.

### Table 1: The number of sampled DA data for each language pair in WMT15 and WMT16.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT15</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>−</td>
<td>500</td>
<td>−</td>
<td>500</td>
</tr>
<tr>
<td>WMT16</td>
<td>560</td>
<td>560</td>
<td>560</td>
<td>560</td>
<td>560</td>
<td>560</td>
<td>560</td>
</tr>
</tbody>
</table>

Although RR reflects the quality of hypotheses to some extent, it has two obvious defects. Firstly, RR provides relative ranks of the given competing MT hypotheses, which only reflects relative differences in quality rather than the absolute quality of hypotheses. On the other hand, RR suffers from low inter-annotator agreement levels. As a result, the capability of the model trained with RR as the golden standard could be limited. However, DA with carefully design of criteria (Graham et al., 2013) produces highly reliable overall quality scores for each hypothesis (Graham et al., 2015). In addition, since DA has replaced RR as the official human evaluation in the news domain in WMT17, more DA data would become available in the coming years. These motivate our new combined metric, specially designed based on DA, rather than RR, named as Blend, which means it is a metric that can blend advantages of arbitrary metrics in a combined metric that has a high correlation with human assessment.

Our metric follows the basic formulation of DPMFcomb. However, since DA is an absolute quality judgment, which is different from RR, the...
Table 2: System-level Pearson correlation of metric scores and DA human scores with 10K hybrid systems for to-English language pairs on WMT16, where “avg” denotes the average Pearson correlation of all language pairs.

<table>
<thead>
<tr>
<th></th>
<th>cs-en</th>
<th>de-en</th>
<th>fi-en</th>
<th>ro-en</th>
<th>ru-en</th>
<th>tr-en</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blend.all</td>
<td>.991</td>
<td>.954</td>
<td>.969</td>
<td>.879</td>
<td>.942</td>
<td>.972</td>
<td>.951</td>
</tr>
<tr>
<td>MPEDA</td>
<td>.988</td>
<td>.923</td>
<td>.971</td>
<td>.905</td>
<td>.923</td>
<td>.975</td>
<td>.948</td>
</tr>
<tr>
<td>BEER</td>
<td>.985</td>
<td>.871</td>
<td>.964</td>
<td>.828</td>
<td>.894</td>
<td>.975</td>
<td>.920</td>
</tr>
</tbody>
</table>

Table 3: Segment-level Pearson correlation of metric scores and DA human scores for to-English language pairs on WMT16, where “avg” denotes the average Pearson correlation of all language pairs.

<table>
<thead>
<tr>
<th></th>
<th>cs-en</th>
<th>de-en</th>
<th>fi-en</th>
<th>ro-en</th>
<th>ru-en</th>
<th>tr-en</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blend.all</td>
<td>.710</td>
<td>.615</td>
<td>.602</td>
<td>.636</td>
<td>.622</td>
<td>.658</td>
<td>.641</td>
</tr>
<tr>
<td>DPMFcomb</td>
<td>.713</td>
<td>.598</td>
<td>.584</td>
<td>.627</td>
<td>.615</td>
<td>.663</td>
<td>.633</td>
</tr>
<tr>
<td>METRICS-F</td>
<td>.696</td>
<td>.601</td>
<td>.557</td>
<td>.662</td>
<td>.618</td>
<td>.649</td>
<td>.631</td>
</tr>
</tbody>
</table>

3 Experiments

We carry out experiments to compare the performance of DPMFcomb and Blend. We also explore the contribution of incorporated metrics in Blend to find a trade-off between performance and efficiency.

3.1 Setups

Our experiments are tested on WMT16 to-English and English-Russian (en-ru) language pairs. We use DA data sampled from WMT15 and WMT16 (Table 1) for Blend. Since there is only a limited amount of DA data available at present, we employ all other to-English DA data as training data (4800 sentences) when testing on each to-English language pair (560 sentences) in WMT16. For en-ru, we use en-ru DA data in WMT15 (500 sentences) to train and test on en-ru DA data in WMT16 (560 sentences).

Features in both the training data and the test data are scaled to be in [-1,1]. We use epsilon-SVR with RBF kernel, and the epsilon is set to 0.1.

3.2 Blend vs DPMFcomb

In WMT16, DPMFcomb incorporates 57 metrics and was trained with SVM-rank on 445K training segments extracted from WMT12-WMT14 to-English language pairs according to human judgments in terms of RR. For comparison, Blend incorporates the same 57 metrics but is trained with SVM regression on only 4,800 training data extracted from sampled DA data in WMT15-WMT16 for each to-English language pair. We name it Blend.all.

We present the system and segment-level Pearson correlation results in Table 2 and Table 3, respectively. Table 2 shows Blend.all has higher average system-level Pearson correlation (.951) with DA human scores compared to the two high-performing metrics MPEDA (.948) and BEER (.920) on WMT16 for to-English language pairs.

Table 3 shows segment-level Pearson correlations of Blend.all and two other high-performing metrics DPMFcomb and EMTRICS-F on WMT16 for to-English language pairs. From Table 3 we can see Blend.all achieves the best performance in 3 out of 6 to-English languages pairs and state-of-the-art performance on average. It is worth noting that even though the training data of Blend.all is far less than that of DPMFcomb, Blend.all has higher average Pearson correlation (.641), trained on DA scores, than that of DPMFcomb (.633), trained on RR scores.

In all, the above results show Blend trained with DA data outperforms DPMFcomb trained with RR data on WMT16 for to-English language pairs.
Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

<table>
<thead>
<tr>
<th>Language Pairs</th>
<th>cs-en</th>
<th>de-en</th>
<th>fi-en</th>
<th>ro-en</th>
<th>ru-en</th>
<th>tr-en</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blend.all</td>
<td>.710</td>
<td>.615</td>
<td>.602</td>
<td>.636</td>
<td>.622</td>
<td>.658</td>
<td>.641</td>
</tr>
<tr>
<td>Blend.lex</td>
<td>.704</td>
<td>.589</td>
<td>.583</td>
<td>.625</td>
<td>.620</td>
<td>.674</td>
<td>.632</td>
</tr>
<tr>
<td>Blend.syn</td>
<td>.656</td>
<td>.528</td>
<td>.494</td>
<td>.560</td>
<td>.533</td>
<td>.610</td>
<td>.564</td>
</tr>
<tr>
<td>Blend.sem</td>
<td>.610</td>
<td>.533</td>
<td>.492</td>
<td>.507</td>
<td>.501</td>
<td>.554</td>
<td>.533</td>
</tr>
</tbody>
</table>

Table 4: Segment-level Pearson correlation of Blend incorporating different level of linguistic metrics for to-English language pairs on WMT16, where “avg” denotes the average Pearson correlation of all language pairs.

<table>
<thead>
<tr>
<th>Language Pairs</th>
<th>cs-en</th>
<th>de-en</th>
<th>fi-en</th>
<th>ro-en</th>
<th>ru-en</th>
<th>tr-en</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blend.lex</td>
<td>.704</td>
<td>.589</td>
<td>.583</td>
<td>.625</td>
<td>.620</td>
<td>.674</td>
<td>.632</td>
</tr>
<tr>
<td>Blend.lex+CharacTer</td>
<td>.707</td>
<td>.596</td>
<td>.575</td>
<td>.628</td>
<td>.620</td>
<td>.680</td>
<td>.634</td>
</tr>
<tr>
<td>Blend.lex+BEER</td>
<td>.709</td>
<td>.589</td>
<td>.580</td>
<td>.627</td>
<td>.622</td>
<td>.675</td>
<td>.634</td>
</tr>
<tr>
<td>Blend.lex+DPMF</td>
<td>.706</td>
<td>.592</td>
<td>.590</td>
<td>.632</td>
<td>.626</td>
<td>.670</td>
<td>.636</td>
</tr>
<tr>
<td>Blend.lex+ENTF</td>
<td>.703</td>
<td>.595</td>
<td>.588</td>
<td>.629</td>
<td>.629</td>
<td>.676</td>
<td>.637</td>
</tr>
<tr>
<td>Blend.lex+4</td>
<td>.709</td>
<td>.601</td>
<td>.584</td>
<td>.636</td>
<td>.633</td>
<td>.675</td>
<td>.640</td>
</tr>
</tbody>
</table>

Table 5: Segment-level Pearson correlation of Blend.lex incorporating 4 other metrics for to-English language pairs on WMT16, where “avg” denotes the average Pearson correlation of all language pairs.

3.3 Trade-off between Performance and Efficiency

It is convenient for Blend to combine arbitrary metrics in order to achieve a high correlation with human assessment. However, it would be useful to know if any metric does not contribute greatly to Blend in terms of performance, while at the same time leads to low efficiency. To explore this, we separate out the default metrics for to-English language pairs provided by Asiya toolkit into three categories, namely, lexical, syntactic, and semantic based metrics. Blend.lex is the variant that incorporates only default lexical based metrics in Asiya toolkit, while Blend.syn, and Blend.sem incorporate only syntactic and semantic metrics, respectively. Blend.lex includes 25 metrics, but with only 9 kinds of metrics, since some of them are simply different variants of the same metric. Blend.syn includes 17 metrics and Blend.sem 13 metrics but in reality each only corresponds to 3 distinct metrics, similar to Blend.lex.

The experimental results on WMT16 are shown in Table 4. It is not all that surprising that Blend.all incorporated with all default Asiya metrics achieves the best performance in 5 out of 6 language pairs and on average. However, it may be worth noting that the average Pearson correlation of Blend.lex is only 0.009 less than that of Blend.all, while the performance of Blend.syn and Blend.sem are quite far worse than that of Blend.all, and even that of Blend.lex. Since syntactic and semantic based metrics are usually complex, and the performance of Blend.lex is comparable with that of Blend.all, Blend can operate effectively with only incorporating the default lexical based metrics from Asiya toolkit.

We further add 4 other metrics to Blend.lex, CharacTer (Wang et al., 2016), a novel character-based metric; BEER (Stanojević and Sima’an, 2015), a metric combining different kinds of features; DPMF and ENTF, which proved to be effective. All of these 4 metrics are convenient to use. Table 5 shows Blend.lex+4 (.640) achieves better performance than that of Blend.lex (.632), and is very close to that of Blend.all (.641) as shown in Table 3.

Hence, we submit Blend.lex+4 to WMT17 Metrics task for to-English language pairs, since it provides a good trade-off between performance and efficiency for Blend.

3.4 Experiments on from-English language pairs

Blend can be effective to evaluate the quality of from-English MT hypotheses if incorporated metrics support from-English language pairs. We carry out experiments on WMT16 for en-ru language pair as shown in Table 6.

For from-English language pairs, there is only en-ru DA data available at present.
Table 6: Segment-level Pearson correlation for en-ru in WMT16.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blend.default</td>
<td>.613</td>
</tr>
<tr>
<td>Blend.default+2</td>
<td>.675</td>
</tr>
<tr>
<td>BEER</td>
<td>.666</td>
</tr>
</tbody>
</table>

is trained on only 500 sentences and incorporates default lexical based metrics from Asiya toolkit for en-ru, including 20 metrics, but with 9 kinds of metrics only. Compared with Blend.default, Blend.default+2 incorporates two more metrics, CharacTer and BEER, but achieves great improvement with segment-level Pearson correlation from .613 to .675. The incorporated metric BEER is the best performing metric (.666) on WMT16 for en-ru, which is trained with large amounts of data. Beer contributes to Blend apparently, meanwhile Blend can further improve the performance of BEER, indicating the effectiveness of the combined metric Blend. We submit Blend.default+2 to WMT17 Metrics task for en-ru.

4 Conclusions

The performance of DPMFcomb proves the effectiveness of the idea of combining metrics. However, DPMFcomb cannot extend itself to the new development of human evaluation. Therefore, we propose a novel metric Blend to employ DA data. Blend is also a combined metric that can take good advantage of the merits of existing metrics, and performs better than DPMFcomb, even with far less training data. Blend is easy to be trained and flexible to be applied to any language pairs. In this paper we present experiments on WMT16 Metrics task, which shows Blend achieves state-of-the-art performance on average for to-English language pairs and for en-ru. Furthermore, we carry out experiments with different settings and find a good trade-off for Blend in terms of performance and efficiency.

Acknowledgments

This research is supported by Chinas NSFC grant 61379086 and the European Union Horizon 2020 Programme (H2020) under grant agreement no. 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.

References


Can machine translation systems be evaluated by the crowd alone

YVETTE GRAHAM1,2, TIMOTHY BALDWIN1, ALISTAIR MOFFAT1 and JUSTIN ZOBEL1
1Department of Computing and Information Systems, The University of Melbourne, Parkville, 3010 VIC, Australia
c-mail: tbaldwin@unimelb.edu.au, amoffat@unimelb.edu.au, jzobel@unimelb.edu.au
2ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin, College Green, Dublin 2, Ireland
c-mail: graham.yvette@gmail.com

(Received 13 December 2014; revised 7 August 2015; accepted 10 August 2015; first published online 16 September 2015)

Abstract
Crowd-sourced assessments of machine translation quality allow evaluations to be carried out cheaply and on a large scale. It is essential, however, that the crowd’s work be filtered to avoid contamination of results through the inclusion of false assessments. One method is to filter via agreement with experts, but even amongst experts agreement levels may not be high. In this paper, we present a new methodology for crowd-sourcing human assessments of translation quality, which allows individual workers to develop their own individual assessment strategy. Agreement with experts is no longer required, and a worker is deemed reliable if they are consistent relative to their own previous work. Individual translations are assessed in isolation from all others in the form of direct estimates of translation quality. This allows more meaningful statistics to be computed for systems and enables significance to be determined on smaller sets of assessments. We demonstrate the methodology’s feasibility in large-scale human evaluation through replication of the human evaluation component of Workshop on Statistical Machine Translation shared translation task for two language pairs, Spanish-to-English and English-to-Spanish. Results for measurement based solely on crowd-sourced assessments show system rankings in line with those of the original evaluation. Comparison of results produced by the relative preference approach and the direct estimate method described here demonstrate that the direct estimate method has a substantially increased ability to identify significant differences between translation systems.

1 Introduction
The ability to develop and refine machine translation (MT) systems is critically reliant on the availability of reliable methods of assessing the quality of translations. Expert assessment of translation quality is widely held as a ‘gold standard’ yardstick, but is costly. Automatic evaluation is often used as a substitute, as a means of, for example, supporting automatic MT system comparison, rapid MT deployment, and automatic tuning of system parameters (Och 2003; Kumar and Byrne 2004). However, it is well documented that the automatic MT evaluation metrics in current use fall short of human assessment (Callison-Burch 2009; Graham, Mathur and
Baldwin 2014). As a result, manual evaluation is still widely used as the primary means of evaluation in large-scale shared tasks, such as the annual Workshop on Statistical Machine Translation (WMT) (Bojar et al. 2014). Additionally, for the purpose of benchmarking and development of automatic MT evaluation metrics, it is vital that data sets are developed with high-quality human assessments and over a variety of language pairs.

Human assessment can be expert or crowd-sourced, or a mixture of these. To date, WMT shared translation tasks have mainly used expert-only human evaluations as the basis of official evaluation (Callison-Burch et al. 2007, 2008, 2009, 2010; Bojar et al. 2014). There was experimentation with a mix of expert and crowd-sourced judgments to produce official results in 2012–2013 (Callison-Burch et al. 2012; Bojar et al. 2013), but in 2014, the organizers reverted to expert-only assessments due to worryingly low inter-annotator agreement rates for some language pairs.

We argue that it is possible to measure MT systems reliably based on crowd-sourced judgments alone. To demonstrate that claim, we have explored and quantified an approach to crowd-sourcing of human assessments of translation quality using the Amazon Mechanical Turk (AMT) platform. Key features of our proposed methodology are:

- It can be used to assess both adequacy and fluency;
- It requires only monolingual annotators conversant in the target language, thus allowing use of a larger pool of lower-skilled annotators than is possible with standard manual evaluation approaches, which require bilingual annotators with high-level competency in both the source and target languages;
- The ratings are captured via direct estimates on a 100-point Likert scale, enabling fine-grained statistical analysis (Graham et al. 2013);
- It incorporates mechanisms for quality control, based on internal consistency over pairings of original and ‘degraded’ translations (Graham et al. 2014);
- It is backwards-compatible with the style of system preference judgment used for WMT evaluations, and provides a mechanism for enabling longitudinal evaluation of MT systems;
- It is cheap: we obtained high-quality, statistically significant assessments at a cost of around US$40 per system for a given language direction and evaluation modality (that is, adequacy or fluency).

To investigate the feasibility of the approach for large-scale evaluations, we replicated the original WMT-12 human evaluation of all participating systems for two language pairs: Spanish-to-English (ES-EN) and English-to-Spanish (EN-ES). Results show that our method results in high-quality data at low cost, without the use of expert assessments. Indeed, we demonstrate that it is possible to confirm the original system rankings; and also, using comparable numbers of judgments, identify a larger number of significant differences between systems.

The low cost of the method suggests that, beyond applications for cross-system evaluation in shared tasks, it may also provide a viable means for individual MT researchers to assess system improvements over a baseline. While development of the tools used by the workers involved significant effort and refinement, this effort
need not be repeated, as we make these tools available. Overall, we show that carefully-gathered crowd-sourced assessments lead to more sensitive measurements than do assessments from other sources, suggesting that our methods should be used widely for the measurement of MT systems.

2 Background

2.1 Automatic measurement of MT systems

The development of effective mechanisms for evaluation of MT system output has long been a research objective within MT, with several of the recommendations of the early ALPAC Report (Pierce et al. 1966), for example, relating to evaluation:

1. Practical methods for evaluation of translations; . . .
3. Evaluation of quality and cost of various sources of translations;

In practical terms, improvements are often established through the use of an automatic metric that computes a similarity score between the candidate translation and one or more human-generated reference translations. However, it is well known that automatic metrics are not necessarily a good substitute for human assessments of translation quality, and must be used with caution (Turian, Shen and Melamed 2003; Callison-Burch, Osborne and Koehn 2006; Koehn and Monz 2006; Lopez 2008). Particular issues include:

- there are generally many different ways of translating the same source input, and therefore comparison with a reference translation risks artificially upscoring translations that happen to be more reference-like compared to equally-valid translations that make different lexical or structural choices; and
- automatic metrics are generally based on lexical similarity and fail to capture meaning, either in terms of fundamental semantic infelicity (for example, the lack of a negation marker or core argument) or underlying semantic similarity but lexical divergence with a reference translation (Koehn and Monz 2006; Lo et al. 2013).

One way of reducing bias in evaluation towards the particular decisions made in a reference translation is to source multiple reference translations and calculate an aggregated score across them (Culy and Riehemann 2003; Madnani et al. 2008). Even here, however, automatic evaluation metrics tend to focus too much on local similarity with translations, and ignore the global fidelity and coherence of the translation (Lo et al. 2013); this introduces a bias when comparing systems that are based on differing principles. Other approaches, such as HyTER (Dreyer and Marcu 2012), aim to encode all possible correct translations in a compact reference translation network, and match the output of an MT system against this using string edit distance. Such approaches are hampered by a lack of automation in crafting the reference translations, and currently require up to two hours per sentence.

1 See https://github.com/ygraham/crowd-alone
To alleviate these concerns, direct human assessments of translation quality are also collected when possible. During the evaluation of MT shared tasks, for example, human assessments of MT outputs have been used to determine the ranking of participating systems. The same human assessments can also be used in the evaluation of automatic metrics, by comparing the degree to which automatic scores (or ranks) of translations correlate with them. This aspect of MT measurement is discussed in the following section.

2.2 Validation of automatic metrics

In order to validate the effectiveness of an automatic MT evaluation metric, a common approach is to measure correlation with human assessments of MT quality. A corpus of test sentences is selected, and a set of MT systems is used to translate each sentence (often, the pool of systems participating in a shared task). Human assessors are then asked to assess the quality of the translations, or a randomly selected subset of them if experimental cost is a limiting constraint. In addition, some translation assessments are repeated, to facilitate later measurement of assessment consistency levels.

Once sufficiently many sentence-level human assessments have been collected, they are used to decide a best-to-worst ranking of the participating MT systems. A range of methods for going from sentence-level human assessments to a totally-ordered system ranking have been proposed (Bojar, Ercegovcic and Popel 2011; Callison-Burch et al. 2012; Lopez 2012; Hopkins and May 2013), with no consensus on the best aggregation method. An automatic MT evaluation metric can be used to mechanically rank the same set of systems, perhaps based on document-level scoring. The system ranking generated based on the human assessments and the automatic metric can then be compared based on correlation, using, for example, Spearman’s ρ or Pearson’s r, with a high correlation interpreted as evidence that the metric is sound. The robustness of this use of Spearman correlation has been questioned, as it cannot discriminate between errors with respect to varying magnitude in system scores (Graham and Baldwin 2014; Machacek and Bojar 2014).

Since the validity of an automatic MT evaluation measure is assessed relative to human assessments, it is vital that the human assessments are reliable. In practice, measurement of evaluation reliability is based on evaluation of intra- and inter-annotator agreement. There is a worrying trend of low agreement levels for MT shared tasks, however. For example, Cohen’s κ coefficient (Cohen 1960), is commonly used to quantify inter-annotator agreement levels and incorporates the likelihood of agreement occurring by chance, with a coefficient ranging between 0, signifying agreement at chance levels, to 1 for complete agreement. In recent WMT shared tasks, κ coefficients as low as κ = 0.40 (2011), κ = 0.33 (2012), κ = 0.26 (2013), and κ = 0.37 (2014) have been reported. Intra-annotator agreement levels have not been much better: κ = 0.58 (2011), κ = 0.41 (2012), κ = 0.48 (2013), and κ = 0.52 (2014) (Callison-Burch et al. 2011, 2012; Bojar et al. 2013, 2014). It is possible that alternative measures that capture different types of disagreement, such as Krippendorff’s α, are more appropriate for the task, given that it involves multiple
Natural language engineering

coders (Poesio and Artstein 2005; Mathet et al. 2012). However, such low levels of agreement are unlikely to be wholly attributable to a deficiency in the $\kappa$ coefficient as a measure of agreement, as individual annotators only weakly agree with themselves, let alone other annotators.

The lack of coherence amongst human assessments raises a critical question: are assessments of MT evaluation metrics robust, if they are validated via low-quality human assessments of translation quality? One possible reaction to this question is that the automatic evaluation measures are no worse than human assessment. A more robust response is to find ways of increasing the reliability of the human assessments used as the yardstick for automatic metrics, by identifying better ways of collecting and assessing translation quality. It may be, for example, that we are not asking assessors ‘is this a good translation’ in a form that leads to a consistent interpretation.

There has been significant effort invested in developing metrics that correlate with human assessments of translation quality. However, given the extent to which accurate human assessment of translation quality is fundamental to empirical MT, the underlying topic of finding ways of increasing the reliability of those assessments to date has received surprisingly little attention (Callison-Burch et al. 2007, 2008, 2009; Przybocki et al. 2009; Callison-Burch et al. 2010; Denkowski and Lavie 2010).

2.3 Past and current methodologies

The ALPAC Report (Pierce et al. 1966) was one of the earliest published attempts to perform cross-system MT evaluation, in determining whether progress had been made over the preceding decade. The (somewhat anecdotal) conclusion was that:

The reader will find it instructive to compare the samples above with the results obtained on simple, selected, text 10 years earlier . . . in that the earlier samples are more readable than the later ones.

The DARPA MT initiative of the 1990s incorporated MT evaluation as a central tenet, and periodically evaluated the three MT systems funded by the program (CANDIDE (Berger et al. 1994), PANGLOSS (Foderking et al. 1994), and LINGSTAT (Yamron et al. 1994)). As part of this, it examined whether post-editing of MT system output was faster than simply translating the original from scratch (White, O’Connell and O’Mara 1994). One of the major outcomes of the project was the proposal that adequacy and fluency be used as the primary means of human MT evaluation, supplemented by other human-assisted measurements. Adequacy is the degree to which the information in the source language string is preserved in the translation,2 while fluency is the determination of whether the translation is a well-formed natural utterance in the target language.

2 Or, in the case of White, O’Connell and O’Mara (1994), the degree to which the information in a professional translation can be found in the machine translation, as judged by monolingual speakers of the target language.
Many of the large corporate MT systems use regression testing to establish whether new methods have a positive impact on MT quality. Annotators are asked to select which of two randomly-ordered translations they prefer, one from each system (Bond, Ogura and Ikehara 1995; Schwartz, Aikawa and Quirk 2003), often over a reference set of translation pairs (Ikehara, Shirai and Ogura 1994).

Approaches trialed for human judgments of translation quality include all of:

- ordinal level scales (ranking a number of translations from best-to-worst) or direct estimates (interval-level scales) of fluency or adequacy judgments;
- different lexical units (for example, sub-sentential constituents rather than whole sentences);
- differing numbers of points on an interval-level scale (for example, having 4, 5, or 7 points);
- displaying or not displaying interval-level scale numbering to annotators;
- simultaneously assessing fluency and adequacy items, or separating the assessment of fluency and adequacy;
- changing the wording of the question used to elicit judgments (for example, asking which translation is better, or asking which is more adequate);
- including or not including the reference translation among the set being judged;
- displaying the translation of the sentences either side of the target sentence, or not displaying any surrounding context;
- displaying or not displaying session or overall participation meta-information to the assessor (for example, the number of translations assessed so far, the time taken so far, or the number of translations left to be assessed);
- employing or not employing crowd-sourced judgments.

In recent years, the annual WMT event has become the main forum for collection of human assessments of translation quality, despite the primary focus of the workshop being to provide a regular cross-system comparison over standardized data sets for a variety of language pairs by means of a shared translation task (Koehn and Monz 2006; Callison-Burch et al. 2007, 2008, 2009, 2010, 2011, 2012; Bojar et al. 2013, 2014). When WMT began in 2006, fluency and adequacy were adopted, and were also used in the LDC report (LDC 2005) to assess participating systems in the form of direct estimates on a five-point interval-level scale. Too few human assessments were recorded in the first year to be able to estimate the reliability of human judgments (Koehn and Monz 2006). In 2007, the workshop measured the consistency of the human judgments, to boost the robustness of evaluation; to achieve this goal, human assessments of translations were provided by participants in the shared task.

The reliability of human judgments was then estimated by measuring levels of agreement. In addition, two new further methods of human assessment were added: (1) generate a partial ranking of translations of full sentences from best to worst (or relative preference judgments); and (2) generate a partial ranking of translations of sub-sentential source syntactic constituents from best to worst. The first of these methods has continued in recent WMT evaluations; in both cases, ties are
allowed. The highest levels of agreement were reported for the sub-sentential source syntactic constituent ranking method ($\kappa_{\text{inter}} = 0.54$, $\kappa_{\text{intra}} = 0.74$), followed by the full sentence ranking method ($\kappa_{\text{inter}} = 0.37$, $\kappa_{\text{intra}} = 0.62$). The lowest agreement levels occurred for adequacy ($\kappa_{\text{inter}} = 0.31$, $\kappa_{\text{intra}} = 0.54$). Additional methods of human assessment have also been trialed, but the only method still currently used is the best-to-worst partial ranking of translations, known as ‘relative preference judgments’.

Another finding from WMT 2007 with widespread consequences was high correlation between fluency and adequacy, which was taken as evidence for redundancy in separately assessing the two, and led to the conflation of translation quality into a single scale (Przybocki et al. 2009; Denkowski and Lavie 2010). For example, Przybocki et al. (2009) use, as part of their larger human evaluation, a single (seven-point) scale (labeled ‘adequacy’) to assess the quality of translations. Inter-annotator agreement for this method was $\kappa = 0.25$, even lower than the results for adequacy and fluency reported in WMT 2007 (noting that caution is required when directly comparing agreement measurements, especially over scales of varying granularity, such as five- versus seven-point assessments).

Three recent WMT shared tasks (2011, 2012, and 2013) have limited their human assessment to partially ranking translations from best-to-worst. Even with this simpler conceptualization of the annotation task, consistency levels are still low. Agreement levels reported in, for example, WMT-12 using translation ranking were lower than those reported in 2007 with fluency and adequacy assessments.

As an additional confound, in the absence of a better alternative, WMT human evaluations are typically carried out by MT researchers participating in the shared task, rather than by independent volunteers, or by expert translators. Although the assessments of translations are blind, and the system that produced any particular translation is hidden from the judges, researchers have been shown to slightly favor translations produced by their own system (Bojar et al. 2011), highlighting the need for other methods of incorporating human evaluation into MT evaluation. Crowd-sourced human judgments eliminate that particular confound, but bring a different risk – vastly different skill and/or levels of care. Hence, until now, participating researchers have been preferred as judges to crowd-sourced assessors.

Turning to the question of what number of judgments is needed to produce statistically significant rankings, simulation experiments suggest that when assessment is in the form of relative preference judgments, this number is likely to grow quadratically in the number of participating systems (Koehn 2012; Bojar et al. 2014). For example, up to triple, the number of judgments collected in WMT-12 may have been required to identify sufficient proportions of significant differences between systems (Koehn 2012). Although a range of models have been proposed that aim to leverage the shortcomings of relative preference judgments to identify significant differences between systems, such as Hopkins and May (2013) and Sakaguchi, Post and Durme (2014), our work instead focuses on attending to the root cause of what we believe to be the problem: the information-loss that takes place when there is no attempt to capture the degree to which one translation is better than another.
3 Simplified monolingual assessment

Direct (or absolute) estimates of translation quality facilitate more powerful statistical analyses of systems than do partial ordinal rankings via relative preference judgments, which, for example, cannot be readily combined into mean and median scores. Being able to combine individual direct estimate scores into a mean or median score simplifies the decision as to how an overall ranking should be concluded; moreover, the law of large numbers suggests that as increasing numbers of individual scores are collected for each system, the mean score will increasingly approach the true score. By simply increasing the number of assessments (given access to suitably skilled assessors), accuracy is improved.

3.1 Capturing direct estimate assessments

To address those concerns, we propose that assessments be based on direct estimates of quality using a continuous rating scale. We further suggest that it is undesirable to assess several translations at a time, as even when performing direct estimation, there is the potential for ratings of different translations to influence each other. For example, if a translation happens to appear in the same displayed set as a high quality translation, it is likely that its score will be pushed down (Bojar et al. 2011). To minimize this kind of bias, we present a single translation per screen to human assessors, and require that each translation be considered in isolation. Assessors can only proceed to the next translation once they have committed their score for the current one, with no facility to revisit or revise earlier decisions. A third component of our proposal is to revert to the use of two-dimensional fluency and adequacy assessments, in part because doing so allows the assessments to be presented in the form of focused questions that are more likely to elicit consistent responses. In addition, by making the fluency component independent, part of the overall evaluation becomes completely reference-free, and hence not subject to possible comparison bias. We use the same continuous scale for both fluency and adequacy.

For technical reasons, the continuous rating scale is presented as a hundred-point slider, with the assessor selecting a rating by moving the slider with their mouse. Strictly speaking, it is thus not continuous; but the actual rating obtained is interpreted as a real-valued number, and the proposed methodology could trivially be applied to a finer-grained scale (for example, when higher-resolution screens become commonplace). Numbering is intentionally avoided, but the slider is marked at the mid and quarterly points for rough calibration purposes. We make every effort to make the assessment task compatible with crowd-sourcing, including couching both the fluency and adequacy questions as monolingual (target language) tasks, to maximize the pool of potential assessors. We intentionally obscure the fact that the text is the product of MT, to eliminate that knowledge as a potential source of bias.

The sections that follow describe the assessment procedure for fluency and adequacy in detail below, and the processes employed for score standardization and filtering out poor-quality assessors. The platform we use to collect the assessments is AMT (see https://www.mturk.com), in which jobs are presented to ‘workers’ in the form of human intelligence tasks (abbreviated as ‘HITs’), each corresponding...
to a single unit that a worker accepts through the AMT interface. Each HIT is made up of either all adequacy or all fluency assessment tasks, and each task is presented on a separate screen, with no facility for workers to return to revise earlier assessments. In all cases, the translated text is presented to the worker as a bit-mapped image, in order to deter robotic workers who could, for example, screen-scrape the translated text and use some automatic method to calculate their assessment. Although the assessments in this work were collected specifically with AMT, other crowd-sourcing services could also be used or set-up for this purpose. Therefore, although precise replication of the experimental results presented in this work may rely on the existence and continuity of the AMT service, more importantly the approach itself does not.

### 3.2 Adequacy: measuring equivalence of meaning

The first dimension of the evaluation is assessment of adequacy. Adequacy is assessed as a monolingual task, with a single translation presented per screen. The reference translation is provided in gray font at the top of the screen, with the system translation displayed below it in black. Assessors are asked to state the degree to which they agree that: The black text adequately expresses the meaning of the gray text.

The task is thus restructured into a less cognitively-taxing monolingual similarity-of-meaning task, under the fundamental assumption that the reference translation accurately captures the meaning of the source sentence. Once that presumption is made, it is clear that the source is not actually required for evaluation. An obvious benefit of this change is that the task now requires only monolingual speakers, without any knowledge of translation or MT. Figure 1 is a screenshot of a single adequacy assessment as posted on AMT.

Reference translations used for the purpose of adequacy assessments are those included as standard in MT test sets, and this avoids the need to, for example, attempt to generate reference translations through expert translators or crowd-sourcing. If reference translations were to be crowd-sourced, care would of course need to be taken to ensure effective quality control of this process to avoid introduction of noise into adequacy evaluations.

### 3.3 Reference-free evaluation of fluency

The second dimension of the evaluation is assessment of fluency. Fluency assessments are presented using the translated text only, with neither the source language input text shown, nor any reference translation(s). This removes any bias towards systems that happen to produce reference-like translations, a common criticism of automatic metrics such as BLEU. It also forces our assessors to make an independent fluency judgment on the translation, with the intention of minimizing biasing from other assessments.

Figure 2 shows a screenshot of a fluency assessment, as one task within an AMT HIT of similar assessments. As with the adequacy assessments, a single translation
Read the text below and rate it by how much you agree that:

The black text adequately expresses the meaning of the gray text.

On Facebook, it’s impossible to know how much of a user’s profile is true.

With Facebook, it’s difficult to know how many of a user profile information is true.

Fig. 1. (Colour online) Screenshot of the adequacy assessment interface, as presented to an AMT worker. All of the text is presented as an image. The slider is initially centered; workers move it to the left or right in reaction to the question. No scores or numeric information are available to the assessor.

Read the text below and rate it by how much you agree that:

The text is fluent English.

With Facebook, it’s difficult to know how many of a user profile information is true.

Fig. 2. (Colour online) Screenshot of the fluency assessment interface, as presented to an AMT worker. Many of the details are the same as for the adequacy assessment shown in Figure 1.

is displayed to the human assessor at a time, which they are then asked to rate for fluency. The assessment is carried out based on the strength of agreement with the Likert-type statement: *The text is fluent English.*

With source and reference sentences not shown, fluency assessments alone cannot be used to determine if one system is better than another. This is because it is possible for a system to produce highly-fluent text with complete disregard for the actual content of the source input. In our evaluation, fluency assessments are used as a means of breaking ties between systems that are measured to have equal adequacy, as well as a diagnostic tool that is unbiased in favor of systems that produce reference-like translations.

### 3.4 HIT structure and score standardization

Since fluency and adequacy assessments are carried out separately, we set these assessments up as separate HITs. Both kinds of HIT contain hundred translation assessments each, one per screen. The worker is thus required to iterate through translations, rating them one at a time. Within each HIT genuine system output translations may be paired with a repeat item, a corresponding reference translation, or a ‘bad reference’ translation (a degraded version of the same translation,
explained in more detail below). The aim of this selective pairing is that post-HIT completion, regardless of the overall scoring strategy of individual workers, their internal consistency at scoring better or worse translations can be examined without comparison to someone else’s scores. Structuring the HIT as hundred translations allows us to manipulate the task in such a way that we have a high level of control of same-judge repeat and quality-control items, as follows.

Within a hundred-translation HIT, we include: (a) ten reference translations and ten MT system outputs for the same source language strings as the reference translations (making a total of twenty translations); (b) ten MT system outputs, along with a mechanically-degraded ‘bad reference’ version of each (making a total of another twenty translations; see below for details of the degradation process); and (c) another fifty MT system outputs, out of which we select ten and repeat them verbatim (making a total of sixty translations). That is, each hundred-translation HIT consists of:

- seventy MT system outputs,
- ten reference translations, corresponding to ten of the seventy system outputs,
- ten bad reference translations, corresponding to a different ten of the seventy system outputs,
- ten repeat MT system outputs, drawn from the remaining fifty of the original seventy system outputs.

The role of the reference, bad reference, and repeat translations is explained in the next section.

The order in which translations are assessed by a worker is controlled so that a minimum of forty assessments intervene between each member of a pair of quality-control items (bad reference versus MT system, or MT system versus MT system repeat, or reference translation versus MT system). The selection is further controlled so that each hundred-translation HIT contains approximately equal numbers of randomly-selected translations from each contributing MT system, so as to provide overall balance in the number of translations that are judged for each system. That is, no matter how many HITs each worker completes, they will return roughly the same number of assessments for each of the contributing systems. This helps avoid any potential skewing of results arising from particularly harsh or lenient assessors.

To further homogenize the outputs of the different workers, the set of scores generated by each person is standardized. This is done in the usual way, by computing the mean and standard deviation of the scores returned by that particular assessor, and then translating each of their raw scores into a z score. The result is a set of scores that, for each assessor, has a mean of 0.0 and a standard deviation of 1.0; it is those values that are then averaged across systems to get system scores. Standardizing the scores removes any individual biases for particular sub-regions of the scale (for example, workers who tend to rate everything low or high), and also for the relative spread of scores used by a given worker (for example, workers who use the full scale versus those who use only the central region). In the results tables below, we report system averages of both the raw worker responses and of the standardized values.
4 Quality control and assessor consistency

The quality of a human evaluation regime can be estimated from its consistency: whether the same outcome is achieved if the same question is asked a second time. Two different measurements can be made: whether a judge is consistent with other assessments performed by themselves (intra-annotator agreement), and whether a judge is consistent with other judges (inter-annotator agreement).

In MT, annotator consistency is commonly measured using Cohen’s $\kappa$ coefficient, or some variant thereof (Artstein and Poesio 2008). Cohen’s $\kappa$ is intended for use with categorical assessments, but is also commonly used with five-point adjectival-scale assessments, where the set of categories has an explicit ordering. A particular issue with five-point assessments is that there is no notion of ‘nearness’ – a judge who assigns two neighboring intervals is awarded the same ‘penalty’ for being different as a judge who selects two extreme values.

The $\kappa$ coefficient cannot be applied to continuous data; and nor can any judge, when given the same translation to evaluate twice on a continuous rating scale, be expected to give the same score for each assessment. A more flexible tool is thus required. Our approach to measuring assessor consistency is based on two core assumptions:

(A) When a consistent judge is presented with a set of assessments for translations from two systems, one of which is known to produce better translations than the other, the score sample of the better system will be significantly greater than that of the inferior system.

(B) When a consistent judge is presented with a set of repeat assessments, the score sample across the initial presentations will not be significantly different from the score sample across the second presentations.

We evaluate Assumption A based on the set of bad reference translations and the corresponding set of MT system translations, and evaluate Assumption B based on the pairs of repeat judgments in each HIT. The idea behind bad reference translations is that if words are removed from a translation to shorten it, worse adequacy judgments can be expected; and if words in a translation are duplicated, fluency ratings should suffer. The null hypothesis to be tested for each AMT worker is that the score difference for repeat judgment pairs is not less than the score difference for bad reference pairs. To test statistical significance, we use the Wilcoxon rank sum test, with lower $p$ values indicating more reliable workers (that is, greater differentiation between repeat judgments and bad reference pairs). We use $p < 0.05$ as a threshold of reliability, and discard all of the HITs contributed by workers who do not meet this threshold. Note that if HITs are rejected in this manner, the workers may still receive payment.

The bad reference translations that are inserted into each HIT are deliberately degraded relative to their matching system translations, on the assumption that...
a measurable drop in the assessor’s rating should be observed. For the adequacy HITs, we degrade the translation by randomly deleting a short sub-string, emulating omission of a phrase. Initial experimentation showed that for this approach to be effective, we needed to delete words in rough proportion to the length of the original translation, as follows:

- for 2–3 word translations, remove one word;
- for 4–5 word translations, remove two words;
- for 6–8 word translations, remove three words;
- for 9–15 word translations, remove four words;
- for 16–20 word translations, remove five words;
- for translations of word length \( n > 20 \), remove \( \lfloor n/5 \rfloor \) words.

The bad reference translations for fluency HITs are created by randomly selecting two words in the translation and duplicating them elsewhere in the string, excepting adjacent to the original, string-initial, and string-final positions. Preliminary experiments indicated that it was not necessary to modify the length of the randomly-selected phrase according to the length of the overall translation. Duplicating a pair of words was always sufficient to obtain the desired effect.

Automatic introduction of errors to corpora have been used for other purposes besides quality control of crowd-sourced data. Pevzner and Hearst (2002) simulate errors for the purpose of analyzing evaluation metrics of thematic segmentation, while Mathet et al. (2012) use reference annotation shuffles according to different error paradigms to model the behavior of a range of agreement measures. Additionally, the automatic introduction of errors into corpora have been used to create training data for error detection applications (Izumi et al. 2003; Bigert 2004; Bigert et al. 2005; Sjoberg and Knutsen 2005; Smith and Eisner 2005a,b; Brockett, Dolan and Gamon 2006; Foster 2007; Okanohara and Tsujii 2007; Wagner, Foster and van Genabith 2007; Lee and Senell 2008; Foster and Andersen 2009; Wagner, Foster and van Genabith 2009; Dickinson 2010; Rozovskaya and Roth 2010; Yuan and Felice 2013).

### 4.1 Crowd-sourcing specifics

Mechanical Turk was used directly to post HITs to crowd-sourced workers, as opposed to any intermediary service, and fluency and adequacy HITs are collected in entirely separate sessions. Since our HIT structure is somewhat unconventional, with hundred translations per HIT, it was important that information about the quantity of work involved per HIT was communicated to workers prior to their acceptance of an HIT. An additional specification was that only native speakers of the target language complete HITs, and although we do not have any way of verifying that workers adhered to this request, it is important to communicate expected language fluency levels to workers. Payment was at the rate of US$0.50 per hundred-translation fluency HIT, and increased to US$0.90 per adequacy HIT – the difference because, in the latter case, HITs involved reading both a reference translation and the assessed translation. Workers were not required to complete a qualification test prior to...
carrying out HITs, as such qualification restrictions unfortunately do not ensure high quality work. Due to the anonymous nature of crowd-sourcing, it is of course entirely possible for workers lacking the necessary skills to employ someone else to complete his/her test. In addition, qualification tests do not provide any assurance that skilled workers do not carefully completing tests, and then aggressively optimize earnings at a later stage. In contrast, the quality control mechanism we employ does not rely on one-off tests but applies quality checks across all of the HITs provided by each worker.

Since the quality control mechanism we apply could in fact be too high a bar for some genuine workers to meet, we do not use or recommend its use as the sole basis for accepting or rejecting HITs. Indeed, some workers may simply lack the necessary literacy skills to accurately complete evaluations effectively. During collection of crowd-sourced assessments, we therefore only reject workers who we believe are almost certainly attempting to aggressively optimize earnings by gaming the system. Aggressive optimizers are identified by comparison of the worker’s mean score for reference translations, genuine system outputs, and bad reference translations. For example, random-clicker type aggressive optimizers commonly have extremely close mean scores for all three kinds of quality control items. An additional check is also put in place to search for score sequences within individual HITs with low variation. Another helpful tool to identify aggressive optimizers specifically with adequacy assessments, is that when a reference translation appears as the item to be assessed it is in fact identical to the reference translation displayed on screen. Although such items appear to be too obvious to be useful in assessments, they in fact act as a further modest hurdle for workers to meet, and caught many workers. For example, in some cases mean scores for degraded and genuine system outputs may be suspiciously close: mean scores for reference items for such workers often reveal that they rated identical items with low adequacy, and therefore can be rejected with confidence.

Not rejecting all workers whose assessments do not meet the quality control threshold unfortunately results in a set of HITs from workers whose data is not good enough to be useful in evaluations but nonetheless require payment. Table 1 shows numbers of HITs approved and rejected in experiments and corresponding numbers of HITs belonging to workers who passed quality control. Although proportions of low quality workers for whom we accept HITs varies from one language pair to the other, volumes of such workers are not so large for either language pair to be in any way considered prohibitively costly.

It may be worth noting, however, that in pilot posting of HITs on Mechanical Turk, we encountered a substantial increase in numbers of workers that fall within this costly group when we attempted to increase payment levels. In particular, when payment for HITs is increased to US$1 or more per HIT, this appears to attract substantially more workers who fall into this category. Although we have otherwise complied with ethics guidelines available to researchers, such as http://wiki.wearedynamo.org/index.php?title=Guidelines_for_Academic_Requesters, our experience when attempting to increase payment levels is that they (unfortunately) must remain low to avoid attracting large numbers of aggressive optimizers.
Fort, Adda and Cohen (2011) identify as a main cause of low payment to workers the lack of adequate worker reputation systems: without a method of accurately targeting good workers, increased payment simply attracts larger numbers of aggressive optimizers. The quality control mechanism we propose accurately discriminates good work from bad, and in this respect, could improve the situation for workers by facilitating such a reputation system. For example, although we do not refuse payment based solely on whether or not a particular worker meets the quality control threshold, we use to filter low quality data, information about the quality of work could be collected over the longer term and used to rate individual workers. This information could then be used by requesters, not as a strict cut-off, but to provide reliable fine-grained information about the quality likely to be produced by workers. Other ethical issues associated with crowd-sourcing – such as intellectual property, potential for states being deprived of tax payments, and unknown working conditions and rights of employment, due to the anonymous nature of crowd-sourcing services – remain a real concern but for the same reason are challenging to reliably investigate. Gupta et al. (2014) suggest that relationship-based crowd-sourcing could potentially be more fruitful than many current modes of operation, where relationships are maintained with a group of known good workers through e-mailing when batches of work are ready. Such an approach is compatible with our proposed methodology.

Languages that can be evaluated effectively through crowd-sourcing are of course limited to the native languages of speakers on a particular service. In this respect, previous efforts to elicit evaluations for Czech, for example, have resulted in so few genuine responses to HITs that we believe evaluation of this language pair by the crowd alone is currently not possible (Graham et al. 2012).

5 System-level evaluation

We have described a method for collecting fine-grained assessments of MT quality, and for filtering out unreliable workers. To investigate the utility and robustness of the proposed methodology, we replicate the human evaluation component of the WMT-12 shared translation task for Spanish and English in both translation directions, generating fresh crowd-sourced human assessments for all...
Y. Graham et al.

Table 2. Numbers of workers and translations, before and after quality control (broken down based on Assumption A only, and both Assumptions A&B).

<table>
<thead>
<tr>
<th>Evaluation modality</th>
<th>Workers</th>
<th>Translations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>A holds A&amp;B hold</td>
</tr>
<tr>
<td>ES- Fluency</td>
<td>102</td>
<td>54 (53%) 53 (52%)</td>
</tr>
<tr>
<td>Adequacy</td>
<td>319</td>
<td>94 (30%) 94 (30%)</td>
</tr>
<tr>
<td>EN-ES Fluency</td>
<td>37</td>
<td>22 (60%) 21 (57%)</td>
</tr>
<tr>
<td>Adequacy</td>
<td>45</td>
<td>21 (46%) 21 (46%)</td>
</tr>
</tbody>
</table>

participating systems. This allows the system rankings produced by the new evaluation methodology to be compared with those of the original shared task. In addition, we compute the degree to which statistically significant differences can be identified for pairs of participating systems.

5.1 Data

Samples of translations from the published WMT-12 shared task data set (Callison-Burch et al. 2012) for ES-EN and EN-ES translation were selected at random and evaluated by AMT workers. Twelve systems participated in the original ES-EN shared translation task, and eleven systems in EN-ES. Including quality control items, we collected a total of approximately 62k human fluency judgments and 82k human adequacy judgments.

Table 2 shows the number and percentage of workers who passed quality control, and the percentage of workers with no significant difference between mean scores for exact repeat items. Consistent with previous findings (Graham et al. 2013), the quality of English-speaking workers on AMT appears to be lower than for Spanish-speaking workers. Note that while we use the statistical tests described in the previous section to determine which HITs we use for the system evaluation, we do not use them as the basis for accepting/rejecting HITs – that is, for determining whether a given worker should be paid for performing an HIT. Our method of quality control is a high bar to reach, and workers might act in good faith and yet still not be consistent enough to meet the significance threshold we imposed. Instead, we individually examined mean score differences for reference translation, system output, and bad reference pairs, and declined payment only when there was no doubt that the response was either automatic or extremely careless.

After quality-control filtering over bad-reference pairs based on Assumption A, approximately 36k fluency judgments and 41k adequacy judgments remained. When we subsequently applied quality control based on Assumption B using exact-repeat translations, no additional HITs were filtered for adequacy for either language direction (that is, there were no instances of significant differences between score distributions for exact repeat items), whereas for fluency, two additional workers, one for each language direction, were filtered out.
Natural language engineering

Table 3. Average time per assessment (seconds) with fluency and adequacy direct estimate assessments and WMT-12 relative preference assessments

<table>
<thead>
<tr>
<th></th>
<th>Adequacy</th>
<th>Fluency</th>
<th>WMT-12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish-to-English</td>
<td>25.5</td>
<td>12.3</td>
<td>50.9</td>
</tr>
<tr>
<td>English-to-Spanish</td>
<td>29.0</td>
<td>12.0</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 4. Spanish-to-English mean human adequacy and fluency scores ('z' is the mean standardized z-score, and 'n' is the total number of judgments for that system after quality filtering is applied)

<table>
<thead>
<tr>
<th></th>
<th>Adequacy</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z</td>
<td>raw n</td>
</tr>
<tr>
<td>System A</td>
<td>0.21</td>
<td>65.7 1,328</td>
</tr>
<tr>
<td>System C</td>
<td>0.15</td>
<td>62.0 1,287</td>
</tr>
<tr>
<td>System B</td>
<td>0.12</td>
<td>61.7 1,270</td>
</tr>
<tr>
<td>System D</td>
<td>0.12</td>
<td>61.7 1,274</td>
</tr>
<tr>
<td>System E</td>
<td>0.10</td>
<td>60.6 1,256</td>
</tr>
<tr>
<td>System F</td>
<td>0.03</td>
<td>59.4 1,264</td>
</tr>
<tr>
<td>System H</td>
<td>0.03</td>
<td>59.4 1,232</td>
</tr>
<tr>
<td>System G</td>
<td>0.02</td>
<td>59.2 1,272</td>
</tr>
<tr>
<td>System J</td>
<td>–0.03</td>
<td>58.3 1,344</td>
</tr>
<tr>
<td>System K</td>
<td>–0.05</td>
<td>57.1 1,273</td>
</tr>
<tr>
<td>System L</td>
<td>–0.17</td>
<td>53.5 1,288</td>
</tr>
<tr>
<td>System M</td>
<td>–0.54</td>
<td>42.6 1,272</td>
</tr>
</tbody>
</table>

5.2 Evaluation of Spanish-to-English systems

Table 4 shows mean raw and standardized human adequacy and fluency scores for each participating WMT-12 system. Rows in the table are ordered from best to worst according to mean standardized adequacy scores, with mean standardized fluency scores used as a secondary key where adequacy scores agree to two decimal places.

Systems are anonymized with ordered names according to their rank order derived from subsequent significance tests (Figure 3), with ‘I’ unused to avoid confusion with the number ‘1’. Out-of-sequence system names in this table indicate a divergence between final conclusions based on combined adequacy/fluency significance tests (Figure 3) and the numeric ordering based on mean scores alone (this table).

Table 3 shows the average per translation time taken by assessors who passed our quality control requirements, and for the assessors involved in WMT-12. A comparison reveals a reduction in time taken to assess translations by approximately half when direct estimates are elicited by our monolingual set-up compared to relative preference judgments.

4 No information about WMT-12 times for English-to-Spanish is currently available.
For this language pair, the raw and standardized mean scores are closely correlated, a by-product of the structure of the HITs, which ensures that there is a relatively even distribution of systems across workers, as described in Section 3.4. If it had not been possible to put this constraint on assessments – for example, if assessments from two separate groups of systems and human judges were being combined – score standardization is likely to have a greater effect. In general, significance tests on standardized scores are more robust than on raw scores, and if a genuine difference in behavior exists, it is more likely to be identified in the standardized data.

Significance tests are used to estimate the degree to which rankings between pairs of systems are likely to have occurred simply by chance. For our human-sourced data, we apply a one-sided Wilcoxon rank-sum test to standardized score distributions for pairs of systems. Results of significance tests for all pairs of systems for ES-EN for standardized adequacy and fluency assessments are shown as the first and second heat maps in the lower half of Figure 3.

The third heat map in the lower half of the figure is a combined adequacy and fluency test, constructed as follows: if system X's adequacy score is significantly greater than that of system Y at some \( p \)-value, then the combined conclusion is...
that X is significantly better than Y at that p-value. If the outcome of a significance test for a pair of systems' adequacy score distributions is not significant at the desired significance level, we consider this to be a 'tie' in terms of adequacy, in which case the conclusions of a significance test on fluency assessments are used to derive the system ordering. In that case, conclusions from the significance test for fluency scores for that pair of systems should be taken as the overall outcome. For example, adequacy tests for Systems B and C show no significant difference in Figure 3, but tests on additional fluency assessments reveal that System B produces translations that are judged to be significantly more fluent than those of System C. System B is therefore regarded as being superior to System C. Note that this strategy of tie-breaking is only applied when there is no significant difference in adequacy at \( p < 0.05 \). As has already been noted, in our test environment, fluency alone cannot be used as criteria to rank systems. For example, although System J achieves higher fluency than System G in Figure 3, System G achieves significantly higher adequacy than System J, and in this pair it is System G that is regarded as superior.

The upper part of Figure 3 shows significance test results for the WMT-12 evaluation, computed on the published data set ('WMT-12 data set'), and on the official results ('WMT-12 paper'), the latter generated from the former by filtering via agreement with expert items (Callison-Burch et al. 2012). Since the original WMT-12 human evaluation took the form of relative preference judgments, we are unable to use the Wilcoxon rank sum test, and instead use a sign test to test for statistical significance in both the WMT-12 published data set and the official results. Since the rank order of systems is known at the time of significance testing, the application of a single instance of a one-tailed test to each pair of competing systems is appropriate, testing if the score distribution of the higher ranking system is significantly greater than that of the lower ranking system. All p-values reported in Figure 3 are for one-tailed tests, and therefore the heat map matrices in Figure 3 have maximally \( n(n - 1)/2 \) filled cells.

Comparing the combined adequacy–fluency significance matrix with those of the original relative preference evaluation, it is clear that a similar system ordering has emerged; but our methodology yields a higher proportion of significant differences between systems, and provides a more conclusive system ranking with fewer uncertainties. As a single exception to the overall pattern, System D was found to significantly outperform System B in the WMT evaluation, while we found no significant difference in adequacy, and in contrast to WMT results, that System B in fact outperformed System D in terms of fluency. The new evaluation data allows System A to be declared the outright winner for this language pair, with significantly higher adequacy scores than all other participating systems. This level of separation was not possible using the data generated by the original relative preference-based evaluation.

5.3 Evaluation of English-to-Spanish systems

Table 5 shows mean scores of each participating system in the WMT-12 EN-ES task, based on our evaluation methodology, with the table rows ordered using the
Table 5. English-to-Spanish mean human adequacy and fluency scores (‘z’ is the mean standardized z-score, and ‘n’ is the total number of judgments for that system after quality filtering is applied)

<table>
<thead>
<tr>
<th>System</th>
<th>Adequacy</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z</td>
<td>raw</td>
</tr>
<tr>
<td>System A</td>
<td>0.18</td>
<td>65.7</td>
</tr>
<tr>
<td>System B</td>
<td>0.05</td>
<td>61.8</td>
</tr>
<tr>
<td>System C</td>
<td>0.05</td>
<td>60.4</td>
</tr>
<tr>
<td>System E</td>
<td>0.05</td>
<td>61.4</td>
</tr>
<tr>
<td>System F</td>
<td>0.03</td>
<td>61.5</td>
</tr>
<tr>
<td>System D</td>
<td>0.02</td>
<td>60.3</td>
</tr>
<tr>
<td>System J</td>
<td>−0.03</td>
<td>58.3</td>
</tr>
<tr>
<td>System K</td>
<td>−0.07</td>
<td>56.7</td>
</tr>
<tr>
<td>System G</td>
<td>−0.08</td>
<td>57.0</td>
</tr>
<tr>
<td>System H</td>
<td>−0.10</td>
<td>57.3</td>
</tr>
<tr>
<td>System L</td>
<td>−0.11</td>
<td>57.2</td>
</tr>
</tbody>
</table>

mean standardized adequacy scores as a primary key, and the fluency scores as a secondary key. The same system ranking and naming convention is used as for Table 4.

For this language pair, differences in system rankings with respect to raw and standardized scores are again not large. Nonetheless, the standardized score ranking pushes System C up two places, as its raw mean of 60.4 is below those of System E (61.4) and System F (61.5). When computed on the (less-biased) standardized scores, the mean score for System C is now (roughly) equal to System E (0.05) and higher than System F (0.03).

As before, significance testing provides insight into the relative performance of the set of systems. Figure 4 parallels Figure 3, and includes results from the original WMT-12 shared task human evaluation used to determine the official results (‘WMT-12 paper’), and results based on the published WMT-12 data set (‘WMT-12 data set’); to provide a meaningful comparison, all tests in Figure 4 are one-tailed. In this translation environment, the fluency scores allow several ties in adequacy to be broken, including the five-way tie for second place between Systems B, C, D, E, and F. For EN-ES, the new evaluation methodology again produces almost a superset of official WMT-12 results but with more certainty, this time with three exceptions. In the original results, System E was shown to beat System F, System J to beat System K, and System L to beat System K. None of these conclusions are supported by the results of the new methodology.

Overall, System A is identified as an outright winner in our experimentation. In contrast, the original relative preference-based human evaluation was unable to identify any single system that significantly outperformed all others. Moreover, the outright winner identified for this EN-ES language pair, System A, is the same system identified for ES-EN. In addition, the systems taking second and third place for the two translation directions are also matched. That is, System B and System C
5.4 Varying the number of assessments

The results described above suggest that our crowd-sourced approach to gathering judgments achieves higher levels of system discrimination than the approach used at WMT-12. In addition, there was a relatively high level of overall concord between the fluency and adequacy judgments, allowing us to break ties in adequacy based on fluency. However, our experiments made use of more judgments than the WMT-12 evaluation, and it could be that the larger pool of judgments collected from the AMT workers is the factor responsible for the greater certainty. We investigate that possibility by reducing the number of HITs used in our analyses.

Since our method diverges considerably from the original WMT-12 approach to collecting assessments, determining the correct number of judgments to use for the purpose of comparison is not entirely straightforward. In the WMT-12 evaluation, five translations are assessed per screen, and ordered from best to worst to generate ten pairwise labels. To order those five competing translations, it seems likely that
Table 6. Proportions of significant differences between system pairs identified at different significance thresholds, using the WMT-12 relative preference judgments, and the new direct estimate method.

<table>
<thead>
<tr>
<th></th>
<th>Spanish-to-English</th>
<th>English-to-Spanish</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Relative preference</td>
<td>Direct estimate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 0.05$</td>
<td>19.7%</td>
<td>69.7%</td>
</tr>
<tr>
<td>$p = 0.01$</td>
<td>6.1%</td>
<td>56.1%</td>
</tr>
<tr>
<td>$p &lt; 0.001$</td>
<td>–</td>
<td>39.4%</td>
</tr>
</tbody>
</table>

Each translation is read more than once. If the average number of readings is two, then ten labels are generated each time ten translations are read. On the other hand, in the direct estimate evaluation, ten labels are produced by reading and assessing ten translations, with only limited re-reading anticipated. We therefore compare methods based on numbers of labels produced by each method, as a reasonable estimate of the effort involved in generating them.

Table 6 shows the proportion of significant differences identified by our direct estimate-based evaluation compared to those of the original WMT-12 relative preference evaluation, for equal numbers of judgments. The number of judgments for the direct estimate method is limited to the number in the official WMT-12 data set, namely, 475 judgments per system for the 12 ES-EN systems, and 670 judgments per system for the 11 EN-ES systems. In the latter case, the subset of direct estimate judgments was made by taking HITs in the order they were completed by AMT workers.

These results show that our method reveals substantially higher proportions of significant differences between pairs of participating systems compared to the original relative preference evaluation. At $p < 0.05$, for example, for the same number of judgments, our direct estimate evaluation produces approximately 3.5 times the number of significant differences between pairs of ES-EN systems, and close to 4 times as many significant differences for EN-ES systems. At the higher confidence level of $p < 0.01$, the relative proportion of significant differences detected by the two methods diverges even further, with our direct estimate method identifying 9–10 times the proportion of significant differences identified by relative preference judgments.

The heat maps in Figure 5 provide additional insight into the relationship between the number of judgments per system and the level of statistical significance for different system pairings, based on direct estimation methodology. As few as 300 crowd-generated adequacy judgments per system is sufficient for mid-range and low-range systems to be separated from the better-performing ones. Note that even the last 80 judgments collected, from 1,200 per system on average to 1,280 per system on average (which is what is shown in Figure 3), are helpful in distinguishing between systems.
The aim of the evaluation we have detailed is to provide an accurate mechanism for evaluation of MT on the system-level. Although ratings for translations assessed on a continuous rating scale can be expected to contain random variations in individual scores, when ratings are collected for a sufficiently large sample of translations belonging to a given system, positive and negative random errors present in individual assessments cancel out to produce accurate mean and median scores for systems. For evaluations to be accurate at the segment-level, a different approach is required, where assessments are repeated per segment as opposed to per system; see Graham, Baldwin and Mathur (2015) for further details.

6 Conclusion

We have presented a new methodology for human evaluation of MT quality. To our knowledge, this is the first method that relies entirely on assessments sourced from the crowd, in our case using Amazon's Mechanical Turk. Our approach is based on actively removing sources of bias, including mechanisms to accommodate assessors with consistent individual scoring strategies. By restructuring the task as an assessment of monolingual similarity of meaning, assessing individual translations, and separating fluency and adequacy, the task was made substantially less cognitively taxing, and allowed participation by much larger pools of workers.

To assess the feasibility of our methodology for large-scale MT evaluation, we replicated the WMT-12 shared task human evaluation for two language pairs, using the system translations released by the task organizers. Our results show that capturing direct estimates of translation quality on a continuous rating scale leads to more informative judgments that reflect not only the fact that one translation is superior to another, but also the degree to which it was preferred. In addition, direct estimates provide the advantage of a straightforward combination of individual standardized scores for translations into mean system scores that, given large numbers of assessments, are more precise. The same logic does not apply to the same degree to methods based on combinations of relative preference judgments. Once the quality control mechanisms were applied, high consistency between workers was observed. Score standardization at the worker level was also helpful.
With our new assessments, score distributions for translations of competing systems provide useful discrimination between systems, and we identified substantially higher proportions of significant differences. More conclusive rankings not only provide greater insight into the relative performance of MT systems, but also establish a better foundation for evaluation of automatic metrics. The evaluation methodology is scalable, and highly efficient and cost-effective: the judgments used in this paper were collected at approximately US$40 per system, for each language pair and evaluation modality (EN-ES or ES-EN, and adequacy or fluency). With greater consistency per assessment than previous approaches, and clearer rankings across the systems considered, we have answered in the affirmative the question we posed as the title of this paper: MT systems can indeed be evaluated by the crowd alone.

Acknowledgments
This work was supported by the Australian Research Council’s Discovery Projects Scheme (grant DP110101934) and Science Foundation Ireland through the CNGL Programme (Grant 12/CE/12267) in the ADAPT Centre (www.adaptcentre.ie) at Trinity College Dublin.

References
on Computational Linguistics and the Association for Computational Linguistics, Sydney, Australia, pp. 249–56.


28 Y. Graham et al.


CUNI Experiments for WMT17 Metrics Task

David Mareček Ondřej Bojar
Ondřej Hubsch Rudolf Rosa Dušan Varioš

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
surname@ufal.mff.cuni.cz, except hubschondrej@gmail.com

Abstract

In this paper, we propose three different methods for automatic evaluation of the machine translation (MT) quality. Two of the metrics are trainable on direct-assessment scores and two of them use dependency structures. The trainable metric AutoDA, which uses deep-syntactic features, achieved better correlation with humans compared e.g. to the chrF3 metric.

1 Introduction

With the ongoing research of the machine translation (MT) systems in the past the need for accurate automatic evaluation of the translation quality became unquestionable. Even though the human judgment of the MT system outputs still holds as the most reliable form of evaluation, the high cost of human evaluation together with the amount of time required for such evaluation makes human judgment unsuitable for large scale experiments where we need to evaluate many different system configurations in a relatively short timespan. An additional important limitation of human evaluation is that it cannot be exactly repeated. This led to development of various methods for automatic MT evaluation in the past with the aim to eliminate the need for the expensive human assessment of the developed MT systems.

In this paper we suggest three novel methods for automatic MT evaluation together with their direct comparison:

1. AutoDA: A linear regression model using semantic features trained on WMT Direct Assessment scores (Bojar et al., 2016) or HUMEseg scores (Birch et al., 2016).
2. TreeAggreg: N-gram based metric computed over aligned syntactic structures instead of the linear representation of the translated sentences.
3. NMTScorer: A neural sequence classifier which assigns correct/incorrect flags to the evaluated sentence segments.

Table 1 shows the main properties of the proposed methods. Some of them were mainly developed for Czech as the target language and were later modified to be applied to other languages. The differences in the data preprocessing and their impact on the resulting evaluator are also described in this paper.

2 AutoDA: Automatic Direct Assessment

AutoDA is a sentence-level metric trainable on any direct assessment scores. The metric is based on a simple linear regression combining several features extracted from the automatically aligned translation-reference pair. There may be also other established metrics within the features.

The training data with golden direct-assessment scores available are shown in Table 2.

We describe two variants. The first one works only on Czech and uses many semantic features based on rich Czech tectogrammatical annotation (Böhmová et al., 2003). The second one uses much fewer features, however, it is language universal and needs only a dependency parsing model available.

2.1 AutoDA Using Czech Tectogrammatics

This metric automatically parses the Czech translation candidate and the reference translation and uses various semantic features to compute the final score.

2.1.1 Word Alignment

AutoDA relies on automatic alignment between the translation candidate and the reference trans-
Table 1: Overview of the examined methods. Currently, AutoDA uses only monolingual resources even though extracting additional features from the bilingual data (*) is possible. TreeAggreg can use any string-level metric for score computation instead of ChrF (**).

<table>
<thead>
<tr>
<th>Method</th>
<th>Resource Type</th>
<th>Trainable</th>
<th>Metric Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AutoDA</td>
<td>Monolingual/Bilingual*</td>
<td>Yes</td>
<td>Segment-level Linear Regression</td>
</tr>
<tr>
<td>TreeAggreg</td>
<td>Monolingual</td>
<td>No</td>
<td>Tree Segment-level ChrF**</td>
</tr>
<tr>
<td>NMTScorer</td>
<td>Bilingual</td>
<td>Yes</td>
<td>Segment-level Classification</td>
</tr>
</tbody>
</table>

Table 2: Overview of the available data for training AutoDA.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>Target</th>
<th># Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT16 DAsseg</td>
<td>TR/FI/CS/RO/RI/DE</td>
<td>EN</td>
<td>560</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RU</td>
<td></td>
</tr>
<tr>
<td>WMT15 DAsseg</td>
<td>DE/RI/FI/CS</td>
<td>EN</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RU</td>
<td></td>
</tr>
<tr>
<td>WMT16 HUMESeg</td>
<td>EN</td>
<td>CS/DE/PL/RO</td>
<td>~350</td>
</tr>
</tbody>
</table>

2.1.2 Tectogrammatic Parsing
We use Treex framework (Popel and Žabokrtský, 2010) to do the tagging, parsing and tectogrammatical annotation. Tectogrammatical annotation of sentence is a dependency tree, in which only content words are represented by nodes. The main label of the node is a tectogrammatical lemma – mostly the same as the morphological lemma, sometimes combined with a function word in case it changes its meaning. Other function words and grammatical features of the words are expressed by other attributes of the tectogrammatical node. An example of a pair of tectogrammatical trees is provided in Figure 1. The main attributes are:

- **tectogrammatical lemma (t-lemma)**: the lexical value of the node,
- **functor**: the semantic value of the syntactic dependency relation. Functors express the functions of individual modifications in the sentence, e.g. ACT (Actor), PAT (Patient), ADDR (Addressee), LOC (Location), MANN (Manner),
- **sempos**: semantic part of speech: n (noun), adj (adjective), v (verb), or adv (adverbial),
- **formeme**: morphosyntactic form of the node. The formeme includes for example prepositions and cases of the nouns, e.g. n:jako+1 for nominative case with preposition jako.
- **grammatemes**: tectogrammatical counterparts of morphological categories, such as number, gender, person, negation, modality, aspect, etc.

2.1.3 Scores for Matching Attributes Ratios
Given the word- (or node-) alignment links between tectogrammatical annotations of the translation and reference sentences, we can count the percentage of links where individual attributes agree, e.g. the number of pairs of tectogrammatical nodes that have the same tectogrammatical lemma. These scores capture only a portion of what the tectogrammatical annotations offer, for instance, we they do not consider the structure of the trees at all. For the time being, we take these scores as individual features and use them in a combined model.
2.1.4 Linear Regression Training

We collect 83 various features based on matching tectogrammatical attributes computed on all nodes or a subsets defined by particular semantic part-of-speech tags. To this set of features, we add two BLEU scores (Papineni et al., 2002) computed on forms and on lemmas and two chrF3 scores (Popovic, 2015) computed on trigrams and sixgrams, so we have 87 features in total.

We train a linear regression model to obtain a weighted mix of features that fits best the WMT16 HUMEseg scores. Since the amount of annotated data available is low, we use the jackknife strategy:

- We split the annotated data into ten parts.
- For each tenth, we train the regression on all the rest data and apply it to this tenth.

By this procedure, we obtain automatically assigned scores for all sentences in the data. The correlation coefficients are shown in Table 3, along with the individual features.

In addition to the regression using all 87 features, we also did a feature selection, in which we manually chose only 23 features with a positive impact on the overall correlation score. For instance, we found that the BLEU scores can be easily omitted without worsening the correlation. Conversely, the chrF scores are very valuable and omitting them would lower the correlation significantly.

We see that chrF3 alone performs reasonably well (Pearson of 0.54), If we combine it with a selected subset our features, we are able to achieve the correlation of up to 0.659.
2.2 Language Universal AutoDA

We have seen that deep-syntactic features help to train an automatic metric with higher correlation for Czech. Even though we have no similar tools for other languages so far, we try to extract similar features for them as well. The source code is available online.  

2.2.1 Universal Parsing

We use Universal Dependencies (UD) by Nivre et al. (2016b), a collection of treebanks in a common annotation style, where all our testing languages are present – version 1.3 covers 40 languages (Nivre et al., 2016a). For syntactic analysis, we use UDPipe by Straka et al. (2016), a tokenizer, tagger, and parser in one tool, which is trained on UD. The UD tagset consists of 17 POS tags; the big advantage is that the tagset is the same for all the languages and therefore we can easily extract e.g. content words, prepositional phrases, etc.

2.2.2 Monolingual Alignment

Unlike from Czech, we did not known about the existing corpus of paraphrases available across other languages, so we used a simple monolingual aligner based on word similarities and relative positions in the sentence. Our implementation is inspired by the heuristic Monolingual Greedy Aligner written by Martin Popel (Rosa et al., 2012), which is available in the Treeex framework.  

First, we compute scores for all possible alignment connections between tokens of the reference and translated sentence:

\[
\text{score}(i, j) = w_1 \text{JaroWinkler}(W_i^r, W_j^t) + w_2 I(T_i^r = T_j^t) + w_3 (1 - |\frac{i}{\text{len}(r)} - \frac{j}{\text{len}(r)}|),
\]

where JaroWinkler($W_i^r, W_j^t$) defines similarity between the given words (Winkler, 1990), $I(T_i^r = T_j^t)$ is a binary indicator testing the identity of POS tags, and $(1 - |\frac{i}{\text{len}(r)} - \frac{j}{\text{len}(r)}|)$ tells us how close are the two words according to their relative positions in the sentences. The weights were set manually to $w_1 = 8$, $w_2 = 3$, and $w_3 = 3$; they were not tuned for this specific task. When we have the scores, we can simply produce unidirectional alignments (i.e. find the best token in the translation for each token in the reference and vice versa) and then symmetrize them to create intersection (one-to-one) or union (many-to-many) alignments. We finally use union symmetrization, since it achieved slightly better correlation with humans.

2.2.3 Extracting Features

We distinguish content words from function ones by the POS tag. The tags for nouns (NOUN, PROPN), verbs (VERB), adjectives (ADJ), and adverbs (ADV) correspond more or less to content words. Then there are pronouns (PRON), symbols (SYM), and other (X), which may be sometimes content words as well, but we do not count them. The rest of POS tags represent function words.

Now, using the alignment links and the content words, we can compute numbers of matching content word forms and matching content word lemmas. The universal annotations contains also morphological features of words: case, number, tense, etc. Therefore, we also create equivalents of tectogrammatical for memes or grammatemes. Our features can thus check for instance the percentage of aligned words with matching morphological number or tense.

2.2.4 Regression and Results

We compute all the scores proposed in the previous section on the four languages and test the correlation on WMT16 HUMEseg dataset (Birch et al., 2016). German UD annotation does not contain lemmas and morphological features, so some scores for German could not be computed. Similarly as in Section 2.1.4, we trained a linear regression on all the features together with chrF3 score. The results computed by 10-fold cross-validation on WMT16 HUMEseg dataset and comparison with chrF and NIST scores is shown in Table 4.

3 Tree Aggregated Evaluation

TreeAgg is a simple sentence-level metric, remotely inspired by HUME. Rather than being a full standalone metric, it can be regarded as

\footnote{Multilingual corpus of paraphrases has been released by Chris Callison-Burch’s group and is available here: \url{http://paraphrase.org/\#download}}
Table 4: Pearson correlations of different sentence-level metrics on WMT16 HUMEseg dataset. Standard NIST and chrF metrics are compared with our individual features matching. AutoDA combines all the features together with the chrF3 score and the NIST score computed on content lemmas only. Other NIST scores are not included in AutoDA, since they do not bring any improvement.

<table>
<thead>
<tr>
<th>Metric</th>
<th>en-cs</th>
<th>en-de</th>
<th>en-pl</th>
<th>en-ro</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST</td>
<td>0.436</td>
<td>0.481</td>
<td>0.418</td>
<td>0.611</td>
</tr>
<tr>
<td>NIST cased</td>
<td>0.421</td>
<td>0.481</td>
<td>0.410</td>
<td>0.611</td>
</tr>
<tr>
<td>chrF1</td>
<td>0.505</td>
<td>0.497</td>
<td>0.428</td>
<td>0.608</td>
</tr>
<tr>
<td>chrF3</td>
<td>0.540-0.511</td>
<td>0.419-0.638</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIST on content lemmas</td>
<td>0.416</td>
<td>–</td>
<td>0.361</td>
<td>0.542</td>
</tr>
<tr>
<td>matching lemmas</td>
<td>0.431</td>
<td>–</td>
<td>0.393</td>
<td>0.565</td>
</tr>
<tr>
<td>matching forms</td>
<td>0.372</td>
<td>0.478</td>
<td>0.405</td>
<td>0.576</td>
</tr>
<tr>
<td>matching content lemmas</td>
<td>0.359</td>
<td>–</td>
<td>0.408</td>
<td>0.536</td>
</tr>
<tr>
<td>matching content forms</td>
<td>0.321</td>
<td>0.470</td>
<td>0.427</td>
<td>0.552</td>
</tr>
<tr>
<td>matching formemes</td>
<td>0.347</td>
<td>0.170</td>
<td>0.357</td>
<td>0.420</td>
</tr>
<tr>
<td>matching tense</td>
<td>-0.094</td>
<td>–</td>
<td>-0.118</td>
<td>0.079</td>
</tr>
<tr>
<td>matching number</td>
<td>0.286</td>
<td>–</td>
<td>0.205</td>
<td>0.404</td>
</tr>
<tr>
<td>AutoDA (linear regression)</td>
<td>0.604</td>
<td>0.525</td>
<td>0.453</td>
<td>0.656</td>
</tr>
</tbody>
</table>

To be able to apply TreeAggreg to measuring the correspondence of a translation \( t \) to the reference \( r \), we first need to apply a set of NLP tools in a pre-processing pipeline:

1. align reference and translation
2. parse reference
3. parse translation

Next, both the reference and the translation are split into the following types of segments:

- **Whole sentence**: This is simply the base string-based MT metric applied in the standard way.
- **Sentence root**: The sentence root is selected according to the parse trees; usually this is the main verb in the sentence.
- **Subtree spans**: As we expect the dependency analysis of the reference to be much more accurate than that of the translation, we only use the reference parse tree to identify the root dependents’
spans, and the word alignment to identify the corresponding spans in the translation:

- $p_i$ contains all words from $s_i$ that are transitively dependent on $d_i$; the $i^{th}$ dependent of $r_i$; $p_i$ includes $d_i$ but excludes $r_i$
- $p'_i$ contains the first and last word from $s_i$, which are aligned to any of the words in $p_i$, and all of the words between them

The string-level metric $m(r, t)$ is then computed on each corresponding pair of the reference and translation segments. A weighted average of the segment-level scores is computed, where longer segments are given higher weight: the weight is the sum of the numbers of words in the reference segment and in the translation segment. Additionally, for the $(s_i, t_i)$ segment pair, which is still the most important component of the metric, we use a double weight. Thus, the final score $m$ is computed as follows:

$$m_s = m(s_i, s_j) \cdot (|s_i| + |s_j|) \cdot 2$$
$$m_r = m(r_i, r_j) \cdot 2$$
$$m_p^s = m(p_i', p_j') \cdot (|p_i'| + |p_j'|)$$
$$m_p^t = m(p_i', p_j') \cdot (|p_i| + |p_j|)$$
$$m = \frac{m_s + m_r + \sum_{i \in Dep(r_i)} m_p^i}{2(|s_i| + 2|s_j| + 2 + \sum_{i \in Dep(r_j)} |p_i'| + |p_j|)}$$

$Dep(r_i)$ are all immediate dependents of $r_i$.

### 3.2 Development

When developing the TreeAggreg metric, we tried multiple configurations, evaluating each of them on the WMT16 HUMEseg dataset for correlation with human judgments, and then selected the one that performed best, which we have just described.

For example, we also experimented with more fine-grained segmentations, such as taking each node together only with its immediate dependents as a span. However, such setups performed poorer, probably because they depend more heavily on the high structural similarity of the translation to the reference. Still, it seems reasonable to assume that at least the arguments of the root node should usually correspond well between the reference and the candidate translation.

We also tried to put more weight to certain words that we expected to be more important, such as $d_i$ (immediate dependents of the root $r_i$). However, this always led to a deterioration in the correlation of the metric to human judgments. Thus, an important property of our metric seems to be that each reference word is taken into account exactly twice.\(^7\)

### 3.3 Evaluation

To evaluate our metric, we measured Pearson's correlation of chrF3-based TreeAggreg scores with sentence-level human judgments on the WMT16 HUMEseg dataset. For comparison, we also measured the correlation of a baseline metric, which is the vanilla sentence-level chrF3. As shown in Table 5, our metric performs comparably to the chrF3 baseline, leading to a slight improvement for two language pairs, and a slight deterioration for the other two.

Thus, our approach of employing sentence syntactic structure into a string-based MT metric seems to affect the metric only minimally. Moreover, the TreeAggreg metric was developed and evaluated on the same data and therefore the comparison in Table 5 is not quite fair, however, the number of configurations tested was very little.

### 4 Neural MT Scorer

Neural MT Scorer is a model that predicts a probability for a given source/target translation pair using a simplified architecture that is based on existing NMT models with attention. The predicted number should reflect how much the meaning of source and target matches. We used that model for a different task (scoring phrase table entries in PBMT) where it performed well. Note that as of now, Neural MT Scorer indeed does not make any use of the reference translation, so it is effectively a quality estimation method.

The training data for the model are bilingual corpus (set of sentences that should be classified into a different category).

\(^7\)The same holds for words in the translation only if the $p'_i$ spans do not overlap, are contiguous, include both the first and the last word in the sentence, and do not include $r_i$. 

---

**Table 5: Evaluation of TreeAggreg (our metric) and chrF3 (baseline) with Pearson’s correlation to human judgments.**

<table>
<thead>
<tr>
<th>Lang</th>
<th>chrF3</th>
<th>TreeAggreg</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-cs</td>
<td>0.5403</td>
<td>0.5873</td>
<td>+0.0070</td>
</tr>
<tr>
<td>en-de</td>
<td>0.5111</td>
<td>0.5078</td>
<td>−0.0033</td>
</tr>
<tr>
<td>en-pl</td>
<td>0.4186</td>
<td>0.4266</td>
<td>+0.0080</td>
</tr>
<tr>
<td>en-ro</td>
<td>0.6314</td>
<td>0.6226</td>
<td>−0.0088</td>
</tr>
</tbody>
</table>

Average 0.5254 \(±0.0007\)
as entirely correct) as well as a set of sentences that should be classified as incorrect (we obtain these by performing some random operations on the bilingual corpus). We do not train it on data specific for the metrics task (i.e. the model is only trained to recognize correct and incorrect translations, but small differences among different translations of the same sentence might not be recognized), therefore there is a room for potential improvement.

We do not use any smoother labeling than 0/1 (correct/incorrect), since even a single word omission may cause completely different meaning of the sentence. At inference time, the output is a float number between 0 and 1.

4.1 Architecture
We use two LSTM encoders, one for source and one for target side. The vector representations of the source words are fed into the source LSTM encoder to obtain one representation $p_s$ of the entire sentence. Also, the intermediate outputs of the source LSTM encoder are used in an attentional layer when processing the target sentence in the target LSTM encoder. The final cell states $p_s$ and $p_t$ are used to measure the bilingual similarity by $\sigma(p_s^T, p_t)$. The entire architecture is very similar to (Bahdanau et al., 2014), except that we use the attention mechanism while encoding the target side.

Note that there is also no softmax layer over the word dictionary – we know the entire source and target sentences and so we do not need to predict the next word; we just need one score between 0 and 1. This should allow for faster training of the model; however, we need to provide labeled training data. We currently generate wrong sentences using these basic operations:

- change a few words to completely random ones from the source/target dictionary
- take a translation of a completely different sentence
- utilize WordNet to change the polarity of a sentence
- remove/add some random words at a random place

4.2 Evaluation
We evaluated the model on the WMT16 HUMESeg dataset, but currently it performs poorly. It should be possible to improve it significantly by optimizing the training process for the metrics task (for example by adding another layer that uses the final representations $p_s$ and $p_t$ to predict human scores and finetune the entire model on some manually evaluated datasets). The Pearson correlation coefficients to human judgements are shown in Table 6.

5 Conclusion
We presented three metrics. AutoDA is a trainable metric combining syntactic features matching and chrF and naturally significantly outperforms chrF on all four tested languages.

In TreeAggreg, we tried to enrich a string-based MT metric with light-weight information about the syntactic structure of the sentences, but the results seem rather disappointing.

NMTScorer in which we used two LSTM encoders for source sentence and candidate translation and predicted sentence similarity also did not prove to work well.

Acknowledgments
This work has been in part supported by the European Union’s Horizon 2020 research and innovation programme under grant agreements No 644402 (HimL) and 645452 (QT21), by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16_013/0001781), by the grant No. DG16P02B048 of the Ministry of Culture of the Czech Republic and by the SVV 260 453 grant.

References

<table>
<thead>
<tr>
<th>Languages</th>
<th>NMT Scorer</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-cs</td>
<td>0.4099</td>
</tr>
<tr>
<td>en-de</td>
<td>0.3462</td>
</tr>
<tr>
<td>en-pl</td>
<td>0.3261</td>
</tr>
<tr>
<td>en-ro</td>
<td>0.4792</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>0.3903</strong></td>
</tr>
</tbody>
</table>

Table 6: Evaluation of NMT Scorer with Pearson correlation to human judgments.


Further Investigation into Reference Bias in Monolingual Evaluation of Machine Translation

Qingsong Ma∗† Yvette Graham† Timothy Baldwin‡ Qun Liu†

∗Institute of Computing Technology Chinese Academy of Sciences
†ADAPT Centre Dublin City University
‡Computing and Info Systems University of Melbourne
naqingsong@ict.ac.cnfirstname.surname@dcu.ie tb@ldwin.net

Abstract
Monolingual evaluation of Machine Translation (MT) aims to simplify human assessment by requiring assessors to compare the meaning of the MT output with a reference translation, opening up the task to a much larger pool of genuinely qualified evaluators. Monolingual evaluation runs the risk, however, of bias in favour of MT systems that happen to produce translations superficially similar to the reference and, consistent with this intuition, previous investigations have concluded monolingual assessment to be strongly biased in this respect. On re-examination of past analyses, we identify a series of potential analytical errors that force some important questions to be raised about the reliability of past conclusions, however. We subsequently carry out further investigation into reference bias via direct human assessment of MT adequacy via quality controlled crowd-sourcing. Contrary to both intuition and past conclusions, results show no significant evidence of reference bias in monolingual evaluation of MT.

1 Introduction

Despite it being known for some time now that automatic metrics, such as BLEU (Papineni et al., 2002), provide a less than perfect substitute for human assessment (Callison-Burch et al., 2006), evaluation in MT more often than not still comprises BLEU scores. Besides increased time and resources required by the alternative, human evaluation of systems, human assessment of MT faces additional challenges, in particular the fact that human assessors of translation quality tend to be highly inconsistent. In recent Conference on Machine Translation (WMT) shared tasks, for example, manual evaluators complete a relative ranking (RR) of the output of five alternate MT systems, where they must rank the quality of competing translations from best to worst. Within this set-up, when presented with the same pair of MT output translations, human assessors often disagree with one another’s preference, and even their own previous judgment about which translation is better (Callison-Burch et al., 2007; Bojar et al., 2016). Low levels of inter-annotator agreement in human evaluation of MT not only cause problems with respect to the reliability of MT system evaluations, but unfortunately have an additional knock-on effect with respect to the meta-evaluation of metrics, in providing an unstable gold standard. As such, provision of a fair and reliable human evaluation of MT remains a high priority for empirical evaluation.

Direct assessment (DA) (Graham et al., 2013, 2014, 2016) is a relatively new human evaluation approach that overcomes previous challenges with respect to lack of reliability of human judges. DA collects assessments of translations separately in the form of both fluency and adequacy on a 0–100 rating scale, and, by combination of repeat judgments for translations, produces scores that have been shown to be highly reliable in self-replication experiments (Graham et al., 2015). The main component of DA used to provide a primary ranking of systems is adequacy, where the MT output is assessed via a monolingual similarity of meaning assessment. A reference translation is displayed to the human assessor (rendered in gray) and below it the MT output (in black), with the human judge asked to state the degree to which they agree that The black text adequately expresses the meaning of the gray text in English. The motivation behind
constructing DA as a monolingual MT evaluation are as follows:

- Monolingual assessment of MT opens up the annotation task to a larger pool of genuinely qualified human assessors;
- Crowd-sourced workers are unlikely to make use of information that is not entirely necessary for completing a given task; and are therefore unlikely to use the source language input if the reference is also displayed or to make use of the source input inconsistently;
- Displaying only the source without a reference greatly increases both the difficulty of the task and the time required to complete each annotation, which is too serious a trade-off when we wish to carry out human assessment on a very large scale;
- Varying levels of proficiency in the source language across different human assessors could contribute to inconsistency in bilingual MT evaluations.

Although DA has been shown to overcome the long-standing challenge of lack of reliability in human evaluation of MT, the possibility still exists that, although scores collected with DA have been shown to be almost perfectly reliable in self-replication experiments, both sets of scores, although consistent with each other, could in fact both be biased in the same way. Graham et al. (2013) include in the design of DA a number of criteria aimed at minimizing such bias: (i) assessment of individual translations in isolation from others to avoid a given system being scored unfairly low due to its translations being assessed more frequently alongside high quality translations (Bojar et al., 2011); (ii) eliciting assessment scores via a Likert-style question without intermediate labeling, motivated by medical research showing patients’ ratings of their own health to be highly dependent on the exact wording of descriptors (Seymour et al., 1985); (iii) accurate quality control by assessing the consistency of judges with reference only to their own rating distributions, to accurately remove inconsistent crowd-sourced data while avoiding removal of data that legitimately diverges from the scoring strategy of a given expert judge; and (iv) score standardization to avoid bias introduced by legitimate variations in scoring strategies.

Despite efforts to avoid bias in Graham et al. (2013), since DA is a monolingual evaluation of MT that operates via comparison of MT output with a reference translation, it is therefore still possible, while avoiding other sources of bias, that DA incurs reference bias where the level of superficial similarity of translations with reference translations results in an unfair gain, or indeed an unfair disadvantage for systems that yield translations that legitimately deviate from the surface form of reference translations. Following this intuition, Fomicheva and Specia (2016) carry out an investigation into bias in monolingual evaluation of MT and conclude that in a monolingual setting, human assessors of MT are strongly biased by the reference translation. In this paper, we provide further analysis of experiments originally provided in Fomicheva and Specia (2016), in addition to further investigation into the degree to which the intuition about reference bias can be supported.

2 Background

Fomicheva and Specia (2016) provide an investigation into reference bias in monolingual evaluation of MT. 100 Chinese to English MT output translations are assessed by 25 human judges on a five-point scale, in the form of their response (None, Little, Much, Most, or All) to the following question: how much of the meaning of the human translation is also expressed in the machine translation?: Precisely the same 100 translations were assessed by all 25 judges. Human judges were divided into five groups of five: Group 1 (G₁) was shown the source language input and the MT output only and carried out a bilingual assessment, while Groups 2–5 (G₂–G₅) were not shown the source input but instead compared the MT output to a human-generated reference translation. A distinct set of reference translations was assigned to each group G₂–G₅. Inter-annotator agreement (IAA) was measured for pairs of judges as follows (the total number of judge pairs resulting from each setting is provided in parentheses):

- SOURCE: a given pair of judges assessed translations in a bilingual setting (all possible pairs within G₁ = \( \binom{25}{2} = 10 \) pairs);
- SAME: a given pair of judges assessed translations in a monolingual setting by comparison with precisely the same reference translation (the sum of all possible pairs result-
<table>
<thead>
<tr>
<th>DIFF</th>
<th>SAME</th>
<th>SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.163 ± 0.01</td>
<td>0.197 ± 0.01</td>
<td>0.190 ± 0.02</td>
</tr>
</tbody>
</table>

Table 1: Average Kappa coefficients and 99% confidence intervals reported in Fomicheva and Specia (2016)

Table 1 provides a clear indication of the precise approach to CI estimation attempted in Fomicheva and Specia (2016) is unfortunately not explicitly stated but out of the range of methods that exist the approach that is applied most resembles bootstrap resampling. Conventional speaking, bootstrap resampling can be applied to CI estimation of a point estimate for a sample, D, of size N, by simulating the variance in the population sampling distribution (Efron and Tibshirani, 1993). A standard method of estimating CIs via bootstrap resampling is to generate a bootstrap distribution for the statistic of interest made up of M repeat computations of it, each time drawing a random sample of size N from D with replacement. Although most similar to bootstrap resampling, the application in Fomicheva and Specia (2016) to CI estimation of Kappa coefficients diverges in some important ways from a standard application, however.

We therefore provide a comparison of the analysis drawn in Fomicheva and Specia (2016) with a standard bootstrap implementation.

Figure 1(a) shows SAME and DIFF bootstrap distributions, reproduced from code released with Fomicheva and Specia (2016), originally yielding non-overlapping CIs that led to the conclusion of strong reference bias. Although the level of statistical significance is reported to be 99%, CIs in Figure 1(a) show that the proportion of each bootstrap distribution was substantially underestimated leading to overly narrow CI limits for both SAME and DIFF. In contrast, Figure 1(b) shows CIs resulting from an accurately computed proportion of 95% of the same bootstrap distribution, where even at the lower level of 95% significance (as opposed to 99%) CIs for SAME and DIFF now overlap, reversing the conclusion of strong reference bias.

In addition, CI estimation diverges from bootstrap resampling with respect to the number of bootstrap samples employed. Since there are a total of $N^M$ possible distinct bootstrap samples for a given sample D (taking order into account), in a conventional bootstrap implementation a Monte Carlo approximation of size M is employed, and the larger M is, the closer the CI distribution approaches the true bootstrap distribution (Chernick and LaBudde, 2014). In Fomicheva and Specia (2016), CIs are computed via only 50 bootstrap samples, however. Figure 1(c) shows the change in location of CIs for a typical M = 1,000, as opposed to M = 50 (Figure 1(b)).

---

1We provide a re-analysis of experiment data specifically with respect to Cohen’s Kappa. All errors outlined for Cohen’s Kappa also lead to the same inaccuracies for weighted Kappa in Fomicheva and Specia (2016), however.

2We note that M is described as 100 in the publication, but 50 in the released code. Our question raised about the methodology also stands for M = 100.
Figure 1: (a) Original bootstrap distribution (BD) and confidence intervals (CI) for average Kappa coefficients when human annotators employ the same reference translation (Same K) or a different reference translation (Diff K) in Fomicheva and Specia (2016) (“Inacc. BD”=inaccurate BD proportion; “Acc. BD”=accurate BD proportion; “M”=number of bootstrap samples; “n”=bootstrap sample size; “R=yes”: sampled with replacement; “R=no”: sampled without replacement); (b) is (a) with accurate BD proportion; (c) is (b) with conventional M; (d) is (c) with R=yes; (e) is (d) with N=1000 (N is the full sample size); (f) is (d) with M=50; (f) corresponds to correct BD with all CI errors corrected.

Furthermore, bootstrap distributions in Fomicheva and Specia (2016) are computed by random sampling without replacement, and the size of each bootstrap does not equal the original sample size N.5 Figure 1(d) shows bootstrap distributions of Figure 1(c) when the sampling without replacement error is corrected, and Figure 1(f) shows bootstrap distributions of Figure 1(d) when the sample size error is corrected.

In summary, Figure 1(f) shows all errors with respect to CI estimate in Fomicheva and Specia (2016) corrected, and subsequently CIs for a standard implementation of bootstrap, which can be contrasted to those that led to the original conclusion of strong reference bias in Figure 1(a). CIs in Figure 1(f) for SAME and DIFF now overlap revealing that experiments in Fomicheva and Specia (2016), thus far do not show any evidence of reference bias.

2.1 Measures of Central Tendency

Even if the correct implementation of bootstrap resampling, shown in Figure 1(f), had shown non-overlapping confidence intervals, it would still unfortunately not have been appropriate to draw a conclusion from this of reference bias, however, due to the fact that significant differences are not investigated for the statistic of interest, the Kappa coefficient, but only for a measure of central tendency of two Kappa coefficient distributions, the average Kappa of each Kappa distribution. One reason for avoiding a comparison based on significant differences in average Kappa, as opposed to the Kappa point estimates themselves, is that it is...
possible for the average of two distributions to be equal, or indeed have a small but non-significant difference, while the underlying distributions differing considerably in several other respects.

Figure 2 shows Kappa coefficient distributions for all pairs of judges in SAME (40 pairs), DIFF (150 pairs) and SOURCE (10 pairs), revealing all distributions to have very similar Kappa coefficient distributions, with the one exception arising for SOURCE, where two of the human annotator pairs had an unusually high agreement level.\textsuperscript{6}

A more informative comparison about levels of agreement in SAME and DIFF examines significant differences in Kappa point estimates, as opposed to comparison based on a measure of central tendency. For this reason, despite there being no significant difference in average Kappa for SAME and DIFF, we also examine the proportion of Kappa point estimates of judge pairs in SAME that are significantly different from agreement levels of judge pairs in DIFF, which will provide genuine insight into differences in levels of agreement between the two groups.

Table 2 shows proportions of all judge pairs with significant differences in Kappa point estimates (non-overlapping confidence intervals) for each combination of settings (Revelle, 2014).\textsuperscript{7}

The number of significant differences in Kappa point estimates for pairs of judges in SAME and DIFF is only 13%, or, in other words, 87% of judge pairs across SAME and DIFF have no significant difference in agreement levels. Table 2 also includes proportions of significant differences for Kappa point estimates resulting from judges belonging to a single setting (significance testing all Kappa of SAME with respect to all other Kappa belonging to SAME, for example), revealing that the proportion of significant differences within SAME (12%) to be very similar to that of SAME × DIFF (15%), and similarly for DIFF (12%), with only a single percentage point difference in both cases in proportions of significant differences. Subsequently, even after correcting the measure of central tendency error in Fomicheva and Specia (2016), evidence of reference bias can still not be concluded.

\textsuperscript{6}The difference in distributions for SOURCE is exaggerated to some degree due to the total number of annotator pairs in SOURCE being substantially lower than the other two settings (only 10 pairs).

\textsuperscript{7}Our re-analysis code is available at https://github.com/qingsongma/percentage-refBias

Table 2: Percentage of human annotator pairs in Fomicheva and Specia (2016) with significant differences in Kappa coefficients for pairs of annotators shown the same reference translation (“SAME”), different reference translations (“DIFF”) and source sentences (“SRC”) (Fomicheva and Specia (2016) data set).

<table>
<thead>
<tr>
<th></th>
<th>SOURCE</th>
<th>SAME</th>
<th>DIFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAME</td>
<td>47% (45)</td>
<td>29% (400)</td>
<td>27% (1,500)</td>
</tr>
<tr>
<td>DIFF</td>
<td>–</td>
<td>12% (780)</td>
<td>13% (6,000)</td>
</tr>
<tr>
<td>SRC</td>
<td>–</td>
<td>12% (11,175)</td>
<td>–</td>
</tr>
</tbody>
</table>

2.2 Differences in Ratings

The effect that reference bias may or may not have on actual 1–5 ratings attributed to translations, is again only reported in terms of a measure of central tendency, i.e. average ratings, in Fomicheva and Specia (2016). The average rating of each group shown a distinct reference translation is reported, showing distinct average scores for assessors employing a distinct set of reference translations. Due to the fact that each group had a distinct average rating, the conclusion is drawn that MT quality is perceived differently depending on the human translation used as gold-standard. It is however, entirely possible that, the difference in average ratings is in part or even fully caused by
the known lack of consistency across human annotators in general.

Quite a substantial leap is made therefore between the difference in average ratings and the cause of that difference. To investigate this further, we reproduce the average ratings for assessors shown a distinct reference translation, each represented by a green square along the line labeled “DIFF(G2−G5)” in Figure 3, where the overall range in average ratings is 0.76. The extremity of this range is better put into context by comparison with the average rating of human assessors shown the same reference translation, each labeled SAME in Figure 3, where the range of average ratings attributed to human assessors shown the same reference can be as large as 0.97 (G5).

Thus, it cannot be concluded from a difference in average ratings for annotators shown distinct reference translations that the cause of this difference is the reference translation.

However, comparison of ratings based only on averages, again hides detail that an analysis could otherwise benefit from. We therefore examine the distribution of individual ratings attributed to translations, and how well ratings for the same translation correspond when pairs of annotators employ the same or distinct reference translation (or indeed the source input) in Figure 4.5 The rating pattern in Figure 4 (a) of judge pairs employing a distinct reference translation compared to those in Figure 4 (b), where assessors employ the same reference translation, shows agreement at the level of individual ratings to be almost indistinguishable, showing no evidence of reference bias.

3 Alternate Reference Bias Investigation

Although we can now say that experiments in Fomicheva and Specia (2016) showed no evidence of reference bias, a further issue lies in the fact that low IAA was incurred throughout the study, and low IAA unfortunately provides no assurance with respect to the reliability of conclusions, even when corrected for analytical errors. In addition, the fact that IAA was itself the measure by which bias was investigated is also likely to exacerbate any problems with respect to reliability of conclusions. We therefore provide our own additional investigation into reference bias in monolingual evaluation of MT. Instead of investigating via IAA, we explore the degree to which unfairly high or low ratings might be assigned to translations with respect to
Figure 5: (a) Scatter plot of direct assessment (DA) scores for 100 Chinese to English translations carried out by comparison with a generic reference translation (DA Gen-Ref) or DA with the reference replaced by a human post-edit of the MT output (DA Postedit); (b) sentence-level (smoothed) BLEU scores for the same translations also plotted against DA POST-EDIT; translations and references of (a) and (b) data set of Fomicheva and Specia (2016); post-edits provided by professional translators with access to the source and MT output only. BLEU and DA scores are standardized for ease of comparison in all plots.

Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

3.1 Reference Bias Experiments

Experiments were carried out using the original 100 Chinese to English translations released by Fomicheva and Specia (2016), in addition to 70 English to Spanish MT translations (WMT-13 Quality Estimation Task 1.1)\(^9\). Professional translators, entirely blind to the purpose of the study, were employed to post-edit the MT outputs used in the POST-EDIT setting, and were shown the source input document and the MT output document only (no reference translations).\(^{10}\)

Once post-edits had been created, DA was employed in two separate runs on Amazon Mechan-
3.2 Results and Discussion

Figure 5(a) shows a scatter-plot of DA scores attributed to translations for GEN-REF compared to POST-EDIT in the Chinese to English experiment. Translations that encounter reference-dissimilarity bias are expected to appear in the lower-right quadrant of Figure 5(a), receiving an unfairly low GEN-REF score combined with a high POST-EDIT score. As can be seen from Figure 5(a) only a very small number of translations fall into this quadrant, all of which are very closely located to adjacent upper-right and lower-left quadrants. A single translation in Figure 5(a) is an outlier in this respect, receiving a high POST-EDIT score in combination with a lower than average GEN-REF score, possibly indicating reference bias. On closer inspection, however, the score combination is in fact the result of a mistake in the reference translation. Although the low GEN-REF score was the result of an error in the reference translation, a single translation having this score combination is not sufficient evidence to conclude strong reference bias. In future work we would like to investigate the frequency of erroneous reference translations in existing MT test sets, although we expect them to be few, accurate statistics would provide a better indication of the degree to which they could negatively impact the accuracy of DA evaluations.

Figure 5(a) is also void of evidence of reference-similarity bias, as only a small number of translations lie in the upper-left quadrant and are all very close to the origin and/or adjacent quadrants.

Contrasting Figure 5(a), the correspondence of GEN-REF scores to POST-EDIT scores, with Figure 5(b), the correspondence of known reference-biased BLEU scores, in contrast a large number of BLEU scores for translations do encounter reference bias, as seen by the spread of translations appearing across both the bottom-right and upper-left quadrants.
Similarly for English to Spanish, the correspondence between GEN-REF and POST-EDIT scores for translations are shown in Figure 6(a), where, again, only a small number of translations appear in the bottom-right and upper-left quadrants, all lying very close to adjacent quadrants, again, showing no significant indication of reference bias. A single translation appears to break the trend again, however, receiving a low GEN-REF score combined with a high POST-EDIT score, located in the lower-right quadrant of Figure 6(a). On closer inspection, the low GEN-REF score is the result of something unexpected, as the MT output is in fact an accurate translation while at the same time the generic reference is also correct, but unusually the meaning of the two diverge from each other.\(^{12}\) Again, a single translation receiving this score combination is not sufficient evidence to conclude reference bias to be a significant problem for monolingual evaluation. The lack of reference bias in Figure 6(a) can again be contrasted to known reference-biased BLEU scores in Figure 6(b) for English to Spanish.

### 4 Conclusions

In this paper, we provided an investigation into reference bias in monolingual evaluation of MT. Our review of past investigations reveals potential analytical errors and raises questions about the reliability of past conclusions of strong reference bias. This motivates our further investigation for Chinese to English and English to Spanish MT employing direct human assessment in a monolingual MT evaluation setting. Results showed no significant evidence of reference bias, contrary to prior reports and intuition.

### Acknowledgments

This project has received funding from NSFC Grant No. 61379086 and the European Union Horizon 2020 research and innovation programme under grant agreement 645452 (QT21) and Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

---

\(^{12}\)Source: A straightforward man; MT: Un hombre sen-cible; Reference: Un hombre sincero

### References


Translation Quality and Productivity: A Study on Rich Morphology Languages

Lucia Specia  
Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK  
l.specia@sheffield.ac.uk

Kim Harris  
text & form GmbH, 10179 Berlin, Germany  
kim.harris@textform.com

Frédéric Blain  
Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK  
f.blain@sheffield.ac.uk

Aljoscha Burchardt  
Department of Computer Science, University of Sheffield, Sheffield, S1 4DP, UK  
ajoscha.burchardt@dfki.de

Vivien Macketanz  
Language Technology Lab, DFKI, Alt-Moabit 91c, 10559 Berlin, Germany  
vivien.macketanz@dfki.de

Inguna Skadina  
Tilde, Vienibas gatve 75a, Riga, LV 1004, Latvia  
inguna.skadina@tilde.lv

Matteo Negri  
Trento, 38100, Italy  
negri@fbk.eu

Marco Turchi  
Trento, 38100, Italy  
turchi@fbk.eu

Abstract

This paper introduces a unique large-scale machine translation dataset with various levels of human annotation combined with automatically recorded productivity features such as time and keystroke logging and manual scoring during the annotation process. The data was collected as part of the EU-funded QT21 project and comprises 20,000–45,000 sentences of industry-generated content with translation into English and three morphologically rich languages: English–German/Latvian/Czech and German–English, in either the information technology or life sciences domain. Altogether, the data consists of 176,476 tuples including a source sentence, the respective machine translation by a statistical system (additionally, by a neural system for two language pairs), a post-edited version of such translation by a native-speaking professional translator, an independently created reference translation, and information on post-editing: time, keystrokes, Likert scores, and annotator identifier. A subset of 2,000 sentences from this data per language pair and system type was also manually annotated with translation errors for deeper linguistic analysis. We describe the data collection process, provide a brief analysis of the resulting annotations and discuss the use of the data in quality estimation and automatic post-editing tasks.

1 Introduction

Data-driven approaches to machine translation (MT) rely largely on datasets of source sentences and their corresponding translations previously created by humans, so-called parallel corpora. MT systems, be they statistical or neural, are built in static fashion and (if at all) updated from time to time as more translations become available. With the popularisation of post-editing
(PE), a natural question is whether the corrected version of the MT output could be used in feedback loops to improve the current system via model retraining, model tuning or the addition of explicit model components. Additionally, by studying PE data, one can get insights on the errors made by the MT system to try and remedy them in different ways. PE data can also be used to build and benchmark metrics for the automatic evaluation of MT output, as well as quality estimation metrics and automatic PE systems.

To facilitate research in these and related areas, we have created a unique large-scale dataset with various levels of human annotation combined with automatically recorded productivity features. The data comprises 20,000–45,000 sentences of industry-generated content for English from or into three morphologically rich languages and was collected as part of the EU-funded QT21 project. The PE of all four language pairs was performed using a tool to record detailed process and product information at the sentence level during PE, including time, keystrokes, actual edits and Likert scores for the PE effort as given by the translator immediately after completion of the editing.

Most of the data was translated by a phrase-based statistical MT (PBMT) system. In addition, subsets of 15,000–20,000 sentences for EN–DE and EN–LV – respectively – were also translated using a neural MT (NMT) engine that was trained on exactly the same data used to train the original PBMT system. The PE of identical input data for both the PBMT and NMT systems facilitates large-scale direct comparisons between the actual output of these systems, as well as between process cues. For example, PE productivity can be calculated and compared using the time and keystroke information recorded during PE. The “preference” of translators can be compared through the scores given to the perceived quality of the output by such translators. A number of other comparative analyses and benchmarking in both research and industry scenarios become possible with this data.

Finally, a subset of 2,000 sentences was selected for each language pair and MT system type and manually annotated with word-level errors for deeper linguistic analysis. Both PE and error annotations were performed by professional translators.

While other datasets with PE data have been created in the past and also released for research purposes, these are limited in either their scale (e.g. see those used for the WMT13–14 shared tasks on quality estimation), have been post-edited by non-professional translators (Wosniewski et al., 2013; Bojar et al., 2015), or make only the actual post-edits available, providing no additional information on the process and no explicit annotations. The most notable example of the latter is the Autodesk dataset (Zhechev, 2012). It contains sentences predominantly belonging to Autodesk software user manuals, covering 13 language pairs with English as the source language. The source sentence, its machine translation and its post-edit are provided. The translated sentences are produced by an MT system or are translation memory suggestions with a fuzzy match score larger than 75%.

In the remainder of this paper we first describe our data sources (Section 2) and the MT systems built (Section 3) to translate this data. We introduce the PE process and its results in Section 4, and the error annotation in Section 5. In Section 6 we present two uses of the dataset.

2 Data

The post-edited and annotated data described in this paper belongs to two specific domains: information technology (IT) and life sciences. These domains were chosen because of the high demand for this type of content in multiple languages due to its economic impact on businesses active on global markets where language is key. The use of this data in research can therefore play a significant role in building the necessary bridges between the constituencies most interested in achieving progress in the field of MT: research and industry.
Table 1: Domain-specific datasets: number of sentences and source and target tokens.

<table>
<thead>
<tr>
<th>Language pair</th>
<th># sentences</th>
<th># source tokens</th>
<th># target tokens</th>
<th>Domain</th>
<th>Data provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN–DE</td>
<td>80,874</td>
<td>1,322,775</td>
<td>1,312,975</td>
<td>IT</td>
<td>Adobe</td>
</tr>
<tr>
<td>EN–CS</td>
<td>81,352</td>
<td>1,332,654</td>
<td>1,175,463</td>
<td>IT</td>
<td>Adobe</td>
</tr>
<tr>
<td>EN–LV</td>
<td>231,028</td>
<td>3,713,803</td>
<td>3,168,740</td>
<td>Pharma</td>
<td>EMEA</td>
</tr>
<tr>
<td>DE–EN</td>
<td>193,637</td>
<td>3,120,482</td>
<td>3,228,761</td>
<td>Pharma</td>
<td>EMEA</td>
</tr>
</tbody>
</table>

Table 2: Statistics on the in-domain training data. The number of words is reported in millions.

<table>
<thead>
<tr>
<th>Language pair</th>
<th># sentences</th>
<th># source words</th>
<th># target words</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN–DE</td>
<td>21,873</td>
<td>0.53</td>
<td></td>
</tr>
<tr>
<td>EN–CS</td>
<td>32,352</td>
<td>0.59</td>
<td></td>
</tr>
<tr>
<td>EN–LV</td>
<td>204,528</td>
<td>3.19</td>
<td></td>
</tr>
<tr>
<td>DE–EN</td>
<td>135,884</td>
<td>2.41</td>
<td></td>
</tr>
</tbody>
</table>

Four sets of parallel data in four language combinations (English–German/Latvian/Czech and German–English) were selected from the web. English Adobe software manuals translated into German and Czech were chosen for the IT domain, and a subset of the European Medicines Agency (EMEA) corpus was selected for the life sciences domain (which we also refer to as “pharma”) to cover the English–Latvian and German–English language pairs.2

To create datasets that can satisfy different research needs and thus increase their usability, a set of criteria was applied to data selection and pre-processing. For English–German/Czech and German–English, sentences that did not end with a punctuation mark or contained less than three or more than 35 words were discarded, and duplicate sentences were removed. These strategies reduced the number of sentence pairs by approx. 45%. For English–Latvian, a part of parallel sentences were obtained by extracting textual sentences from PDF files in the EMEA repository. First, we used Adobe Acrobat v10 Professional to convert PDF files to HTML format, as this preserved most of the original document structure. Then we ran customised scripts to convert the HTML files to plain text and clean the data. The Microsoft Bilingual Sentence Aligner (Moore, 2002) was used for sentence alignment of the parallel plain text files. Duplicate sentence pairs and sentences with less than three or more than 35 words were removed. This sentence size filtering only marginally affected the size of the final corpus. The statistics of the final sets are reported in Table 1.

For each language pair, we selected a subset of data for annotation (see Table 5), and used the remaining sentence pairs as in-domain training data to build the MT systems (Section 3). This remaining data was split into training (see Table 2), development (2,000) and test (2,000) sets.

3 MT Engine Building

3.1 Training Data
A crucial aspect for creating a set of reliable post-edited sentences and error annotations is the availability of domain-adapted translations. This is necessary because a generic translation system is not able to correctly translate domain-specific terms or expressions, which would, in turn, cause translators to rewrite translations from scratch, rendering accurate error annotation

2The German–English dataset was created by taking the available English–German data and then inverting the language direction. This is not ideal; however, very little domain-specific data exists for under-resourced language pairs, including those whose source language is German.
impossible.

When building a domain-adapted MT system we rely on different external resources depending on the size of the in-domain data. For the language pairs for which there are less than 100,000 in-domain sentence pairs (i.e. EN–DE and EN–CS), a large collection of in- and out-of-domain monolingual and parallel corpora was gathered from the web, while for the remaining languages (EN–LV and DE–EN) only in-domain corpora were used. This process resulted in:

- **EN–DE**: Over 20 million generic and in-domain sentence pairs obtained by merging the datasets available in the OPUS (Tiedemann, 2012), TAUS, WMT and JRC repositories (e.g. Europarl, CDEP, CommonCrawl, etc.);
- **EN–CS**: Over 51 million generic and in-domain sentence pairs available in the CzEng 1.6 dataset (Bojar et al., 2016b). In addition, translating into a language with free word order suggests the use of a large collection (more than 50M sentences) of monolingual generic data obtained from the Translation task at WMT16;
- **EN–LV**: Over 385,000 parallel medical sentences from the EMEA corpus available in OPUS and the most recent documents from the EMEA website (years 2009-2014);
- **DE–EN**: Over 2 million in-domain sentence pairs collected from OPUS and the data released for the medical translation task at WMT14 (Bojar et al., 2014). These resources include MuchMore, ParTr, and the Wikipedia parallel titles. In addition to these parallel sentences, monolingual data (approx. 2 million) obtained from the medical translation task at WMT14.

A summary of the external resources used to train the MT system is shown in Table 3.

### Table 3: External resources collected to train the MT systems. The reported numbers represent millions of sentences.

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>In-domain</th>
<th>Out-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN–DE</td>
<td>7.2</td>
<td>12.7</td>
</tr>
<tr>
<td>EN–CS</td>
<td>-</td>
<td>50.34</td>
</tr>
<tr>
<td>EN–LV</td>
<td>0.181</td>
<td>-</td>
</tr>
<tr>
<td>DE–EN</td>
<td>2.09</td>
<td>2.35</td>
</tr>
</tbody>
</table>

A summary of the external resources used to train the MT system is shown in Table 3.

### 3.1.1 Data Selection

In MT literature, it has been shown that when large generic datasets and a small in-domain corpus exist, the use of data selection techniques can help improve translation quality (Eetemadi et al., 2015). To optimally leverage a domain-specific corpus, we used cross-entropy-based selection for monolingual data (Moore and Lewis, 2010), its extended version for bilingual texts proposed by Axelrod et al. (2011) and the latent-domain translation method (Cuong and Simaan, 2014).

**Entropy-based method**: Originally proposed by Gao and Zhang (2002), entropy-based approaches consist in computing the perplexity score of each sentence of a generic corpus against both an in-domain language model (LM) and an LM trained on the generic corpus. The sentences are then ranked according to the difference between their two perplexity scores. Once all of the generic sentences have been ranked, the size of the subset to extract is determined by minimising the perplexity of a development set against an LM trained on an increasing amount of the sorted corpus (e.g. 5%, 10%, ...). According to (Moore and Lewis, 2010), perplexity

4http://ufal.mff.cuni.cz/czeng/ceeng16pre
decreases when less but more relevant data is used. We used the freely available open-source tool XenC (Rousseau, 2013).

**Latent-domain translation method:** This technique is able to give priors to different domains that comprise the generic data set. The goal is to estimate the probability of whether a sentence pair belongs to the in- or out-of-domain data, using in-domain corpus statistics as prior. The Expectation-Maximisation training algorithm is derived and used to estimate the out-of-domain models (given only in and mixed-domain data). This technique provides the selected data directly without the need to choose a cut-off point in the ranked list of sentence pairs. Both methods were first tested on the EN–DE language pair, and the best performing method was applied to EN–CS. In our experiments, we used the data shown in Table 1 as in-domain and the concatenation of the data in Table 3 as out-of-domain data. Although an in-domain corpus exists for EN–DE in the additional resources, it represents a mix of datasets resulting from a different distribution compared to the training data in Table 1. For this reason, all corpora in the additional resources are considered out-of-domain data.

The perplexity computed on the target side of the development set using all available data is 207. When applying both data selection methods, it significantly decreased to 150, indicating that selecting data in this fashion can be advantageous. The entropy-based method achieved a perplexity of 150, and selecting only the top 15% of the ranked sentences resulted in 3.3 million sentence pairs. The latent domain method obtained a similar perplexity (157) but selected a larger number of sentences. For this reason, the entropy-based technique is also used for EN–CS. In this case, the perplexity is higher than for EN–DE (1900), but using the top 5% of the ranked data (2.5 million sentences) allowed us to significantly reduce it to 1300. These high perplexity values stem from the fact that the external resources for EN–CS do not contain any IT data.

3.2 MT Systems

Different systems were built for each language pair using the selected and the in-domain data for EN–DE and EN–CS and the in-domain data for the other language pairs.

- **EN–DE:** Two different MT systems were created: a PBMT and a NMT system. The PBMT system was trained on all of the selected parallel training data. The phrase table was adapted to the in-domain data using the approach proposed in (Niehues and Waibel, 2012). To deal with complex reordering in the German language, this system uses a pre-reordering technique (Herrmann et al., 2013) in combination with lexical reordering. In addition, it takes advantage of two word-based n-gram language models and three additional non-word language models, namely, two automatic word class-based (Och, 1999) language models using 100 and 1,000 word classes, and a POS-based language model using fine-grained POS tags (Schmid and Laws, 2008). For the NMT system, we trained the Nematus toolkit (Sennrich et al., 2017) which is an implementation of the attentional encoder-decoder architecture (Bahdanau et al., 2014). To handle large vocabulary, the training data was previously segmented using the byte-pair encoding compression algorithm (Sennrich et al., 2016), resulting in a vocabulary of 40,000 sub-word units for both languages. We used mini-batches of 100, word embeddings of 500 dimensions, and gated recurrent unit layers of 1,024 units. The maximum sentence length was set to 50. The models were trained using Adam and by reshuffling the training set at each epoch. The NMT system was trained on the selected data and then fine-tuned on the in-domain data.

- **EN–CS:** The PBMT system was trained using Moses (Koehn et al., 2007) combined with TectoMT (Zabokrtský et al., 2008). This was done by adding the source development and test sentences and their translations obtained by TectoMT as additional (synthetic) parallel
data to the Moses system previously trained on the selected data. This new corpus and
the in-domain data were used to train separated phrase tables. At test time, we ran Moses
using all of the phrase tables and we corrected its output using Depfix (Rosa et al., 2012).
In addition, we trained a 7-gram LM on surface forms from all monolingual resources.
Similar to the EN–DE system, two additional LMs over morphological tags were built to
help maintain morphological coherence in the translation output. The system is described
in (Tamchyna et al., 2016).

- EN–LV: The PBMT system was trained on Tilde’s MT platform (Vasilevs et al., 2012).
The system is based on the Moses toolkit using the standard components. Nematus with
sub-word units was used to train the NMT system with a vocabulary size of 40,000 sub-
words. The models were trained with a projection (embedding) layer of 500 dimensions,
recurrent units of 1024 dimensions, a batch size of 20 and dropout enabled. All other
parameters were set to their default values.

- DE–EN: The PBMT system was trained using the same components and adaptation tech-
niques as those used for the EN–DE model.

The results of the different systems for each of the language pairs are reported in Table
4 according to BLEU (Papineni et al., 2002). The parameters of the models were optimised
on the development set and the final results computed on the test set. When comparing the
PBMT and NMT performance, we noticed that when using a large collection of training data
(i.e. EN–DE) the NMT system can significantly outperform the PBMT as shown in several
evaluation campaigns. However, when the training data is limited (i.e. EN–LV), the PBMT
performs better than the NMT. The language pairs with the lowest out-of-vocabulary rate (EN–
LV: 0.2 and DE–EN: 0.5) achieve the best BLEU score values. The DE–EN system obtains
better performance compared to EN–LV because it can leverage more in-domain training data.

4 Post-Editing Process

Post-editing was performed using the PET tool (Aziz et al., 2012). This is a simple and freely
available, open-source tool that tracks PE using a number of indicators. Figure 1 shows a
screenshot of the tool with an English–German PE task. The tool tracks the process of PE,
records PE time per sentence, and logs all keystrokes pressed by the annotator. This allows us
to reproduce the PE activity, which can be useful for research on topics such as PE process,
productivity gains, and automatic PE. The following information was recorded during PE:

- Editing time: time spent translating or editing a unit.
- Keystrokes: number of keys pressed during the PE according to type of keys (deletion,
  alpha-numeric, etc.).
- HTER: edit distance between the draft translation and its post-edited version.
- Evaluation: quality assessment based on a pre-defined set. We ask a question about the
  usefulness of the draft translation for PE (top left corner in Figure 1).

<table>
<thead>
<tr>
<th>EN–DE</th>
<th>EN–CS</th>
<th>EN–LV</th>
<th>DE–EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBMT</td>
<td>NMT</td>
<td>PBMT</td>
<td>NMT</td>
</tr>
<tr>
<td>35.9</td>
<td>45.8</td>
<td>38.7</td>
<td>46.5</td>
</tr>
</tbody>
</table>

Table 4: BLEU score of the PBMT and NMT systems on different language pairs.
Time, one of the most important indicators collected by the tool, is computed from the moment the target box of the unit is clicked to the moment the task is completed (either the job is closed or the navigation button "next" is pressed). The tool allows for multiple revisions, where the annotator can go back to the same sentence and edit it again. For the statistics reported here, we take the aggregation of PE time and keystrokes, and compute the edit distance between the last version and draft MT output. The outcome of a job is also stored in an XML file.

A set of PE and annotation guidelines created by the QTLaunchpad project were adapted for the PE of our data. To ensure that the quality of the post-edits was consistent and reflected the requirements of the research to be performed on the resulting data, agreement was reached on the level of editing to be done on the data. Based on the previous experience of the language partners involved, the following general rules were defined:

- Use as much of the raw MT output as possible.
- Aim for grammatically and syntactically correct translations.
- Ensure that no information has been accidentally added or omitted.
- Edit any offensive, inappropriate or culturally unacceptable content.
- Ensure proper and appropriate spelling.
- Do not restructure or change word order solely to improve the flow of the text unless dictated by grammar or domain standards.

Additionally, the following domain-specific rules for software localisation were used:

- Ensure that domain-specific terminology is correctly translated.
- Ensure that standard domain and language-specific style issues are followed.
- If formatting is used, ensure that it is correct.

For details: qt21-wiki.dfki.de/index.php?title=Post-editing_guidelines
### Table 5: General statistics of the post-edited data: Total number of sentences, average number of words in source, translation and post-edited sentences.

<table>
<thead>
<tr>
<th>Lang.</th>
<th># sentences</th>
<th># words SRC</th>
<th># words MT</th>
<th># words PE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE–EN</td>
<td>45,000</td>
<td>14.66</td>
<td>15.54</td>
<td>15.58</td>
</tr>
<tr>
<td>EN–LV</td>
<td>20,738</td>
<td>20,738</td>
<td>15.91</td>
<td>13.50</td>
</tr>
<tr>
<td>EN–CS</td>
<td>45,000</td>
<td>–</td>
<td>15.04</td>
<td>13.16</td>
</tr>
</tbody>
</table>

### Table 6: PE utility scores: the lower the score, the more useful the MT output.

<table>
<thead>
<tr>
<th>Lang.</th>
<th>PBMT</th>
<th>NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE–EN</td>
<td>1.62</td>
<td>–</td>
</tr>
<tr>
<td>EN–DE</td>
<td>1.98</td>
<td>1.40</td>
</tr>
<tr>
<td>EN–LV</td>
<td>1.64</td>
<td>1.84</td>
</tr>
<tr>
<td>EN–CS</td>
<td>2.17</td>
<td>–</td>
</tr>
</tbody>
</table>

### Table 7: Average edit distance between PE and original MT (HTER), and between PE and independent reference. The higher the distance, the more edits performed.

<table>
<thead>
<tr>
<th>Lang.</th>
<th>PBMT</th>
<th>NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE–EN</td>
<td>0.17</td>
<td>–</td>
</tr>
<tr>
<td>EN–DE</td>
<td>0.25</td>
<td>0.23</td>
</tr>
<tr>
<td>EN–LV</td>
<td>0.23</td>
<td>–</td>
</tr>
<tr>
<td>EN–CS</td>
<td>0.32</td>
<td>–</td>
</tr>
</tbody>
</table>

The guidelines were made available to all language teams and pre-editing meetings were held to avoid communication issues. Consistency in the application of these rules was critical, which is why professional translators were employed and thorough consultations were performed prior to PE.

Professional translators performed PE on every language pair. Six translators were involved in the PE for EN–DE, 4 for DE–EN, 8 for EN–LV, and 5 for EN–CS. For the evaluation score, the following options were given to the translator after the post-editing of each sentence:

1. Perfect or near perfect (typographical errors only).
2. Very good, could be post-edited quickly.
3. Poor, required significant post-editing.
4. Very poor, required retranslation.

Tables 5–8 summarise the outcome of the PE process. Much more detailed information is available in the XML output files. Table 5 provides general statistics on numbers of sentences and words per language pair and MT system type. The average perceived PE effort scores are given in Table 6. Table 7 measures the edit distance between MT and PE, and between PE and the original reference (REF). Finally, Table 8 shows average PE time and keystrokes. As expected, PE time varies considerably for different sentences, even if outliers are removed. Therefore, Table 8 also shows standard deviations.

### 5 Error Annotation Process using MQM

Our error annotation process follows a 2-step workflow. After PE, the quality of each sentence is evaluated on a scale from 1–4 as explained in the previous section. A subset of sentences scored as 2 (very good) are then selected for the error annotation phase, during which all issues resolved during the PE phase are classified. The errors are annotated using the Multidimensional Quality Metrics (MQM) error annotation framework (Lommel et al., 2014), which is popular in industry and research, and actively supported by XTM, Trados Studio, and other commercial tools. We
D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

<table>
<thead>
<tr>
<th>Lang.</th>
<th>PBMT Avg. PE time</th>
<th>PBMT Avg. PE time w/o outliers</th>
<th>PBMT Avg. keystrokes</th>
<th>NMT Avg. PE time</th>
<th>NMT Avg. PE time w/o outliers</th>
<th>NMT Avg. keystrokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE–EN</td>
<td>42±80</td>
<td>36±45</td>
<td>24.71</td>
<td>46±602</td>
<td>46±39</td>
<td>15.55</td>
</tr>
<tr>
<td>EN–DE</td>
<td>51±78</td>
<td>46±39</td>
<td>18.91</td>
<td>46±602</td>
<td>46±39</td>
<td>13.89</td>
</tr>
<tr>
<td>EN–LV</td>
<td>27±77</td>
<td>23±28</td>
<td>26.08</td>
<td>43±406</td>
<td>36±39</td>
<td>15.55</td>
</tr>
<tr>
<td>EN–CS</td>
<td>44±43</td>
<td>42±35</td>
<td>32±36</td>
<td>43±406</td>
<td>36±39</td>
<td>15.55</td>
</tr>
</tbody>
</table>

Table 8: Post-editing time and keystrokes: average number of seconds per word, with and without outliers (plus standard deviation) and average number of keys pressed during post-editing of a sentence. Outliers are sentences that took more than four minutes to be edited.

used the open-source tool translate5 (see Figure 2), a database-driven tool with a GUI. Source texts, translations, post-edits, and error annotations are organised in a relational database. The tool, originally implemented as a proofreading and PE environment for the translation industry, has been recently extended to support MQM annotation.

Figure 2: MQM error annotation in translate5 (excerpt of screenshot).

An error represents any issue that has been corrected during the PE step in the translated sentence. In the annotation step, a relevant error classification must be provided for all corrections made during PE according to a given list of errors. Error annotation is performed by experienced professional translators supported by detailed annotation guidelines.

The list of errors is divided into the main issue categories accuracy, fluency and terminology, which fold into a selection of more detailed categories from the MQM hierarchy. Figure 3 shows part of a decision tree that annotators used to select the most appropriate issue. The actual error categories used in the annotation are shown in Table 9.

Annotators are instructed to use the subcategories whenever possible and to resort to the more general category level only in case of doubt, for example, if the German term Zoomfaktor is incorrectly translated as zoom shot factor, and the annotator is unsure whether this represents a mistranslation or an addition. In this case, the error can be classified as an Accuracy error since it is unclear whether content has been added or a term mistranslated.

The annotation process has been completed for all languages and MT system types, resulting in 1,800 unique sentences per language pair and MT system type, with an additional 200 sentences doubly annotated for agreement analysis. The breakdown of error annotations for all 2,000 sentences per language pair and MT system type is shown in Table 9.

Table 10 shows an initial analysis on the agreement between pairs of annotators. Agreement was computed using Cohen’s kappa (Cohen, 1960) at the word level in two ways: firstly, for each word we count an agreement whenever both annotators agree that it is incorrect (or correct), with agreement by chance = 1/2; second, for each word we count an agreement whenever

7http://translate5.net
both annotators agree on the exact error type assigned to the word (or agree on the word being correct), considering all the 20 categories shown in Table 9 as equally likely (i.e. no distinction was made among different levels in the hierarchy), with agreement by chance = 1/21.

The interpretation of the kappa coefficient is difficult, but it is generally believed that 0.4–0.6 is moderate, while 0.6–0.8 represents substantial agreement, with anything above 0.8 indicating perfect agreement (Landis and Koch, 1977). Considering the subjectivity of the task and the number of error categories and different levels in the hierarchy, we consider the moderate to high agreement found a very positive result towards validating the annotation of the data. In the near future, further quantitative and qualitative analysis will be performed to understand problematic categories and the reasons behind certain disagreements.

6 Examples of Uses of the Dataset

Subsets of the datasets collected have been used in the 2016 and 2017 editions of the WMT shared tasks on Quality Estimation and Automatic Post-editing (Bojar et al., 2016a, 2017).\(^5\) In what follows we summarise some of the outcomes from these tasks.

6.1 Quality Estimation

Quality Estimation (QE) is the task of predicting the quality of the output of an MT system without the use of reference translations (Blatz et al., 2004; Specia et al., 2009). This is approached as a machine learning task, where training data with quality labels is needed. These labels can target different granularity levels: words, phrases, sentences or entire documents.

Early work in the area relied on proxies to quality labels generated using automatic evaluation metrics such as BLEU (Papineni et al., 2002) based on human translations. The task was thus framed as that of predicting an automatic evaluation metric score. This did not prove very successful because of the limitations of the automatic metrics themselves and the lack of a clear interpretation for the predictions (i.e. what does a BLEU score of 0.5 mean?).

Quality labels given by humans have been suggested in (Quirk, 2004) but only started to be

Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

Table 9: MQM error categories and breakdown of annotations completed to data.

<table>
<thead>
<tr>
<th>Error type</th>
<th>DE–EN</th>
<th>EN–DE</th>
<th>EN–LV</th>
<th>EN–CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>3</td>
<td>0</td>
<td>39</td>
<td>50</td>
</tr>
<tr>
<td>Addition</td>
<td>539</td>
<td>332</td>
<td>167</td>
<td>277</td>
</tr>
<tr>
<td>Mistranslation</td>
<td>437</td>
<td>967</td>
<td>852</td>
<td>274</td>
</tr>
<tr>
<td>Omission</td>
<td>576</td>
<td>690</td>
<td>355</td>
<td>395</td>
</tr>
<tr>
<td>Untranslated</td>
<td>278</td>
<td>102</td>
<td>24</td>
<td>79</td>
</tr>
<tr>
<td>Fluency</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>233</td>
</tr>
<tr>
<td>Syntax</td>
<td>302</td>
<td>525</td>
<td>245</td>
<td>49</td>
</tr>
<tr>
<td>Incorrect</td>
<td>139</td>
<td>804</td>
<td>449</td>
<td>56</td>
</tr>
<tr>
<td>Grammatic</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Function words</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Extrinsic</td>
<td>362</td>
<td>779</td>
<td>231</td>
<td>66</td>
</tr>
<tr>
<td>Word form</td>
<td>0</td>
<td>94</td>
<td>267</td>
<td>280</td>
</tr>
<tr>
<td>Part of speech</td>
<td>20</td>
<td>128</td>
<td>132</td>
<td>38</td>
</tr>
<tr>
<td>Agreement</td>
<td>18</td>
<td>506</td>
<td>97</td>
<td>419</td>
</tr>
<tr>
<td>Tense/aspect/mood</td>
<td>63</td>
<td>184</td>
<td>51</td>
<td>60</td>
</tr>
<tr>
<td>Word order</td>
<td>218</td>
<td>868</td>
<td>309</td>
<td>336</td>
</tr>
<tr>
<td>Spelling</td>
<td>118</td>
<td>126</td>
<td>132</td>
<td>324</td>
</tr>
<tr>
<td>Typography</td>
<td>282</td>
<td>553</td>
<td>249</td>
<td>823</td>
</tr>
<tr>
<td>Terminology</td>
<td>27</td>
<td>82</td>
<td>139</td>
<td>34</td>
</tr>
<tr>
<td>All categories</td>
<td>3386</td>
<td>6775</td>
<td>3700</td>
<td>3803</td>
</tr>
</tbody>
</table>

Table 10: Number of annotated words per language pair for each annotator (A1 and A2) and the Cohen’s kappa measuring inter-annotator agreement for MQM error annotations.

<table>
<thead>
<tr>
<th></th>
<th>DE–EN</th>
<th>EN–DE</th>
<th>EN–LV</th>
<th>EN–CS</th>
</tr>
</thead>
<tbody>
<tr>
<td># annotated words</td>
<td>516/643</td>
<td>974/920</td>
<td>338/288</td>
<td>669/682</td>
</tr>
<tr>
<td>Kappa on annotated words</td>
<td>0.61</td>
<td>0.70</td>
<td>0.82</td>
<td>0.69</td>
</tr>
<tr>
<td>Kappa on error type</td>
<td>0.51</td>
<td>0.48</td>
<td>0.69</td>
<td>0.53</td>
</tr>
</tbody>
</table>

used more recently (Specia et al., 2009). In particular, the use of objective labels derived from extrinsic uses of MT output, such as PE, have become popular (Specia, 2011). Labels of this type include normalised PE distance (HTER - Human Targeted Translation Error Rate (Snover et al., 2006)). These can be acquired as a by-product of PE in a translation workflow, are less subjective and less subject to biases such as the annotators’ perception of MT.

The datasets described in this paper open many new avenues for research in QE. The main benefits with respect to previously collected labels include its scale, domain specificity and the availability of multiple types of (reliable) human annotation.

In the WMT16 QE shared task, a subset of the English-German IT domain post-edited data containing 15,000 sentences was used for the sentence, word and phrase-level tasks. The quality labels were automatically derived from the PE of the MT output, e.g. for sentence level, HTER scores were used. Bojar et al. (2016a) claim that, when compared to previous year – approx. 14,000 crowdsourced post-edited sentences – the results of the 2016 task were more conclusive. They attribute this to the higher quality of the new dataset and observe that:
Table 11: QE shared task results on the 2016 test set: baseline and winning systems in 2016 and 2017 (larger training set) for sentence (Pearson), word and phrase ($F_1$-mult = multiplication of $F_1$ for the GOOD and BAD classes) levels.

- for sentence level, the best Pearson correlation between the system prediction and true HTER in 2015 was 0.39 (against 0.14 of the baseline system). In 2016, the winning submission reached 0.52 Pearson correlation (against 0.35 of the same baseline system). One can speculate that the task was made somewhat “easier” by using high quality data, but the delta in the Pearson correlation between the baseline and winning submission is still substantial.
- for word level, 2016 systems performed much better: 0.56 against 0.43 $F_1$-BAD. The baseline systems are not comparable.

In order to further push progress in the QE field, the 2017 QE task was provided with an extended version of the 2016 dataset in addition to data from a different domain and a different language pair. For English-German, the 2016 dataset was extended to include a total of 28,000 sentence pairs. For German-English, 28,000 sentence pairs in the life sciences domain were made available for the task.

The two datasets are significantly larger than any dataset used before in QE shared tasks. The same data was used for the three subtasks: sentence, word and phrase levels. The results of this year’s task (Bojar et al., 2017) show major improvements for all tasks over the 2016 results. In addition to general advances in the field, these can in part be attributed to the larger dataset provided. For the 2016 test set, also used in 2017 for comparison, Table 11 shows the results using the official metrics for the best system and the baseline system using the 2016 vs the 2017 training sets.

This data has proven useful for subsequent work in the field: for instance, (Forcada et al., 2017) focuses on the prediction of PE time at sentence level on the 2016 dataset, while (Martins et al., 2017) proposes a novel word-level QE approach using automatic PE techniques.

6.2 Automatic Post-Editing

Automatic Post-editing (APE) systems are usually trained on (source, MT, human post-edit) triplets from which the appropriate corrections of systematic errors should be learned and possibly generalised. This supervised learning problem is addressed as a “monolingual translation” task in which rough MT output in a given target language has to be translated into a fluent and adequate translation of the original source text. BLEU and TER computed against reference human post-edits are the standard evaluation metrics for the task, and their respective improvements and reductions are usually compared against the baseline scores obtained by the original MT output that has been left untouched (i.e. rough, non post-edited translations).

Early APE systems (Allen and Hogan, 2000; Simard et al., 2007) were developed under the PBMT paradigm, that is, by learning from “parallel” data, either (MT, human post-edit)
pairs or triplets including information from the source text (Béchara et al., 2011; Chatterjee et al., 2015). Recent solutions achieved larger and more significant improvements by exploiting neural methods (Junczys-Dowmunt and Grundkiewicz, 2016; Pal et al., 2016, 2017), which approach the task as a sequence to sequence learning problem.

Both paradigms suffer from drawbacks that have, to date, represented the main obstacles towards a wider adoption of APE technology. According to Bojar et al. (2015), one of the major problems lies in data sparsity, which limits the ability to exploit training data in order to learn correction patterns that can also be applied to test instances. Several factors contribute to raising this data sparsity issue, namely: i) the size of the data (although human post-edits are a by-product of industrial translation workflows, few corpora are available for research), ii) the domain of the data (general domains – like news – are definitely less repetitive than narrow ones – like information technology), and iii) the origin of the post-edits (professional post-editors are definitely more reliable and coherent than non-expert ones).

The datasets described in this paper aim to mitigate the problems related to data sparsity for reasons that are similar to those discussed in the previous section on QE. Indeed, their size, domain specificity and professional PE quality may explain the renewed interest and the impressive progress of APE research in the past few years. The following figures drawn from the WMT experience support our claims:

- Number of tasks and submitted runs. At WMT 2016, only one English-German translation task in the IT domain was organised, while 2017 saw two tasks: English-German (IT) and German-English (life sciences). The new corpora (more repetitive than news data edited by non-experts in 2015) motivated more teams to participate: from 7 submissions in 2016 to 20 in 2017.
- Improvements over the baseline. The switch to new data coincided with significant performance gains that prove the viability of APE in domain-specific settings. While in 2015 none of the participants was able to beat the baseline, the best English-German submissions in 2016 and 2017 improved over the baseline by up to 5.5 and 7.6 BLEU points.
- Improvements over the PBMT approach. While in 2015 all systems followed this paradigm, falling in the same range of performance, the combination of advancements in neural research and the provision of more suitable data resulted in impressive performance gains in the next two evaluation rounds. The same PBMT system used for comparison in all the evaluation rounds was significantly outperformed by most of the participants in 2016 (up to 3.2 BLEU points) and in 2017 (up to 7.1 BLEU points).

7 Conclusions

In this paper we introduced a large and unique set of data points derived from industry data that have been post-edited and annotated by professional translators. This allows for specific features and novel combinations of features to be used for a variety of research and user-oriented purposes, including establishing the actual PE effort by translators based on time and keystrokes and comparing these results to the perceived level of quality of the post-edited sentence, establishing correlations between certain characteristics such as sentence length and post-editing time, or post-editing time and human or automatic quality evaluation metrics. The datasets also measure post-editing productivity and can be used to detect error patterns in the MT output.

In addition, the creation of MQM-annotated subsets of these post-edits for typical industry domains provide information about error patterns and support feature-oriented quality estimation and evaluation, among many other novel avenues for research. This dataset is freely available and can be downloaded from the project website: http://www.qt21.eu/.
Acknowledgements

This work was supported by the QT21 project (H2020 No. 645452).

References


N  Bilexical Embeddings for Quality Estimation

Bilexical Embeddings for Quality Estimation

Frédéric Blain, Carolina Scarton and Lucia Specia
Department of Computer Science
University of Sheffield, UK
{f.blain, c.scarton, l.specia}@sheffield.ac.uk

Abstract

This paper describes the SHEF submissions for the three sub-tasks of the Quality Estimation shared task of WMT17, namely: (i) a word-level prediction system using bilexical embeddings, (ii) a phrase-level labelling approach based on the word-level predictions, (iii) a sentence-level prediction system using word embeddings and handcrafted baseline features. Results are promising for the sentence-level approach, but still very preliminary for the other two levels.

1 Introduction

Quality Estimation (QE) allows the evaluation of Machine Translation (MT) when reference translations are not available. It can be used in various ways such as in post-editing (PE) to predict whether or not an automatically generated sentence is worth publishing, editing or it should be retranslated manually. Word-level predictions can be helpful by highlighting words that cannot be relied upon or should be fixed by post-editors. More recently, QE at phrase-level has emerged as a way of using quality predictions at decoding time in phrase-based Statistical MT (SMT) systems to guide the decoder such as to keep phrases which are predicted as good, and conversely to discard those which are predicted as bad (Logacheva, 2017).

QE models are built based on a list of features along with a Machine Learning algorithm for either regression or classification. These features are usually extracted from the source and target texts or from the MT system that generated the translations. Shah et al. (2015) introduced a new set of features extracted using an unsupervised approach with the use of neural network: continuous-space language model features and word embeddings features.

In our contribution this year we investigate whether we can go beyond engineered features by learning bilexical operators over distributional representations of words in source-target text pairs. Considering the MT pipeline as a noisy black-box, our motivation is to be able to build QE models to predict if information encoded in the source sentence is preserved in the target sentence after translation.

2 Bilinear Model

Madhyastha et al. (2014) propose to use word-level embeddings to predict the strength of different types of lexical relationships between a pair of words, such as head-modifier relations between noun-adjective pairs. They designed a supervised framework for learning bilexical operators over distributional representations, based on learning bilinear forms $W$. We adapted their method to predict the strength of relationship between source and target words. This problem is formulated as a log-bilinear model, parametrized with $W$ as follows:

$$
\Pr(t|s; W) = \frac{\exp \left\{ \phi(t)^\top W \phi(s) \right\}}{\sum_{t' \in T} \exp \left\{ \phi(t')^\top W \phi(s) \right\}}
$$

where $\phi$ denotes the word embeddings of any given word in a vocabulary $V$. The source words $s$ and target words $t$ are respectively taken from subspaces $S \subseteq V$ and $T \subseteq V$.

In essence, the problem can be reduced to first obtaining the corresponding word embeddings of the vocabularies of both source and target sentences using a substantially large monolingual corpus for each of the two languages, followed by using the bilinear model to estimate $W$. $W$ is learned...
using the source-target word alignment by mini-

mizing the negative log-likelihood using a \( l_2 \) reg-

ularized objective as:

\[
L(W) = -\sum_{s,t} \log(Pr(t|s; W)) + \lambda \|W\|_2^2 \tag{2}
\]

where \( \lambda \) is the constant that controls the capacity

of \( W \) with gradient descent-based optimization.

We explore this approach for both word and

phrase-level QE. For training, we rely on both the

word-alignments and the gold QE labels (i.e. the

OK/BAD labels). The former gives us the source-

target pairs, and the latter whether this pair is

valid or not. Our assumption is that this approach

should be able to predict whether or not a word in

the target language (MT output) is correct by ex-

ploring the strength of the linguistic relation with

the source word it is generated from.

3 Experimental Settings

3.1 Data and Gold labels

Each QE shared task has two datasets: Eng-

lish—German segments on the IT do-

main (with 23,000 sentences for training,

1,000 for development and 2,000 for test), and

German—English segments on the Pharmaceu-

tical domain (with 25,000 sentences for training,

1,000 for development and 2,000 for test).

The same data is used for all three tasks: word, phrase

and sentence-level prediction.

For the word-level task, each token of the MT

is annotated with OK or BAD labels. For the

phrase-level task, phrases are segmented as given

by an SMT decoder and also annotated with OK

or BAD labels. Finally, for the sentence-level task,

the quality label is a Human-Targeted Error Rate

(HTER) score (Snover et al., 2009).

3.2 Word Embeddings

Word embeddings were used in our submissions

for the three tasks. We trained in-domain skip-

gram embeddings on the in-domain data shown in

Table 1 using FastText\(^1\) (Bojanowski et al., 2016)

with 300 dimensions and learning rate set to 0.025.

The default training settings are otherwise used.

The in-domain data is the same as that used to train

the SMT system that produced the translations in

the QE datasets, as made available by the task or-

ganizers.

For the word and phrase-level tasks, we used

our word embeddings to obtain a word vector rep-

resentation of 300 dimensions for each word of

both the training and development sets. For the

sentence-level task, the word embeddings are av-

eraged for each sentence, as previously applied

in (Scarton et al., 2016).

3.3 Tool

To learn to predict the labels for the word-level

task, we used BMAPS\(^2\), the toolkit implementing

the method in (Madhyastha et al., 2014) along with

the word alignments provided by the organizers

(as produced by the SMT system). BMAPS is used

to learn the bilexical operators between both

source and target embeddings. The tool relies on

three matrices corresponding to the source and tar-

target vocabularies of the training data, and a third

matrix representing the word-level lexical relation

between them. This matrix is built from the word-

level alignments and the gold labels to indicate

which lexical items form a pair, and whether their

lexical relation is OK or BAD (i.e. if two lexical

items are aligned and labelled as OK, their inter-

section in the third matrix is set to 1, 0 otherwise).

By default, the model is trained over 100 it-

erations with the \( l_2 \) norm as regularizer, and us-

ing the forward-backward splitting algorithm (FO-

BOS) (Duchi and Singer, 2009) as optimization

scheme (\( \text{trc} = 0.1, \text{tau} = 0.1 \)).

3.4 Evaluation

We used the official task metrics to evaluate our

results. For the word and phrase-level tasks, the

metrics are \( F_1 \)-BAD and \( F_1 \)-OK which correspond to

the \( F_1 \) scores on both BAD and OK labels, and \( F_1 \)-

multi which is the product of the two formers. For

the sentence-level task, the metrics for scoring are

Pearson’s correlation (primary metric), Mean Av-

earge Error (MAE) and Root Mean Squared Error

(RMSE), and for ranking, Spearman’s rank corre-

lation (primary metric) and DeltaAvg.

\(^1\)https://github.com/facebookresearch/

\(^2\)https://github.com/f00barin/bmaps
4 Results

4.1 Word-level QE prediction (Task 2)

We investigate different context windows to build our lexical representations, ranging from a wide window considering all sentence-level context, to a much narrower approach representing each word individually:

- **Full context**: each word is associated with its left and right context to capture the exact distributional features of the specific context in which this lexical item occurs. A lexical item is thus a 900-dimensional word vector represented by the tuple \( <\text{emb}_{\text{left}}, \text{emb}_{\text{cur}}, \text{emb}_{\text{right}} > \), where \( \text{emb}_{\text{left}} \) and \( \text{emb}_{\text{right}} \) are the averaged embeddings of the left/right contexts and \( \text{emb}_{\text{cur}} \) the word representation of the current word. Here our assumption is that a lexical item would represent a word within its context and at its position in the sentence, therefore if the word appears twice in the sentence, it would be represented by two different lexical items.

- **Surrounding context**: instead of considering all the left and right context of the current word, we limit ourselves to the two surrounding words. This allows for a model that is as generic as possible while still considering two distributional features corresponding to two different lexical items. Here the assumption is the same as before, the lexical item which represents a word is the same but only considering a window of one word on the left/right to compute \( \text{emb}_{\text{left}}/\text{emb}_{\text{right}} \).

- **Unigram**: we use only the embeddings of the current word without considering any surrounding context. By doing so, we fully rely on the embeddings and the way they are trained (skipgram). In this case, the lexical item is a single word representation of 300 dimensions.

For each context we investigate two variants: with and without the use of the gold labels in order to demonstrate the capacity of our approach to learn how to discriminate the valid lexical pairs from the others.

**Discussion** The results of our approach for the word-level task are given in Table 2. We report the results of our official submissions to the task (†) along with additional experiments we conducted after the task deadline. They are both compared with the official baseline of Task 2.

Our first observation is the overall low performance of our approach compared to the official baseline. However, we found very encouraging the results of our additional experiments compared to those of the systems submitted. The revised training procedure significantly improved the performance in terms of \( F_1 \)-OK for all three contexts types, resulting in a boost in the \( F_1 \)-multi scores.

To better understand the gap between our official and additional results, it is important to mention the technical constraints we faced performing the task with BMAPS for the official submission. In its current implementation, BMAPS relies on non-sparse matrices which in our case lead to a heavy memory print, since the source and the target matrices contain vector representations for each word in the corpus. Therefore, to be able to run BMAPS on our servers we were limited to use up to 2,000 sentences (about 9% of the training corpus) as training instances. This certainly had a significant impact on the performance of the models.

To tackle this constraint we later opted for a mini-batch training approach: we divided the training corpus into batches of 500 sentences, the training for each batch starting from the results from the training with the previous one. By doing so we are able to use all the training data. However, in BMAPS the size of the dev set (in terms of words from which the matrices are built) has to be smaller than that of the training set. Therefore, by using mini-batches we had to reduce our dev set. We selected for the dev set 250 sentences with the highest number of OK labels in order to boost performance for this class. We also refined our training parameters by switching to the nuclear norm (which is expected to converge faster when restricting the training size (Madhyastha et al., 2014)). Finally, we empirically identified the best values for the two main parameters (namely \( lc \) and \( tau \)) for different context types: for both the full and surrounding context, we used \( lc = 0.1 \) and \( tau = 0.001 \), while for the unigram approach we used \( lc = 0.1 \) and \( tau = 0.01 \).

As a second finding, one can notice the impact of considering the surrounding context when predicting each word’s label. In both official and additional results, there is a substantial difference be-
D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

Table 2: Results of our word-level predictions. † denotes our official submissions to the task using the \( l_2 \) norm and single training set of 2k sentences. The other figures are obtained with mini-batch training using 500 sentences at the time. In grey are the results of the official baseline of the task.

<table>
<thead>
<tr>
<th></th>
<th>norm training size</th>
<th>( F_1 )-BAD</th>
<th>( F_1 )-OK</th>
<th>( F_1 )-multi</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>English → German</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMAPS full</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.326</td>
<td>0.103</td>
</tr>
<tr>
<td>BMAPS full</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.311</td>
<td>0.103</td>
</tr>
<tr>
<td>BMAPS full</td>
<td>( \mu )</td>
<td>23k</td>
<td>0.321</td>
<td>0.117</td>
</tr>
<tr>
<td>BMAPS full</td>
<td>( \mu )</td>
<td>23k</td>
<td>0.321</td>
<td>0.117</td>
</tr>
<tr>
<td>BMAPS unigram</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.151</td>
<td>0.222</td>
</tr>
<tr>
<td>BMAPS unigram</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.151</td>
<td>0.222</td>
</tr>
<tr>
<td>BMAPS window</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.151</td>
<td>0.222</td>
</tr>
<tr>
<td>BMAPS window</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.151</td>
<td>0.222</td>
</tr>
<tr>
<td>BMAPS unigram</td>
<td>( \mu )</td>
<td>23k</td>
<td>0.262</td>
<td>0.321</td>
</tr>
<tr>
<td>BMAPS unigram</td>
<td>( \mu )</td>
<td>23k</td>
<td>0.262</td>
<td>0.321</td>
</tr>
<tr>
<td>BASELINE</td>
<td>–</td>
<td>–</td>
<td>0.404</td>
<td>0.404</td>
</tr>
</tbody>
</table>

**German → English**

<table>
<thead>
<tr>
<th></th>
<th>norm training size</th>
<th>( F_1 )-BAD</th>
<th>( F_1 )-OK</th>
<th>( F_1 )-multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMAPS full</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.536</td>
<td>0.412</td>
</tr>
<tr>
<td>BMAPS full</td>
<td>( l_2 )</td>
<td>2k</td>
<td>0.536</td>
<td>0.412</td>
</tr>
<tr>
<td>BMAPS unigram</td>
<td>( \mu )</td>
<td>23k</td>
<td>0.102</td>
<td>0.097</td>
</tr>
<tr>
<td>BMAPS unigram</td>
<td>( \mu )</td>
<td>23k</td>
<td>0.102</td>
<td>0.097</td>
</tr>
<tr>
<td>BASELINE</td>
<td>–</td>
<td>–</td>
<td>0.407</td>
<td>0.407</td>
</tr>
</tbody>
</table>

**Discussion**

The results of these three phrase-level labelling strategies based upon our word-level predictions are given in Table 3. We report the results of our official submissions to the task (†) along with additional experiments we conducted after the task deadline. These are compared with the official baseline for Task 3.

First, similarly to the word-level task, the performance at phrase-level improved with the additional experiments, which was expected since the labelling directly follows from the word-level predictions. Second, while we originally observed better labelling performance using the optimistic approach on test.2016 (see underlined numbers), we now observe better \( F_1 \)-* scores with both pessimistic approaches for en–de. One can also observe comparable performance for en–de when the surrounding context is used: the difference in terms of \( F_1 \)-* scores between the full and window context is marginal. For de–en this is different: the phrase labelling based on word predictions using the window context outperforms the phrase la-

---

between the three types of context: while unigram was the best performing when limited to 2k training instances only, the exact opposite was found when using the full training set with better \( F_1 \)-* scores when the context in which the word occurs is employed. Furthermore, we note a small advantage for the window context over the full context in both language pairs. We believe this means that considering the surrounding context could better help in a situation where a word would appear twice in the same sentence but should be labelled differently.

Overall, these results are encouraging and we aim to pursue further investigations towards improving this approach for the task of word-level QE.

### 4.2 Phrase-level QE labelling (Task 3)

While we could have chosen to predict phrase-level QE labels similarly to our word-level predictions, we opted for generating phrase-level labels from word-level labels following the labelling approaches described in Blain et al. (2016):

- **Optimistic**: if half or more of words have a label \( \text{OK} \), the phrase has the label \( \text{OK} \) (majority labelling).
- **Pessimistic**: if 30% words or more have a label \( \text{BAD} \), the phrase has the label \( \text{BAD} \).
- **Super-pessimistic**: if any word in the phrase has a label \( \text{BAD} \), the whole phrase has the label \( \text{BAD} \).

---

The phrase labelling based on word predictions using the window context outperforms the phrase lab-
Handcrafted features. The idea was to combine word embeddings with by Scarton et al. (2016) for document-level QE.

For the sentence-level task we followed a simplistic approach. On the other hand, we significantly underperform in the two other tasks, which exploit a very simplistic approach. On the other hand, we significantly underperform in the two other tasks, which exploit handcrafted features.

The results of the other two labelling strategies based upon our word-level QE predictions are given in Table 4. Although the sentence-level experiment is different from the approach applied for word and phrase-level tasks, our aim was to test the usability of the in-domain word embeddings. Our results are compared with the official baseline.

### Discussion

The results of our sentence-level predictions are given in Table 4. Although the approach is rather simplistic, it achieves considerably good results by outperforming the baseline system and several other systems that participated in the shared task. For German→English, our system performed seventh out of 13 in the scoring task. For English→German, it performed eight out of 13. Table 4 shows the results of our systems (called QUEST-EMB) for the different language pairs and for both scoring and ranking tasks. We also show the results of the baseline systems for comparison.

### 5 Conclusions

In this paper we report our submissions to the three sub-tasks of the QE campaign of WMT17. We obtained reasonably good results for the sentence-level task despite the use of a very simplistic approach. On the other hand, we significantly underperform in the two other tasks, which exploit word embeddings trained on general purpose data, our embeddings are trained over in-domain data, as previously described. Word embeddings were averaged at sentence level in order to have a single vector representing each sentence. We then concatenated source and target in-domain embeddings with the 17 sentence-level baseline features provided by the organisers. An SVM regressor was used to train our QE model with hyper-parameters optimized via grid-search. For that we used the learning module available at QuEst++ toolkit (Specia et al., 2015).

Although the sentence-level experiment is different from the approach applied for word and phrase-level tasks, our aim was to test the usability of the in-domain word embeddings. Our results are compared with the official baseline.

### 4.3 Sentence-level QE prediction (Task 1)

For the sentence-level task we followed a simple approach, which had been previously applied by Scarton et al. (2016) for document-level QE. The idea was to combine word embeddings with handcrafted features.

However, whilst Scarton et al. (2016) have used...
a bilinear model. Due to limitations regarding the experimental settings of the tool used for the official submissions, it is difficult to conclude whether or not our approach is suitable for the task of QE. In follow up experiments with different training strategies, the results proved substantially better and much more promising, albeit still behind the official baseline. This is particularly encouraging considering that the approach only relies on word embeddings and word alignment information. We plan to further experiment with it and identify possible improvements in BMAPS that could lead to better performance.

It is also worth emphasizing that the approach employed for the sentence-level task is not directly comparable to the approach used for the other tasks; they only share the embeddings trained using in-domain data. However, we can conclude that the in-domain embeddings encode useful information for all tasks.

Acknowledgments

The authors would like to thank Pranava S. Madhyastha for his support regarding the use of BMAPS. This work was supported by the QT21 project (H2020 No. 645452).

References


Feature-Enriched Character-Level Convolutions for Text Regression

Gustavo Henrique Paetzold and Lucia Specia
Department of Computer Science
University of Sheffield, UK
{g.h.paetzold,l.specia}@sheffield.ac.uk

Abstract
We present a new model for text regression that seamlessly combine engineered features and character-level information through deep parallel convolution stacks, multi-layer perceptrons and multi-task learning. We use these models to create the SHEF/CNN systems for the sentence-level Quality Estimation task of WMT 2017 and Emotion Intensity Analysis task of WASSA 2017. Our experiments reveal that combining character-level clues and engineered features offers noticeable performance improvements over using only one of these sources of information in isolation.

1 Introduction

Text regression consists in estimating a numeric label based on information available from the text. The label can represent any abstract property of said text: its appropriateness, sentiment, fluency, simplicity, quality, etc. Due to their wide applicability in both research and industry, some of these tasks have been gaining a lot of attention. These include Quality Estimation and Emotion Intensity Analysis, which are the subjects of shared tasks held at the WMT 2017 conference1 and WASSA 2017 workshop2 (Mohammad and Bravo-Marquez, 2017), respectively.

In Quality Estimation (QE), one attempts to estimate the quality of a machine translated text based on the information that can be extracted from the original sentence and its translation. The task has many variants, given that the quality of a translation can be estimated at word, phrase, sentence or even document level. Quality estimates can be incorporated in Machine Translation (MT) decoding or used for re-ranking of top candidates, for example, allowing for a more intelligently guided translation process (Avramidis, 2012), or they can be used to help human translators decide which automatic translations are worth post-editing, and which should be re-translated from scratch (Turchi et al., 2015). Sentence-level QE is the most popular variant, mostly due the fact that most modern statistical and neural MT systems translate one sentence at a time. In this task, the input is the original-translation sentence pair and the output is some numeric label that represents quality. The most commonly used label is HTER, which measures the human post-editing effort required to fix the translation in question (Snover et al., 2006).

As shown in (Bojar et al., 2016), the performance of QE approaches submitted to the WMT shared tasks have steadily improved in recent years. However, the nature of these approaches have not changed much: most of the top ranking systems employ well-known regression methods and extensive feature engineering. Some of the most notable examples are the RTM systems of WMT 2014 and 15, which managed to reach the top of the ranks by employing Referential Translation Machines trained with SVMs for regression (Bicici, 2016). The LORIA (Langlois, 2015) and YSDA (Kozlova et al., 2016) systems of WMT 2015 and 2016, respectively, achieved similar performance by also pairing SVMs with many resource-heavy features.

Neural Networks for sentence-level QE were introduced in WMT 2016 with the SimpleNets (Paetzold and Specia, 2016) and POSTECH (Kim and Lee, 2016) systems. While the SimpleNets system uses sequence-to-label LSTMs to predict the quality of a translation’s n-grams and then
combines them, the POSTECH system learns quality labels at word-level using a sequence-to-sequence model, and then combines them with a sequence-to-label model to predict quality at sentence-level. Though very interesting and distinct strategies, neither of them managed to outperform the best scoring SVM-based approach of WMT 2016.

In the task of Emotion Intensity Analysis (EIA), Neural Networks have not yet been successfully employed. Unlike typical Sentiment Analysis tasks, which are set up as either binary or multi-class classification problems that require one to determine the opinion or sentiment in a given text, EIA aims at quantifying a certain emotion in a text, such as fear, anger, joy, sadness, etc. In the Emotion Intensity shared task of SemEval 2016 (Krichen et al., 2016), which is the first of its kind, none of the five systems submitted employ neural regressors. We were also unable to find any other contributions outside the SemEval 2016 task that explore neural approaches to EIA.

Given the volume of opportunities available when it comes to neural solutions for text regression, we introduce a new neural approach for the task. We innovate by using deep convolutional networks and multi-task learning to combine character-level information from the texts at hand with engineered features. Using this approach, we create the SHEF/CNN systems for the sentence-level QE task of WMT 2017 and the Emotion Intensity Analysis task of WASSA 2017. In what follows, we describe our approach in detail.

2 Overview of Tasks
As previously mentioned, we address two text regression tasks in this paper: the sentence-level Quality Estimation task of WMT 2017 and Emotion Intensity Analysis task of WASSA 2017. The next Sections describe each of those tasks.

2.1 Quality Estimation at WMT 2017
In the sentence-level QE task of WMT 2017 participants were asked to create systems that predict the human post-editing effort required to correct an automatically translated sentence. Training, development and test sets were provided for two language pairs: English-German and German-English. The training and development sets for both language pairs are composed of 23,000/25,000 and 1,000/1,000 instances, respectively. Each instance is composed of a source (original) and target (translated) sentence pair, as well as the target’s manually post-edited version and an HTER label between 0 and 1 calculated based on the post-edit. The test set is composed of 2,000 instances without post-edits nor HTER labels. For training, development and test sets the organizers made available a set of 17 baseline features.

The task is divided in two sub-tasks: scoring and ranking. In the scoring task, systems had to estimate HTER scores and were evaluated through Pearson correlation. In the ranking task, systems had to rank the translations in the test set from highest to lowest quality, and were evaluated through Spearman correlation. The main difference between the data provided for the WMT 2017 QE tasks and the data of previous editions is that, for the first time, the tasks of all QE levels (sentence, word and phrase) contain annotations for the same set of translations. Because of that, one can very intuitively employ any variety of multi-task learning approaches.

2.2 Emotion Intensity at WASSA 2017
Systems submitted to the Emotion Intensity Analysis task of WASSA 2017 were asked to estimate the intensity of various emotions felt by authors while writing tweets. Training, development and test sets were made available containing four emotions: anger, fear, joy and sadness. The size of the datasets is illustrated in Table 1.

<table>
<thead>
<tr>
<th>Emotion</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anger</td>
<td>857</td>
<td>84</td>
<td>760</td>
</tr>
<tr>
<td>Fear</td>
<td>1,147</td>
<td>110</td>
<td>995</td>
</tr>
<tr>
<td>Joy</td>
<td>823</td>
<td>79</td>
<td>714</td>
</tr>
<tr>
<td>Sadness</td>
<td>786</td>
<td>74</td>
<td>673</td>
</tr>
</tbody>
</table>

Table 1: Dataset sizes for the Emotion Intensity Analysis task of WASSA 2017

Each instance is composed of a tweet and an intensity label between 0 and 1 of the emotion in question. Labels were collected through crowdsourcing. Systems were evaluated through Pearson correlation.

3 Model Architecture
Figure 1 illustrates the neural model architecture of the SHEF/CNN systems for the QE task of
WMT 2017. As it can be noticed, the model takes as input a one-hot character-level representation of the source and target, as well as a set of engineered features. As output, our model produces the numeric labels desired.

The model is divided in three main sections: a pair of deep convolution layer stacks for the source (original) and target (translated) sentences, a multi-layer perceptron for the engineered features, and a final multi-layer perceptron to combine all this information. The model used for the EIA task of WASSA 2017 is identical, except that it only has one set of convolution stacks for the tweet being analysed.

3.1 Extracting Character-Level Clues

In order to exploit the information at character-level from the text, we use a convolution architecture similar to the one introduced by (Kim et al., 2016), who successfully employ character-level information for language modelling. First we transform the one-hot character-level representation of the sentence into a sequence of character embeddings. We then feed them to a series of parallel one-dimensional convolutions of different window sizes. Each of these convolutions captures the information of character n-grams of a given length: a convolution of window size one addresses unigrams, one with size two addresses bigrams, and so on. Finally, the resulting values produced by the convolution filters are passed on to a one-dimensional max-pooling layer.

In order to capture information at different abstraction levels, we stack various convolution and max-pooling layers for each window size, thus creating a deep architecture. This deep architecture differs from the one used by (Kim et al., 2016) in the sense that they apply only one stack of convolution/max-pooling layers for each window size. The values produced by the last max-pooling layer of each window size are then flattened so that they can be easily concatenated.

The intuition behind using such an architecture lies in the assumption that sequences of characters hold important clues with respect to the text’s properties, such as quality and emotion. In QE, these clues could be sequences containing morphological errors in words from the source or target sentences, or sequences in-between tokens of the target that suggest an ungrammatical segment, for example. In EIA, these clues can be emotionally charged emojis, curse words, exclamation marks, etc.

3.2 Incorporating Engineered Features

We complement character-level information with engineered features, given that the most effective QE and EIA methods in previous work heavily exploit them (Kim and Lee, 2016; Kozlova et al., 2016; Refaee and Rieser, 2016; Wang et al., 2016). To do so, we apply a simple multi-layer perceptron (MLP) over a set of input engineered features. This allows to capture abstract relations between the features provided. The output of the outermost layer is then concatenated with the flattened character-level information provided by the remainder of the network.

Finally, we pass the concatenated features and character-level information to another MLP in order for our model to be able to capture any relations between them. At the very edge of our model, we include output nodes for as many tasks as we wish to train our model over.

4 SHEF/CNN Model for QE

As illustrated in Figure 1, the sentence-level QE model employs one convolution stack for each of the source and target sides of the translation pair. We configure the model as follows:

- **Embedding size**: We train character embeddings with 50 dimensions.
- **Window range**: We use 4 parallel stacks of convolutions with window sizes from 1 to 4.
- **Convolution depth**: Each stack contains 4 pairs of convolution/max-pooling layers with 50 convolution filters each and a pool length of 4.
- **Feature MLP depth**: We stack 2 dense layers with 50 hidden units over engineered features.
- **Final MLP depth**: The MLP that combines convolutions and features is composed of 2 stacked dense layers with 50 hidden units each.
- **Engineered feature set**: We use the 17 baseline features provided by the task organizers.

This architecture was selected through experimentation. The output nodes of our multi-task QE setup predict three values:

- HTER from the sentence-level dataset;
Figure 1: Architecture of the SHEF/CNN+BASE systems
The number of BAD labels from the word-level dataset; and

The number of BAD labels from the phrase-level dataset.

Note that the data from the word and phrase-level datasets are used as a mere complement to HTER prediction. It is important to mention that we also tried predicting the full label sequences for word and phrase-level, but the results obtained were not as promising. We train our model until convergence with Stochastic Gradient Descent and Mean Squared Error over all outputs jointly.

5 SHEF/CNN Model for EIA

The model used for the EIA task of WASSA 2017 applies only one convolution stack over the tweet being analysed, given that the task is not characterized by a sentence pair. The window range, convolution depth, as well as feature and final MLP depths are identical to the model used for the WMT 2017 task. We train one model for each emotion targeted in the shared task: anger, fear, joy and sadness.

Since the organizers did not provide a set of baseline features, we produced our own features using the Stanford Sentiment Treebank (Socher et al., 2013), which is composed of 239,232 text segments annotated with respect to their positivity probability i.e. how likely they are to convey a positive emotion. The positivity values range from 0.0 (absolutely negative) to 1.0 (absolutely positive). Using this data, we extract nine features from each tweet:

• Minimum, maximum and average positivity of single words in the tweet;

• Minimum, maximum and average positivity of bigrams in the tweet; and

• Minimum, maximum and average positivity of trigrams in the tweet.

Our multi-task learning setup is composed of two output layers that predict:

• The tweets’ emotion intensity; and

• The tweets’ positivity value.

We first train our models over the sentiment positivity values from the Stanford Sentiment Treebank until convergence, then train them over the emotion intensity training sets of WASSA 2017 until convergence. The training algorithm and metric used are Stochastic Gradient Descent and Mean Squared Error, respectively.

6 WMT 2017 Results

We evaluate the performance of four variants of the SHEF/CNN model:

• SHEF/CNN-F: Uses only the MLP over the engineered features trained over HTER.

• SHEF/CNN-C: Uses only the character-level convolution stacks trained over HTER.

• SHEF/CNN-C+F: Uses both engineered features and character-level information trained over HTER.

• SHEF/CNN-C+F+M: Uses the same architecture of SHEF/CNN-C+F, but the model is trained through multi-task learning over the values listed in Section 4.

Table 2 illustrates the Pearson, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) scores for the scoring task, and Spearman correlation scores for the ranking task of each language pair. Boldface values represent the best scores obtained across SHEF/CNN models. We also include the results from the official baseline and from the top performing team (POSTECH). The results reveal that, although we outperform the task baseline for English-German, the SHEF/CNN models do not offer competitive performance to state-of-the-art QE systems that rely on resource-heavy strategies. Nonetheless, some valuable observations can be drawn from the results. Combining engineered features with character-level clues yields a more reliable model than simply using either of them alone, which suggests that character-level clues can be a valuable source of complementary information to engineered features. Our multi-task learning setup did not improve on the results of our model. We hypothesize that the secondary output labels could not offer a significant volume of complementary information to the model.

7 WASSA 2017 Results

Table 3 illustrates the Pearson and Spearman correlation scores for each emotion. We compare the
performance of all SHEF/CNN variants described in the previous sections and also include the official task baseline and the three top performing approaches in the EIA task: the Prayas, Emkay and venkatesh-1729 systems.

The SHEF/CNN models are outperformed by a noticeable margin by strategies that heavily employ engineered features and external resources, such as large databases of emotion intensity labels. Nonetheless, our results reveal the same phenomenon highlighted in our experiments with QE: for all emotions, combining engineered features with character-level information yields better performance scores than using only one of these information sources. This serves as further evidence that character-level convolutions can be effectively used as a complement to engineered features.

Our multi-task learning approach only managed to obtain performance improvements for anger. We believe this is due to fact that the positivity values present in the Stanford Sentiment Treebank, which is used in our multi-task setup, accurately quantify only the degree with which the reviewer is pleased, and hence happy, or displeased, and hence angry. Because the other emotions in the WASSA 2017 task do not commonly permeate the act of writing a product review, the multi-task setup was not able to help the model trained for them.

8 Conclusions

We introduced a text regression model that uses deep convolution neural networks and multi-layer perceptrons to combine the character-level information present in texts with the information from engineered features.

We tested several variants of our model in two text regression shared tasks: the sentence-level Quality Estimation task of WMT 2017 and the Emotion Intensity Analysis task of WASSA 2017. We found that, although our model is not able to outperform classic resource-heavy strategies, combining character-level data with engineered features results in noticeable performance gains for both tasks. We also found that, although multitask learning can in principle help our model, the setup must be carefully crafted, otherwise it compromises its performance.

We plan to further test with other tasks the hypothesis that character-level convolutions constitute an intuitive way of complementing the performance of typical feature-based text regression models. We will also test more elaborate convolution architectures, such as using stacked LSTMs.
Acknowledgments

This work was supported by the QT21 project (H2020 No. 645452).

References


One-parameter models for sentence-level post-editing effort estimation

Mikel L. Forcada
mlf@ua.es
Miquel Esplà-Gomis
mespla@dlsi.ua.es
Felipe Sánchez-Martínez
fsanchez@dlsi.ua.es
Departament de Llenguatges i Sistemes Informàtics, Universitat d’Alacant, E-03690 Sant Vicent del Raspeig, Spain.

Lucia Specia
l.specia@sheffield.ac.uk
Department of Computer Science, the University of Sheffield, Sheffield S1 4DP, United Kingdom

Abstract
Methods to predict the effort needed to post-edit a given machine translation (MT) output are seen as a promising direction to making MT more useful in the translation industry. Despite the wide variety of approaches that have been proposed, with increasing complexity as regards their number of features and parameters, the problem is far from solved. Focusing on post-editing time as effort indicator, this paper takes a step back and analyses the performance of very simple, easy to interpret one-parameter estimators that are based on general properties of the data: (a) a weighted average of measured post-editing times in a training set, where weights are an exponential function of edit distances between the new segment and those in training data; (b) post-editing time as a linear function of the length of the segment; and (c) source and target statistical language models. These simple estimators outperform strong baselines and are surprisingly competitive compared to more complex estimators, which have many more parameters and combine rich features. These results suggest that before blindly attempting sophisticated machine learning approaches to build post-editing effort predictors, one should first consider simple, intuitive and interpretable models, and only then incrementally improve them by adding new features and gradually increasing their complexity. In a preliminary analysis, simple linear combinations of estimators of types (b) and (c) do not seem to be able to improve the performance of the single best estimator, which suggests that more complex, non-linear models could indeed be beneficial when multiple indicators are used.

1 Introduction
Over the last decade, the interest of the industry in machine translation (MT) has grown, mainly as a consequence of high demand and improvements in translation quality. Modern MT systems have proven to lead to productivity gains (Plitt and Masselot, 2010; Guerberof Arenas, 2009) when used to generate draft translations that are then post-edited (corrected) before publishing (Krings and Koby, 2001; O’Brien and Simard, 2014). However, not all the translations produced by MT systems are worth post-editing. In some cases, it would be faster to translate them from scratch. As a result, a strong focus has been put into developing methods for estimating the quality of machine-translated sentences (Blatz et al., 2004; Specia et al., 2009) to identify those translations that may harm productivity if provided to post-editors. Several methods are
Most approaches to MT quality estimation (QE) work at the sentence level, although there are also approaches that try to estimate the quality at the word or document levels. Sentence-level QE models predict translation quality in terms of post-editing (PE) time, number of edits needed, and other related metrics (Specia, 2011; Bojar et al., 2014). This paper focuses on sentence-level MT QE and measures quality in terms of PE time. This setting has the important advantage that the time predicted for each machine-translated sentence can be directly used to budget a translation job.

As will be discussed below, existing PE time estimators use many parameters and combine rich features extracted from source sentences and their raw MT output, often with the help of one or more pseudo-references obtained using additional MT systems. They are, however, still far from producing human-like predictions (with Pearson correlations between predicted and human effort metrics plateauing around 0.65, (Bojar et al., 2013, 2014)). To try to understand the problem better, we explore the use of three types of very simple, one-parameter, black-box PE time estimators: (a) a weighted average of PE times in the training set, where weights are an exponential function of edit distances computed between the current sentence (source or raw MT) and training sentences (source or raw MT), so that the contribution of nearest examples is more important; (b) a simple model that learns a unit PE time, either per character or per word, and multiplies it by the length of the current sentence (source or raw MT); and (c) logarithmic probabilities obtained by applying a statistical language model of the source or the target language respectively to the source or raw MT.

The results show that some of these very simple models outperform not only rather strong baselines, but also some complex, multi-parameter estimators participating in the WMT13 (Bojar et al., 2013) and WMT14 (Bojar et al., 2014) PE time estimation contests. Results can be taken as an indication that one should take a step back and first analyse simple models with intuitive interpretations, to only then carefully and gradually increase their complexity, before blindly attempting sophisticated machine learning approaches. In a preliminary analysis, simple linear combinations of estimators of types (b) and (c) above does not seem to be able to improve the performance of the single best estimator, which may be taken as an indication that more complex, non-linear models should be considered when multiple indicators are used.

2 Settings and models

2.1 Corpora

We have conducted experiments using the data sets for English-to-Spanish (en→es) translation, which are publicly available as part of the quality estimation shared Task 1.3 of WMT133 (Bojar et al., 2013) and WMT144 (Bojar et al., 2014); Table 1 describes these data sets. For the experiments in this paper the corpora were pre-processed using the vanilla word tokenizer available in the Python NLTK package (Bird et al., 2009).

2.2 Notation and evaluation

The training data consists of a set of \( N \) triplets \( \{(s_i, MT(s_i), t_i)\}_{i=1}^{N} \) where \( s_i \) is a source sentence, \( MT(s_i) \) its raw MT output, and \( t_i \) the time taken to post-edit \( MT(s_i) \) into an adequate...
The goal is to predict the PE time for a new set of $M$ source sentences and their translations, $\{(s_i, MT(s_i))\}_{i=1}^M$.

As in the WMT13 and WMT14 contests, performance will be measured over the test set as the mean absolute error (MAE) of the prediction $\hat{t}_j$, that is,

$$\text{MAE} = \frac{1}{M} \sum_{j=1}^M |\hat{t}_j - t_j|.$$ 

In addition to this, Pearson’s correlation $r$ between the predicted and measured times will also be reported as a secondary comparison metric.

The best parameter for each model will be determined through minimization of the MAE over the training set, as will be explained in the next section.

2.3 Models

In what follows we describe the three one-parameter models we experimented with in order to predict PE time.

2.3.1 Weighted-average model ($\text{Avg}$)

This model estimates the PE time needed to turn $MT(s_j)$ into an adequate translation of $s_j$ as the weighted average

$$\text{Avg}_u(\alpha, x_j) = \frac{1}{N} \sum_{i=1}^N w(\alpha, x_i, x_j) t_i,$$

controlled by a single parameter $\alpha$, whose weights $w(\alpha, x_i, x_j)$ depend on edit distances through

$$w(\alpha, x_i, x_j) = \frac{e^{-\alpha \text{ED}_u(x_i, x_j)}}{\sum_{i=1}^N e^{-\alpha \text{ED}_u(x_i, x_j)}},$$

where $\text{ED}_u(x_i, x_j)$ is the edit distance between $x_i$ and $x_j$, $u$ is the unit used to compute it, either characters ($u = c$) or words ($u = w$), and $x_i$ (resp. $x_j$) is either the source sentence $s_i$ (resp. $s_j$) or its machine translation $MT(s_i)$ (resp. $MT(s_j)$). For positive values of $\alpha$, the contribution $w(\alpha, x_i, x_j)$ of $t_i$ diminishes with the distance between either the source sentences or between their raw machine translations. In particular:

- When $\alpha = 0$, $\text{Avg}_u(0, x_j) = \frac{1}{N} \sum_{i=1}^N t_i$ for all $j$, that is, the arithmetic average of measured PE times; we will refer to this as the naive zero-parameter average;
- When $\alpha \to +\infty$, the $t_i$ corresponding to the minimum $\text{ED}(x_i, x_j)$, that is, the nearest neighbour, is selected. In what follows, this predictor will be referred to as $\text{NN}_u(x_j)$.

It is expected that a careful choice of $\alpha$ in $[0, +\infty)$ will give a better estimate by assigning a higher weight to closer examples. The weighted average effectively acts as a “soft nearest-neighbour” predictor.

### Table 1: Translation direction and number of training and test instances for the corpora used in the experiments.

<table>
<thead>
<tr>
<th>Translation direction</th>
<th>No. of segments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
</tr>
<tr>
<td>WMT13 en → es</td>
<td>803</td>
</tr>
<tr>
<td>WMT14 en → es</td>
<td>650</td>
</tr>
</tbody>
</table>
To find the optimum value of $\alpha$, the training corpus is randomly split in two sets: 80% of the samples are used to compute the edit distances and the remaining 20% are used as a development set.

The idea behind the weighted-average model bears some resemblance to the work by Béchara et al. (2016), where a semantic textual similarity (between the source sentences) is used to select a close example: instead of predicting time, Béchara et al. (2016) predict the BLEU score for sentences that do not have a reference translation available, using as reference that for the close example. Note the weighted-average model is clearly a black-box model, as it does have access to the inner workings of the MT system whose quality is being predicted. It is also an example-based model that computes a prediction for the current segment by looking up measured times for existing segments in a training set.

2.3.2 Models based on the PE time per segment length unit ($T_{Len}$)

These very simple, one-parameter estimators predict the PE time $t_j$ as

$$T_{Len}(a,s_j) = a \cdot \text{len}_{u}(s_j),$$

where $s_j$ is a source sentence $s_j$, or its machine translation $MT(s_j)$, and $\text{len}_{u}(s_j)$ is the length of $s_j$ in characters ($u = c$) or words ($u = w$). Note that the coefficient $a$, which is obtained by directly minimizing the MAE over the whole training corpus, has an easy interpretation in seconds per character or seconds per word, respectively. Again, this is a black-box model, which, in addition, only looks at one property of the source or machine-translated segment: its length. When $s_j = s_j$, it simply predicts that PE time grows linearly with the source sentence. When $s_j = MT(s_j)$, the estimate is similar if one assumes that target-segment length grows linearly with source-segment length. Note, however, that this predictor pays very little attention to the actual post-editability of the translation:

- Any MT output having the same length would have the same post-editing time, regardless of the actual target words.
- Truncated or abnormally short MT outputs would be consistently —and often incorrectly— estimated to be easier to post-edit.

These models are therefore expected to be very limited predictors of PE time.

2.3.3 Statistical language models

Source-language (SLM) and target-language models (TLM), trained on a subset of the WMT13 translation task data\(^5\) (an interpolated combination of Europarl and News Commentary data) were used to compute the logarithm of the probability of $s_j$ and $MT(s_j)$, respectively. This is then multiplied by a coefficient $a$ which is also optimized to minimize MAE on the whole training set. Language models are common indicators used in QE but also have important limitations as PE time predictors:

- A TLM basically measures the fluency of the translation (Specia et al., 2013, p. 80), and would estimate more fluent translations as easier to post-edit, regardless of their actual semantic relationship to the source sentence.
- A SLM would in contrast measure the complexity of the translation (Specia et al., 2013, p. 80), or, if the language model was trained on texts similar to those on which the MT system was trained, its expectedness. Nevertheless, its predictive power may be limited when applied to a system that was not trained on similar data (or to a rule-based system).

\(^5\)http://www.statmt.org/wmt13/translation-task.html
It is however worth mentioning that language models are amongst the best performing features for sentence-level MT QE (Felice and Specia, 2012; Shah et al., 2015) and are therefore included in most models submitted to the WMT QE shared tasks.

3 Results and discussion

3.1 Performance of one-parameter predictors

Tables 2 and 3 summarize the results for the one-parameter models, placing them in the context of the results obtained by other WMT13 and WMT14 participants. The performance of the zero-parameter naïve average, that is, the one obtained using for all test segments the average time in the training set as a fixed estimate, and the four nearest-neighbour estimates $\text{NN}_n(x_j)$ (see Section 2.3.1) are also provided for completeness. The main metric used in the discussion is MAE, the official metric in WMT13 and WMT14. Pearson correlations, also provided, roughly follow the same trend, and their comparison would lead to similar conclusions (but see Section 3.1.3 for a more detailed discussion).

3.1.1 WMT13 results

When ordering results by MAE, as in (Bojar et al., 2013), the one-parameter models (Avg, $TLen$, SLM and TLM) outperform at least 2 of the 14 participants, with $TLen$ models actually outperforming 8 of them and the TLM outperforming 12 of them. The baseline system (Baseline bb17 SVR), using support vector regression and a well-known set of 17 black-box features (Specia et al., 2013) also outperforms 8 of the 14 participant models. It is worth mentioning that language models are also included as features in this baseline set; that is, the baseline system is a superset of the single-parameter models using LM features. Nevertheless, the TLM outperforms the baseline by a rather large margin. This result in particular may reveal problems not only present in the baseline but also in other participating submissions such as (a) additional features adding noise that the learning algorithm could not adequately handle, (b) the regression architecture used (for instance, support vector regression in the case of the baseline) not being adequate, (c) optimization not being good enough (for instance, due to an incorrect choice of hyperparameters or to incomplete convergence), or (d) over-fitting to a rather small training set. All these reasons are in principle possible and worth a closer examination. One of the participating systems is even outperformed by the naïve-average zero-parameter estimate, and two of them by one of the (also parameterless) nearest-neighbour estimates.

In general terms, computationally simpler (linear) $TLen$ models perform better than the more complex (sum of exponentials containing edit distances) Avg models, while the outstanding performance of TLM and SLM may be explained by the fact that they were trained on the same data as the system whose quality was estimated — therefore, in this last case, the black-box assumption would not hold entirely.

3.1.2 WMT14 results

When ordering results by MAE, as in (Bojar et al., 2014), one-parameter models have a more modest performance in this dataset, beating only 3 out of the 10 submissions: one of them (FBK-UPV-UEDIN/NOWP), which uses hundreds of features obtained from the best 100,000 translations produced by a purposely-trained statistical MT system; another one, the baseline, a rather strong model (17 features), equivalent to the WMT13 baseline. Contrary to what happened for WMT13, character-level Avg models seem to perform slightly better than the $TLen$ model and the SLM and TLM models; these language models were trained on the same data as for WMT13, whereas the MT systems evaluated in WMT14 were not. All zero-parameter models (naïve average, nearest-neighbour) rank below all participants.

We note that the performance of the official baseline system (Baseline bb17 SVR) is
particularly poor on this data set. The reason for that were the ranges used for the grid search to optimize the hyperparameters of the support vector machine model, which were different from those used in the WMT13 model. If the same ranges are used, the baseline reaches a MAE of 17.65, which would place it above all of the one-parameter models and above two of the participating systems. This issue shows further evidence that more complex models need to be carefully crafted, with special attention dedicated to their hyperparameters.

3.1.3 Analysis

How can length be such a reasonable estimator? In both datasets, length-based TLen, u(xj) estimators show a rather competitive performance, in spite of the obvious limitations discussed in Section 2.3.2. This may be due to the fact that the output of a single MT system was post-edited and, therefore MT quality and, consequently, the post-editing effort across the segments produced by the MT systems is quite stable, effectively yielding a roughly constant per-word or per-character post-editing time and therefore making length a reasonable estimator in this case. It would therefore be reasonable to expect performance to have been clearly worse if output from at least two MT systems with very different levels of quality had been post-edited.

Pearson correlations between predictions. In addition to the Pearson correlation with the test set, we have computed the Pearson correlation coefficient between the predictions of participating submissions—which are available at the WMT13\(^7\) and WMT14\(^8\) websites—and our best one-parameter TLen, Avg, and TLM models. In general terms, systems showing a good correlation with the one-parameter models happen to perform similarly, an indication that their predictions are very similar for test sentences. There are, however, interesting exceptions. An example of moderate correlation among predictors but similar performance is the SHEF FS submission to WMT13, which has correlation coefficient with $\text{Avg}_\alpha (\alpha = 0.256, MT(s_j))$ of 0.67 and an absolute difference in MAE of only 0.70. This may point at a certain complementarity between the two predictors, which seem to predict differently for many test sentences in spite of similar MAE performance.

Discrepancies between MAE and correlation coefficients may easily be explained in terms of scaling; in fact, by simply scaling the outputs of all participating predictors one can obtain better MAE results, as shown in tables 2 and 3.

---

\(^6\)The actual time per unit, both in the training set and the test set, indeed shows a rather peaked distribution density around the average values used by the length predictors.

\(^7\)http://www.statmt.org/wmt13/quality_estimation_data/QE_WMT13_submissions_task1.3_sentence.zip

\(^8\)http://www.statmt.org/wmt14/quality_estimation_data/QE_WMT14_submissions_task1.3_sentence.zip
<table>
<thead>
<tr>
<th>System ID</th>
<th>MAE</th>
<th>r</th>
<th>Scaling</th>
<th>Scaled MAE</th>
<th>∆MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBK-UEDIN Extra</td>
<td>47.5</td>
<td>0.66</td>
<td>0.009</td>
<td>46.5</td>
<td>1.0</td>
</tr>
<tr>
<td>FBK-UEDIN Rand-SVR</td>
<td>47.9</td>
<td>0.66</td>
<td>1.062</td>
<td>47.6</td>
<td>0.3</td>
</tr>
<tr>
<td>TLM(α = -1.421, MT(s_j))</td>
<td>48.8</td>
<td>0.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNGL SVR</td>
<td>49.2</td>
<td>0.67</td>
<td>1.164</td>
<td>47.6</td>
<td>1.6</td>
</tr>
<tr>
<td>CNGL SVRPLS</td>
<td>49.6</td>
<td>0.68</td>
<td>1.104</td>
<td>48.9</td>
<td>0.7</td>
</tr>
<tr>
<td>SLM(α = -1.249, s_j)</td>
<td>49.7</td>
<td>0.64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMU slim</td>
<td>51.6</td>
<td>0.63</td>
<td>0.902</td>
<td>50.6</td>
<td>1.0</td>
</tr>
<tr>
<td>Baseline bb17 SVR</td>
<td>51.9</td>
<td>0.61</td>
<td>1.103</td>
<td>51.4</td>
<td>0.5</td>
</tr>
<tr>
<td>TLen_a(α = 3.226, MT(s_j))</td>
<td>52.0</td>
<td>0.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLen_a(α = 3.468, s_j)</td>
<td>52.3</td>
<td>0.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFKI linear6</td>
<td>52.4</td>
<td>0.64</td>
<td>0.857</td>
<td>50.7</td>
<td>1.7</td>
</tr>
<tr>
<td>TLen_a(α = 0.664, s_j)</td>
<td>52.4</td>
<td>0.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLen_a(α = 0.601, MT(s_j))</td>
<td>52.5</td>
<td>0.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMU full</td>
<td>53.6</td>
<td>0.58</td>
<td>1.095</td>
<td>53.6</td>
<td>0.0</td>
</tr>
<tr>
<td>DFKI pls8</td>
<td>53.6</td>
<td>0.59</td>
<td>0.874</td>
<td>52.1</td>
<td>1.5</td>
</tr>
<tr>
<td>TCD-DCU-CNGL SVM2</td>
<td>55.8</td>
<td>0.47</td>
<td>1.082</td>
<td>55.4</td>
<td>0.4</td>
</tr>
<tr>
<td>TCD-DCU-CNGL SVM1</td>
<td>55.9</td>
<td>0.48</td>
<td>1.083</td>
<td>55.5</td>
<td>0.4</td>
</tr>
<tr>
<td>SHEF FS</td>
<td>55.9</td>
<td>0.42</td>
<td>0.870</td>
<td>54.7</td>
<td>1.2</td>
</tr>
<tr>
<td>Avg_a(α = 0.256, MT(s_j))</td>
<td>56.6</td>
<td>0.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg_a(α = 0.386, s_j)</td>
<td>57.2</td>
<td>0.56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg_a(α = 1.079, MT(s_j))</td>
<td>61.1</td>
<td>0.52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg_a(α = 0.612, s_j)</td>
<td>61.7</td>
<td>0.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN_a(s_j)</td>
<td>62.5</td>
<td>0.41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHEF FS-AL</td>
<td>64.6</td>
<td>0.57</td>
<td>1.054</td>
<td>64.4</td>
<td>0.2</td>
</tr>
<tr>
<td>NN_a(MT(s_j))</td>
<td>67.8</td>
<td>0.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naive zero-parameter average</td>
<td>68.1</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN_a(s_j)</td>
<td>70.1</td>
<td>0.37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIMSI elastic</td>
<td>70.6</td>
<td>0.58</td>
<td>1.804</td>
<td>54.4</td>
<td>26.2</td>
</tr>
<tr>
<td>NN_a(MT(s_j))</td>
<td>71.3</td>
<td>0.30</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2:** Mean absolute error (MAE) and Pearson correlation coefficient (r) for one-parameter (Avg(α, x_j), TLen(α, x_j), SLM(α, s_j) and TLM(α, MT(s_j))) and zero-parameter (naive average, NN_a(x_j)) quality estimators (all shaded) in the context of WMT13 submissions. For WMT13 participants, the results of oracle scaling (see text) are also given: scaling factor, new MAE and variation of MAE.
<table>
<thead>
<tr>
<th>System ID</th>
<th>MAE</th>
<th>$r$</th>
<th>Scaling</th>
<th>Scaled MAE</th>
<th>$\Delta$MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTM-DCU/RTM-SVR</td>
<td>16.77</td>
<td>0.63</td>
<td>0.863</td>
<td>16.29</td>
<td>0.48</td>
</tr>
<tr>
<td>MULTILIZER/MLZ2</td>
<td>17.07</td>
<td>0.64</td>
<td>0.851</td>
<td>16.22</td>
<td>0.75</td>
</tr>
<tr>
<td>SHEFF-lite</td>
<td>17.13</td>
<td>0.61</td>
<td>0.949</td>
<td>17.05</td>
<td>0.08</td>
</tr>
<tr>
<td>MULTILIZER/MLZ1</td>
<td>17.31</td>
<td>0.65</td>
<td>0.835</td>
<td>16.43</td>
<td>0.88</td>
</tr>
<tr>
<td>SHEFF-lite/sparse</td>
<td>17.42</td>
<td>0.61</td>
<td>0.963</td>
<td>17.38</td>
<td>0.04</td>
</tr>
<tr>
<td>FBK-UPV/UEDIN/WP</td>
<td>17.48</td>
<td>0.66</td>
<td>0.812</td>
<td>15.76</td>
<td>1.72</td>
</tr>
<tr>
<td>RTM-DCU/RTM-RR</td>
<td>17.50</td>
<td>0.64</td>
<td>0.814</td>
<td>16.16</td>
<td>1.34</td>
</tr>
<tr>
<td>Avg$_{\alpha=0.217,s_j}$</td>
<td>17.60</td>
<td>0.58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLM$_{\alpha=-0.538,MT(s_j)}$</td>
<td>17.94</td>
<td>0.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLM$_{\alpha=0.281,MT(s_j)}$</td>
<td>18.38</td>
<td>0.57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLM$_{\alpha=-0.521,s_j}$</td>
<td>18.55</td>
<td>0.58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLM$_{\alpha=1.519,MT(s_j)}$</td>
<td>18.59</td>
<td>0.55</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FBK-UPV/UEDIN/NOWP</td>
<td>18.69</td>
<td>0.62</td>
<td>0.758</td>
<td>16.72</td>
<td>1.97</td>
</tr>
<tr>
<td>Avg$_{\alpha=0.794,MT(s_j)}$</td>
<td>18.75</td>
<td>0.54</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLM$_{\alpha=0.327,s_j}$</td>
<td>18.80</td>
<td>0.59</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLM$_{\alpha=1.616,s_j}$</td>
<td>18.84</td>
<td>0.56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg$_{\alpha=0.61,s_j}$</td>
<td>18.86</td>
<td>0.56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USHEFF</td>
<td>21.48</td>
<td>0.57</td>
<td>0.907</td>
<td>21.25</td>
<td>0.23</td>
</tr>
<tr>
<td>Baseline bb17 SVR</td>
<td>21.49</td>
<td>0.54</td>
<td>0.906</td>
<td>21.25</td>
<td>0.24</td>
</tr>
<tr>
<td>$NN_u(s_j)$</td>
<td>21.53</td>
<td>0.37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NN_u(MT(s_j))$</td>
<td>21.80</td>
<td>0.36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naive zero-parameter average</td>
<td>21.93</td>
<td>—</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NN_u(s_j)$</td>
<td>22.14</td>
<td>0.31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NN_u(MT(s_j))$</td>
<td>22.65</td>
<td>0.32</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Mean absolute error (MAE) and Pearson correlation coefficient ($r$) for one-parameter (Avg$_{\alpha,x_j}$, TLM$_{\alpha,x_j}$, SLM$_{\alpha,s_j}$ and TLM$_{\alpha,MT(s_j)}$) and zero-parameter (naive average, $NN_u(s_j)$) black-box quality estimators (all shaded) in the context of WMT14 submissions. For WMT14 participants, the results of oracle scaling (see text) are also given: scaling factor, new MAE and variation of MAE.
In the case of the LIMSI elastic submission to WMT13, the scaling factor of 1.804 leads to the best possible test-set MAE of 54.4, which is much better and closer to that of other systems having similar Pearson’s coefficients. Note that this is an oracle scaling, since the gold-standard time measurements for the test set are used to obtain the best scaling factor; however, it is reasonable to expect that linear scaling on the training set would also have improved this predictor. For the remaining participants in WMT13, oracle scaling factors in the range [0.857, 1.364] lead to small changes in MAE between 0.3 and 1.7 seconds, that is, around 0.6% to 3.4%. These changes would be expected to be even smaller or even negligible if scaling had been learned on the training set.

The scaling picture for WMT14 is also interesting (see Table 3). Oracle scaling factors in the range [0.758, 0.949] lead to improvements in MAE in the range [0.24 s, 1.97 s], which are sometimes as large as 12%. The improvements are particularly substantial for FBK-UPV-UEIN/NOWP (-1.97 s, scaling 0.758), FBK-UPV-UEIN/WP (-1.72 s, scaling 0.812) and RTM-DCU/RTM-RR (-1.34 s, scaling 0.814), which would explain the discrepancies between Pearson correlation and MAE mentioned above. It is reasonable to expect that a scaling factor obtained using the training set would have also made a difference in the test-set MAE in these three cases.

Approximating complex predictors with just one parameter: Finally, it is worth noting that some systems showing a good Pearson correlation with the models presented in this paper use very many features and parameters. In particular, the Pearson correlation coefficient of the RTM-DCU/RTM-SVR submission to WMT14 with $TLen_c(s_j)$ is 0.90 (the absolute difference in MAE is 1.78) and, while the latter has one feature and a single parameter, the former uses hundreds of features and several other sources of information. Oracle scaling of RTM-DCU/RTM-SVR slightly improves its test-set MAE to 16.23 s.

3.2 Performance of few-parameter predictors

In view of the surprisingly competitive results obtained with some of the single-parameter models presented here, one would immediately ask the following question: would performance improve further by using linear combinations of them? We take the following six linear predicting features: the length-based $TLen_c(s_j)$, $TLen_w(s_j)$, $TLen_c(MT(s_j))$, and $TLen_c(MT(s_j))$, and the two statistical-language models SLM and TLM. For the study, we leave aside the weighted-average features as they are computationally more intensive to use and to train, do not have a linear form, and need a separate development set to be trained.

All $2^6-1 = 63$ possible subsets of these 6 features are studied. We take linear combinations of each subset and use the multidimensional downhill simplex algorithm of Nelder and Mead (1965) as implemented in the Python library *scipy* to search the coefficients that minimize the training set MAE. For more than two parameters, the result of the minimization heavily depends on the starting point (this is expected in view of the strong collinearity, for instance, between length features). Therefore, to ensure the best possible training set MAE, for each subset, 50 searches are performed with starting parameters randomly sampled from the zero-average, unit-variance normal distribution $N(0, 1)$. The results are shown in Table 4.

As expected, the lowest training set MAE is found when all six features are used; however, the resulting test set MAE does not improve the results obtained with the best single-parameter predictor: 48.8 s for WMT13 (same as TLM alone) and 18.39 s for WMT14 (almost the same as TLM alone). Conversely, some combinations having worse training-set MAE get better test set MAE results, such as 48.22 s for a mixture of just $TLen_w(s_j)$ and TLM($MT(s_j)$) in WMT13.

---

9Exhaustive search in feature spaces is sometimes performed in QE, e.g. (Scarton et al., 2015).
Table 4: Post-editing time prediction using a small number of linear features: number of features, best combination, training-set MAE, and test-set MAE.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Features</th>
<th>Best combination</th>
<th>Train MAE</th>
<th>Test MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT13</td>
<td>TLM</td>
<td>41.3</td>
<td>48.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_w(s_j)) + TLM</td>
<td>41.0</td>
<td>48.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_w(s_j)) + TLen(_w(MT(s_j))) + TLM</td>
<td>40.7</td>
<td>49.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_w(s_j)) + TLen(_c(MT(s_j))) + SLM + TLM</td>
<td>40.6</td>
<td>48.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_c(s_j)) + TLen(_c(MT(s_j))) + SLM + TLM</td>
<td>40.5</td>
<td>48.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_c(s_j)) + TLen(_c(MT(s_j))) + SLM + TLM</td>
<td>40.5</td>
<td>48.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All 6</td>
<td>40.5</td>
<td>48.8</td>
<td></td>
</tr>
<tr>
<td>WMT14</td>
<td>SLM</td>
<td>15.92</td>
<td>18.59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_c(MT(s_j))) + SLM</td>
<td>15.60</td>
<td>18.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_c(MT(s_j))) + SLM</td>
<td>15.57</td>
<td>18.51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_c(s_j)) + TLen(_c(MT(s_j))) + SLM</td>
<td>15.53</td>
<td>18.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TLen(_c(s_j)) + TLen(_c(MT(s_j))) + SLM</td>
<td>15.53</td>
<td>18.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All 6</td>
<td>15.53</td>
<td>18.39</td>
<td></td>
</tr>
</tbody>
</table>

or 18.2 s for a mixture of TLen\(_w(s_j)\), TLen\(_c(MT(s_j))\) and TLM\(_w(MT(s_j))\) for WMT14. These results may be a possible indication of over-fitting or a limitation of a simple linear regressor.

3.3 Budgeting translation jobs

An interesting use of PE time predictors is budgeting a PE job, when post-editors are paid by the hour. Given a new translation job, an estimate of time to complete that job may easily be obtained by summing up the predicted PE time over all segments. This is a very practical application of QE.

Disregarding the actual hourly rate (a constant factor), a good estimate of the usefulness for budgeting may be given by studying the Pearson correlation between the total time predicted for a job by a certain estimator and the actual total time for that job.

To simulate that, we repeatedly and randomly extract PE jobs \(\{(s_j, MT(s_j), t_j)\}_{j=1}^n\) of \(n = 100\) sentences from each of the test sets without replacement. Over each one of these sets, we compute the Pearson correlation between the predicted total time and the actual total measured time. The actual regression coefficients obtained vary with the number of random jobs, but their values for job sizes of 0.4, 0.8, 1.0, and 2.0 times the size of the test set and for a fixed number of 1000 jobs show consistent relative trends. The results for a number of jobs equal to the number of segments in the test set are shown in Table 5.

As can be seen, the Pearson correlation reported for the best single-parameter predictors is almost the same as that for the winning system in WMT13, and slightly worse in WMT14. This would suggest that, at least for these datasets, simple predictors could be used instead of very complex predictors having a large number of features and parameters with a very small loss in budgeting accuracy.
4 Concluding remarks

The results obtained by very simple, one-parameter MT QE models happen to be surprisingly competitive with those obtained by complex QE models using strong learning algorithms, tens, hundreds or thousands of features, and, sometimes, additional resources such as existing, custom-trained, or external MT systems. The findings in this study lead us to make the following recommendations for researchers in MT QE:

- First, look at what can be done with very simple models before using a sledgehammer to crack nuts, in order to get an idea of the performance one could obtain and hopefully improve. As some of the features used in the simple models proposed here are usually part of participants’ complex models, the modest performance they obtain may be due to noise introduced by new features that could not be filtered out by the regressors (probably as a result of a non-optimal training process), to learning problems such as over-fitting to the training set, to non-optimal hyper-parameter choice, to incomplete convergence, or to the shortcomings of the regressors used (as revealed by the oracle scaling described in Section 3.1.3); the actual reasons are probably worth a closer analysis.

- Then, incrementally explore more complex models; linear combinations of a few carefully selected features do not seem to help much; therefore, one should probably consider simple non-linear models. The results of this analysis may be expected to shed some light on the problem.

Finally, a better understanding of the contribution of each feature to the QE models using them could open the door to using, in real-life QE scenarios, feasible and computationally simpler predictors.

Acknowledgements: Work supported by the Spanish government through the EFFORTUNE (TIN2015-69632-R) project and through grant PRX16/00043 for Mikel L. Forcada, and by the European Commission through the QT21 project (H2020 No. 645452).

References


Fine-grained word-level machine translation quality estimation

Abstract

This paper describes research on the prediction of specific types of errors in the output of machine translation (MT) systems. For the first time, we isolate errors and build error-specific models for translations produced by statistical and neural MT systems, both annotated with a fine-grained error taxonomy. We explore the state-of-the-art word-level quality estimation approach with different training strategies to draw conclusions about the predictability of the different types of word-level MT errors, as well as about the best strategy to detect them.

1 Introduction

Word-level machine translation (MT) quality estimation (QE) is usually formulated as the task of automatically identifying which words need to be edited (e.g., deleted or replaced) in a translation produced by an MT system. In practice, with very few exceptions (Bojar et al., 2014), this task has been addressed as a binary classification problem, where words are labelled as OK (the word should be kept as is) or BAD (the word should be edited). This simplification of the task is common because of the lack of sizeable data sets with fine-grained error annotation. However, the resulting predictions are not informative enough. A key resource in the objective of obtaining more specific quality estimates is the detailed taxonomy of errors proposed by Lommel et al. (2014) – MQM – whose core version has 19 error categories: two general error types (accuracy and fluency) and 17 fine-grained error types, 4 of them related to accuracy and 13 to fluency.

We use neural and statistical publicly available MT data with both post-editing and error annotation according to MQM (Specia et al., 2017). By mixing post-editing and error annotation, we are able to isolate and study specific errors. We do so by building on the state of the art method proposed by Kim et al. (2017), which was the best performing in the word-level QE shared task at WMT17 (Bojar et al., 2017). The objectives and contributions in this paper are: (i) to draw conclusions about the nature of the different types of word-level errors in MT output: are all of them equally difficult to predict? are they correlated? is it better to identify them separately, taking the rest of errors identified as the context? and (ii) to identify the best strategy to detect these errors using a state-of-the-art approach. We perform a number of experiments moving from the use of generic models built on large amounts of training data to small and error-specific models. In addition, we consider, for the first time, both statistical MT (SMT) and neural MT (NMT) output for the same source segments.

Section 2 describes the data sets and Section 3, the word-level MT QE approaches used in in the experiments. Section 4 presents our results.

2 Data sets

We use data released by Specia et al. (2017). Each instance contains a source language sentence, the translation generated by an MT system, its professionally post-edited version, and MQM-annotations on the MT output. Every data set contains 2,000 instances of this type. For our experiments, we chose English→German (EN–DE) as language pair, given that it has translations both by an SMT system and an NMT system. This allows us to compare the performance of the models built for the same language pair but with very error different error distributions.

Table 1 summarises the total and ratio of all error types occurring in our data sets. As can be seen, the number of errors for the SMT translations is noticeably higher than for NMT. This dif-
ference is even higher for Word order. This is in
line with previous reports that NMT produce more
fluent translations than SMT (Bentivogli et al.,

It is worth mentioning that categories Omission
and Missing are not analysed in this work; these
error types identify missing words and, therefore,
they are not errors that can be assigned to words in
the MT output, but rather to gaps between words.
In addition, the QE method we use Kim et al.
(2017) is not built to detect missing words.

<table>
<thead>
<tr>
<th>Error type</th>
<th>SMT Freq.</th>
<th>NMT Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>552</td>
<td>321</td>
</tr>
<tr>
<td>Agreement</td>
<td>576</td>
<td>104</td>
</tr>
<tr>
<td>Extraneous</td>
<td>555</td>
<td>274</td>
</tr>
<tr>
<td>Incorrect</td>
<td>909</td>
<td>521</td>
</tr>
<tr>
<td>Mistranslation</td>
<td>1.798</td>
<td>1,103</td>
</tr>
<tr>
<td>Part of speech</td>
<td>172</td>
<td>275</td>
</tr>
<tr>
<td>Spelling</td>
<td>177</td>
<td>147</td>
</tr>
<tr>
<td>Tense</td>
<td>210</td>
<td>64</td>
</tr>
<tr>
<td>Terminology</td>
<td>151</td>
<td>185</td>
</tr>
<tr>
<td>Typography</td>
<td>972</td>
<td>171</td>
</tr>
<tr>
<td>Unintelligible</td>
<td>654</td>
<td>0</td>
</tr>
<tr>
<td>Untranslated</td>
<td>120</td>
<td>24</td>
</tr>
<tr>
<td>Word order</td>
<td>104</td>
<td>271</td>
</tr>
<tr>
<td>Word form</td>
<td>3,163</td>
<td>700</td>
</tr>
<tr>
<td>TOTAL</td>
<td>8,965</td>
<td>3,860</td>
</tr>
</tbody>
</table>

Table 1: Frequency and proportion of the 14 relevant MQM fine-grained error types in our data sets.

3 Word-level MT QE approaches

For our experiments we implement the word-level
QE method proposed by Kim et al. (2017). It
uses a multi-task learning approach that combines
a predictor model, which can be thought of as a
basic NMT model, and a word-level QE model,
which builds on the output of the predictor. We
pre-trained a predictor model on 5,642,756 sen-
tence pairs from the parallel corpus of the WMT17
EN–DE News MT shared task (Bojar et al., 2017).
For the word-level estimator, we experimented
with different approaches:

1. Generic classifier: single model trained on
generic data that does not differentiate among
MQM error types, i.e. only annotated with
OK/BAD labels, and evaluated in predict-
ing each MQM error type separately (Sec-
tion 4.1). This model can be trained on larger
amounts of cheaper, OK/BAD data, but it is not
informed about specific error types.

2. Error-specific classifier: 14 different mod-
els, each trained to identify an error type
(Section 4.2). Since MQM-annotated data is
scarce, the training set is smaller, but learning
each specific error type separately can lead to
more informed models.

3. Multi-class classifier: single multi-class
classifier to predict all 14 error types in the
data sets (Section 4.3). The main challenge
for this approach is that multi-class classi-

fication is usually a more difficult problem
than binary classification. On the other hand,
this model can learn relations between types
of errors appearing together, which could be
useful for this task.

4 Results and discussion

The models for the different approaches described
in Section 3 are built using the data described in
Section 2 and evaluated using standard evalua-
tion metrics from the word-level QE shared task
at WMT: F1 score ($F_1$) for the less common class
(BAD in the case of binary classification), and the
product of the $F_1$ scores of each class ($F_1^j$) for
binary classification (official WMT metric).

4.1 Generic binary classifier

In this experiment we trained a binary classifica-
tion model on generic word-level MT QE training
data, i.e. with OK and BAD labels, without differ-
entiating among errors types. To do so, we used
the training corpus provided for the shared task on
MT QE at WMT17, consisting of 23,000 training
instances. This data set was obtained by using a
phrase-based SMT system, which also allows us
to evaluate the impact of applying a model trained
on a type of MT system on data obtained with a
very different one. To build the test set, we di-
vided the MQM annotated data into different data
sets where only one of the MQM errors was an-
notated as BAD, with the rest of the errors fixed
in the MT output using the post-edited version of
the sentence. In this way, we expanded our data
into a collection of 2,000 instances by error type
data set. In total, we obtained 9,805 unique
instances for SMT, and 8,114 for NMT.
### 4.2 Error-specific classifiers

In this approach, smaller and specialised models were built. To do so, the expanded MQM annotated data described above were used by splitting each error-specific data set into training and testing. As a result 14 training and test sets are obtained, each of them only taking into account one of the MQM errors. The objective of this experiment is to explore the viability of smaller models specifically trained for one kind of error.

Columns 6 and 7 in Table 2 show the results of the evaluation of these models trained on 1,500 training instances from the SMT data set, and evaluated on the remaining 500 instances from the same data set for each error type. In general, most of the results for this approach are substantially better than those obtained with generic training data, specially for error types Incorrect, Typographical, and Unintelligible. In the first case, the performance in terms of $F_1^\text{bad}$ is more than twice the one obtained with the large generic model. In the other two cases it is far higher. It is worth noting

### Table 2: $F_1^\text{bad}$ (%) and $F_1^\text{bad}$ (%) for EN–DE when: using generic MT QE training data (23,000 instances with OR/BAD labels) and MQM-annotated test data (2,000) both for SMT and NMT; using error-specific independent models (a model per error type) trained on MQM-annotated MT QE data sets (1,500) each of them containing only one error type, and tested on the same type of data (500), both for SMT and NMT; and using two multi-class classifiers trained on fully annotated MQM data specifically for SMT and NMT (1,500 training and 500 test each).

<table>
<thead>
<tr>
<th>Error type</th>
<th>Generic</th>
<th>Error-specific</th>
<th>Multi-class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SMT</td>
<td>NMT</td>
<td>SMT</td>
</tr>
<tr>
<td>Addition</td>
<td>16.52</td>
<td>16.01</td>
<td>3.25</td>
</tr>
<tr>
<td>Agreement</td>
<td>8.32</td>
<td>8.00</td>
<td>11.25</td>
</tr>
<tr>
<td>Extraneous</td>
<td>20.57</td>
<td>20.54</td>
<td>4.51</td>
</tr>
<tr>
<td>Mistranslation</td>
<td>21.31</td>
<td>20.02</td>
<td>7.12</td>
</tr>
<tr>
<td>Part of speech</td>
<td>2.69</td>
<td>2.68</td>
<td>2.61</td>
</tr>
<tr>
<td>Spelling</td>
<td>2.71</td>
<td>2.63</td>
<td>1.40</td>
</tr>
<tr>
<td>Tense</td>
<td>9.93</td>
<td>9.90</td>
<td>2.50</td>
</tr>
<tr>
<td>Terminology</td>
<td>4.01</td>
<td>3.96</td>
<td>1.85</td>
</tr>
<tr>
<td>Typography</td>
<td>9.00</td>
<td>8.48</td>
<td>2.31</td>
</tr>
<tr>
<td>Unintelligible</td>
<td>2.38</td>
<td>2.36</td>
<td>-</td>
</tr>
<tr>
<td>Untranslated</td>
<td>2.91</td>
<td>2.88</td>
<td>0.85</td>
</tr>
<tr>
<td>Word form</td>
<td>1.65</td>
<td>1.65</td>
<td>2.32</td>
</tr>
<tr>
<td>Word order</td>
<td>22.39</td>
<td>20.58</td>
<td>4.79</td>
</tr>
</tbody>
</table>

Weighted avg. $F_1^\text{bad}$: 15.59, 14.67, 4.67, 4.41

$F_1^\text{bad}$ for Generic: 31.23, 29.44, 11.66, 11.51

$F_1^\text{bad}$ for Error-specific: 19.83, 6.17

As expected, the performance of the model is better on the SMT data, given that the training data was built from this type of MT system. It is noticeable that the model performs especially well for some error types: Addition and Extraneous, referring to the deletion of words, Mistranslation, i.e. replacement of words, and Word order. The last is specially reasonable, given that the predictor included in the architecture is implemented with an LSTM layer (Kim et al., 2017) which uses the context of the entire segment being translated.

The performance obtained for NMT data is much lower in general. Mistranslation is still among the categories with better results, even though far from the results obtained for SMT. Surprisingly, the detection of errors related to word morphology, namely Word form and Agreement, performed slightly better for NMT than for SMT, even though the training set is more related to the last one. This may indicate that these types of errors are less specific of the type of MT system, i.e. they are similar across MT system types.
that the case of the Unintelligible error is somehow especial: the performance is surprisingly high for this type of error, but this is due to the fact that the only Unintelligible errors annotated in the data set are (incorrectly) tokenised URLs. This made the model learn to identify tokenised URLs, but it is hard to say if other types of Unintelligible errors would have been identified as no alternatives occur in the corpus.

On the other hand, the performance for the error types Addition and Mistranslation was worse. This leads us two possible conclusions: one is that, for some error types, it is more important to have larger amounts of training data than a specialised model; the second one is that these errors are easier to be predicted when they happen surrounded by other types of errors that, in the error-specific data set, were fixed. However, the results obtained by the multi-class classifier do not seem to support the former conclusion.

Columns 8 and 9 in Table 2 show the results obtained for the error-specific classifier trained and evaluated on NMT data. As expected, most of the results obtained in this case are worse than those obtained for SMT, something reasonable if we compare the amount of errors in the data sets between both types of MT (see Table 1), which leads to a smaller amount of error samples on which to train our model for NMT. There are, however, a few exceptions worth mentioning: for error error types Addition and Terminology, both accuracy-related error types.

When compared to the results obtained with the larger generic model, the error-specific models outperform it in for most of the error types. A noticeable case is the error type Agreement, which performed reasonably well with the generic model but for which the result with the error-specific model was much worse. This could be a case in which larger amounts of training data, even if it is generic, helps in the identification of errors.

4.3 A classifier to predict all error types

Columns 10 and 11 of Table 2 show the results of using a multi-class classifier trained on a fraction of the MQM annotated data, both for SMT and for NMT, (1,500 instances for training and 500 for testing each). As expected, the results of this experiment show that training a multi-class classifier on such a small amount of data results in lower performance for most error types. As in the previous experiments, the results are worse for the NMT data set than for the SMT one. Note that the F1 metric was not provided for the multi-class classifier, given that in this case it would be the product of multiplying all the 14 F1 scores for all the error types. This means that if only one of them happens to be zero, the result for this metric will be zero.

5 Conclusions

In this paper we compared several strategies that, building on state-of-the-art word-level MT QE systems, try to predict specific MQM translation errors at the word level. This is an incipient research, which is focused on obtaining hints about which are the best ways of tackling this problem.

The experiments carried out confirm that, in general, it is difficult to use models trained on a specific type of data on very different data sets.

The experiments presented in this paper agree in the fact that using models trained on SMT data to predict MQM error types on NMT data result in poorer results.

Among the strategies evaluated in this work, the one using a multi-task classifier provided the worst results. The experiments described in Section 4.3 show that the task of predicting 14 error types is too complex for such small amounts of training data.

Some types of errors benefit from larger amounts of data rather than from training data specific for these types of errors. This is the case for the Addition and Mistranslation errors, which perform clearly better when more training data is available. However, most error types benefit substantially from the use of specialised training data, even when only small amounts are available. This is obviously the case for error types such as Typography or Incorrect, for which the results improve noticeably when using specialised training data, even when it was obtained on MT data systems very different to those used during evaluation (see columns 2 to 5 in Table 2).

In general, the results of this preliminary study point out that identifying different translation errors is a complex task that requires the use of different approaches. The next steps in this research line will consist in designing a combination of different approaches that will allow to identify the different types of MQM errors in any translation output.
References


R Neural Quality Estimation

Abstract

The performance of quality estimation (QE) methods for machine translation (MT) has drastically improved with the introduction of neural approaches. However, these are rather costly: they either still rely on feature extraction, use complex architectures or, most importantly, require extensive pre-training. In this paper we propose a new low-cost neural approach to QE that requires no feature engineering and can yield performance close to that of state-of-the-art methods without pre-training. In addition, for the first time we apply QE models to the output of both statistical and neural MT systems and highlight the new challenges resulting from the use of neural MT.

1 Introduction

Quality estimation (QE) (Blatz et al., 2004; Specia et al., 2009) aims at predicting the quality of machine translation (MT) outputs without human intervention. Most recent work has focused on QE to predict post-editing (PE) effort, i.e. the process of manually correcting MT output to achieve a publishable quality (Bojar et al., 2014, 2015, 2016, 2017). In this case, QE indicates how much an MT unit (word, phrase, sentence, paragraph and document) needs post-editing. Sentence-level QE scores for instance help to select sentences that are worth post-editing, while word-level QE aims to spot words that need to be edited.

Recently, neural methods have been successfully exploited to improve QE performance. The best-performing system at the WMT'17 shared task on QE for all the three levels (word, phrase and sentence) is purely neural and does not rely on feature engineering (Bojar et al., 2017; Kim et al., 2017b). It uses a modification of the standard neural MT encoder-decoder architecture. It predicts quality using weights assigned by the decoder to the words of actual MT that we seek to evaluate, concatenated with representations of the left and right one-word contexts. This architecture is pre-trained using a significant amount of parallel data, which means days of training using costly memory resources, as well dependence on data availability. It uses a stacked architecture for multi-task learning to make quality labels at different levels interdependent. Another well performing WMT shared QE task system, Unbabel (Martins et al., 2017a,b), also uses an encoder-decoder architecture with bidirectional recurrent neural network (RNN) layers as part of its stacked mixed architecture. The input to this encoder-decoder is a feature set: pre-trained word and part-of-speech embeddings, word alignments and contexts. These neural methods rely on complex architectures, require extensive and costly pre-training or feature engineering. We propose a low-cost QE neural architecture, which can yield performance close to the one of state-of-the-art neural models, but without extensive pre-training. We then show how this performance can further be improved by means of denoising autoencoders (Vincent et al., 2010).

We evaluate our proposed approach on both statistical MT (SMT) and neural MT (NMT) outputs. Existing QE solutions, including feature sets, have thus far only been designed for and applied to SMT. To our knowledge, the only attempt to get an insight on NMT translations is that by Rikters and Fishel (2017). They use attention distributions as an indicator the confidence of the neural decoder on its output. The intuition suggests that “good” translations are characterized by strongly focused attention connections (potentially more literal translations). However, this internal information does not map directly into translation quality: a very weak correlation with human judgements in a small-scale assessment was found. Therefore, this is the first time that experiments are performed with full fledged, quality using weights assigned by the decoder to the words of actual MT that we seek to evaluate.
2 Settings

In this section we describe state-of-the-art and baseline methods, to which we compare the performance of the proposed method.

Baseline systems: We reproduced the WMT’17 baselines as described in (Bojar et al., 2017). For the extraction of the sentence-level features for EN–DE we used the additional resources as provided with the WMT’17 QE shared task.\(^1\) For EN–LV, the corresponding resources were created using the data provided for the WMT’17 news translation task,\(^2\) and the EMEA in-domain corpus (Tiedemann, 2009). To extract word-level features word alignments between source and MT were computed using fast_align (Dyer et al., 2013).

State-of-the-art systems: We re-implemented the single- and multi-layer neural solutions proposed by Kim et al. (2017a,b) (POSTECH). We use the NMT-Keras implementation (Chollet et al., 2015; Peris, 2017) in the Keras tool. To train the predictor module in POSTECH for EN–DE we created a model using only the Europarl corpus (Koehn, 2005) (≈ 2M lines).\(^3\) For the EN–LV model, we used the parallel data available for the WMT’17 news translation task (≈ 2M lines).\(^4\)

Evaluation: The QE datasets released in Specia et al. (2017) for EN–DE (IT domain) and EN–LV (life sciences domain) was used to build and evaluate all QE systems. These datasets contain 28,000 and 18,738 English segments, respectively, and translations obtained with SMT and NMT for each of them, together with their post-editions. We randomly split the data for each language pair into train (25K lines for EN–DE, 16K for EN–LV), development (1K lines) and test (2K lines) sets.\(^5\) In line with the WMT QE campaigns, data labelling was performed as described in Bojar et al. (2017) using the TERCOM toolkit;\(^6\) for sentence-level QE, edit distance scores (HTER) are used as labels, while for word-level QE, binary “good”/“bad” labels are generated from alignments between the MT and its post-edition. We use the official evaluation metrics from the WMT QE evaluation campaigns: Pearson \( \rho \) score, Mean Average Error (MAE) and Root Mean Squared Error (RMSE) for sentence-level QE; the multiplication of \( F_1 \) scores for the two classes (\( F_1 \)-multi) for word-level QE.

The results of our experiments are reported in Table 1 for the baseline and state-of-the-art methods. For POSTECH, we report results for single-layer models without pre-training (to make them directly comparable to results for our approach) and for multi-layer models with pre-training. For EN–DE, a first observation is the crucial improvement in NMT quality compared to SMT (HTER = 0.15 vs. HTER = 0.23), which results in highly imbalanced NMT datasets (for EN–DE, ≈ 54% of all the sentences are 0 HTER vs. ≈ 13% for SMT). For EN–LV, the quality of NMT, limited by the amount of training data, is slightly worse than SMT (HTER = 0.15 vs. HTER = 0.23).

These results show that the performance of baseline methods depends more on the quality of translations than on the type of MT system. For instance, the EN–DE baseline shows a drop of on average 60% in performance for NMT as compared to SMT for both sentence-level and word-level QE, as measured by primary metrics (in bold). These performance changes can be attributed to the imbalanced distributions in datasets, as confirmed by our additional experiments with artificially balanced datasets.

3 Our QE Approach

3.1 Architecture and Implementation Details

The encoder-decoder approach proposes an architecture for sequence-to-sequence prediction problems. This approach has become very popular in natural language processing applications, where inputs and outputs to models are very often sequential, typically natural language data. The architecture works as follows: an input sequence is encoded into an internal representation (roughly, features learned automatically), and then an output sequence is generated from this representation.

\(^{1}\)http://www.statmt.org/wmt17/quality-estimation-task.html
\(^{2}\)http://www.statmt.org/wmt17/translation-task.html
\(^{3}\)Note that the average pre-training time for POSTECH models is around 12 hours on a 12G NVIDIA GPU and the batch size of 50.
\(^{4}\)The following hyperparameters were used: the size of the hidden units of the word Predictor was 500, the word embedding dimensionality was 300, the size of the vocabulary was 30K and the QE vectors size was 75.
\(^{5}\)As only part of the EN–DE SMT data was used for the WMT’17 QE task, we could not use the task’s official split.
The QE architecture we propose is a simple encoder-decoder architecture with 2 RNNs (minRNN, see Figure 1). The input to our model is only MT data, whose quality we seek to predict. We use a softmax layer over the binary labels as the output layer for word-level prediction; and a sigmoid layer to produce real-value predictions for sentence-level QE.

As QE data are usually small, representations learned by our minimal architecture risk to be sub-optimal. We propose a strategy to pre-train these representations using parallel data and MT. To do so, we use a denoising autoencoder that shares the encoder and the decoder with our minRNN architecture. A denoising autoencoder is a specific case of encoder-decoder where the input sequence and the predicted output are the same (autoencoder) but where the input is expected to be slightly noisy. In our case, our noisy input would be MT (e) obtained by translating the source side of a parallel corpus, and our clean output would be the target side of the same corpus (e). During pre-training, the output layer of the minRNN architecture is replaced by a softmax over the output vocabulary, which is identical to the input one, to produce the denoising autoencoder output.

We implemented our architecture using the Keras toolkit with Gated Recurrent Units (GRUs) (Cho et al., 2014) as RNNs. The word embedding dimensionality is set to 300. The size of the hidden units of the encoder and the decoder is 25. The models were trained as proposed by Kim et al. (2017a): at the word-level we set the weight of the “bad” label to 3 to handle the data imbalance. For the softmax scores of the output layer, we empirically choose the threshold value $t = 0.5$ to assign labels (“ok” if $\sigma \geq t$, “bad” if $\sigma < t$). The size of the output vocabulary for the autoencoder is 30K.

For our denoising experiments, we produce noisy MT data by translating a random selection of 300K sentences from Europarl (Euro, 0.57 TER) and an IT in-domain corpus (In-Dom, 0.73 TER) with the help of the Nematus toolkit (Sennrich et al., 2017) using the WMT’16 EN–DE model (Sennrich et al., 2016).

However, producing sufficiently large MT data is impracticable, and we use only MT data ($\sigma = 0$) as RNNs. The word embedding dimensionality is set to 300. The size of the hidden units of the encoder and the decoder is 25. The models were trained as proposed by Kim et al. (2017a): at the word-level we set the weight of the “bad” label to 3 to handle the data imbalance. For the softmax scores of the output layer, we empirically choose the threshold value $t = 0.5$ to assign labels (“ok” if $\sigma \geq t$, “bad” if $\sigma < t$). The size of the output vocabulary for the autoencoder is 30K.

This can be seen as an automatic post-editing system.
datasets can be time-consuming and requires ready-to-use models. Inspired by recent work on monolingual MT exploiting denoising autoencoders (Artetxe et al., 2017; Lample et al., 2017), we attempted to artificially produce MT by altering the word order of reference sentences, as well as deleting a certain amount of words from those sentences. Concretely, we randomly deleted two words from each sentence and shuffled the rest of the sequence, so that the resulting TER score will be the sequence, so that the resulting TER score will

\[ \sum |w_i - \hat{w}_i| \]

be equal to the number of words deleted. Such an approach requires no parallel data or MT models, and hence is a cheap pre-training strategy.

### 3.2 Results

<table>
<thead>
<tr>
<th>model</th>
<th>EN–DE</th>
<th>EN–LV</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMT</td>
<td>0.244</td>
<td>0.143</td>
</tr>
<tr>
<td>NMT</td>
<td>0.234</td>
<td>0.090</td>
</tr>
</tbody>
</table>

\( \Delta \rho = 0.05 \) is a cheap pre-training strategy.

![Table 2: Results of the minRNN architecture for sentence-level QE. They include using only QE training data, as well as using pre-trained representations with Noise, with Euro, or In-Dom.](image)

<table>
<thead>
<tr>
<th>model</th>
<th>( F_1 )-mult</th>
<th>( F_1 )-BAD</th>
<th>( F_1 )-OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN–DE</td>
<td>( \text{minRNN} )</td>
<td>0.281</td>
<td>0.117</td>
</tr>
<tr>
<td>NMT</td>
<td>( \text{minRNN} )</td>
<td>0.180</td>
<td>0.189</td>
</tr>
</tbody>
</table>

Finally, the representation learning with the help of denoising autoencoders shows very promising improvements in QE, up to 12% for our sentence-level QE experiments. Our preliminary observations suggest that more noisy data create more “helpful” representations: e.g., for EN–DE NMT sentence-level, we observe a decrease of \( \Delta \rho = 0.026 \) with Euro MT data of higher quality and an increase of \( \Delta \rho = 0.035 \) with In-Dom MT data of lower quality. Our artificially generated MT seems to be potentially useful for representation learning (e.g., for EN–DE SMT sentence-level, the improvement of \( \Delta \rho = 0.04 \)).

<table>
<thead>
<tr>
<th>model</th>
<th>( F_1 )-mult</th>
<th>( F_1 )-BAD</th>
<th>( F_1 )-OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN–DE</td>
<td>( \text{minRNN} )</td>
<td>0.393</td>
<td>0.305</td>
</tr>
<tr>
<td>NMT</td>
<td>( \text{minRNN} )</td>
<td>0.307</td>
<td>0.176</td>
</tr>
</tbody>
</table>

Results of our experiments are reported in Tables 2 and 3. These results show that our proposed architecture outperforms baselines, as well as POSTECH in equal conditions without pre-training (\( \Delta \rho = 0.05 \) for sentence level, \( \Delta F_1 \)-mult=0.06 for word level). Our solution is more robust towards data imbalance: for instance, for EN–DE NMT sentence-level QE, it yields an improvement of \( \Delta \rho = 0.08 \) as compared to the state-of-the-art. Our approach is also less sensitive towards the amount of training data: for instance, for EN–LV we had 40% less training data than for EN–DE. Thus, for sentence-level EN–LV SMT, POSTECH with pre-training outperforms our system by only \( \Delta \rho = 0.05 \), which is a rather small benefit taking the cost of pre-training into account. Note that our neural system faces the challenge of the varying length of EN–LV NMT translations (\( \sigma^2 = 92 \) for NMT vs. \( \sigma^2 = 73 \) for SMT).

4 Conclusions

We have proposed a basic neural architecture that without any pre-training yields results comparable to those of state-of-the-art QE solutions. We also have provided for the first time a study of the performance of state-of-the-art QE approaches for NMT. In the future, we plan to attempt to improve the performance of our neural architecture by means of efficient representation learning techniques.
References


François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.


Improving Evaluation of Document-level Machine Translation
Quality Estimation

Yvette Graham
Dublin City University
yvette.graham@dcu.ie

Qingsong Ma
Chinese Academy of Sciences
maqingsong@ict.ac.cn

Timothy Baldwin
University of Melbourne
tb@udwin.net

Qun Liu
Dublin City University
qun.liu@dcu.ie

Carla Parra
Dublin City University
carla.parra@adaptcentre.ie

Carolina Scarton
University of Sheffield
c.scarton@sheffield.ac.uk

Abstract

Meaningful conclusions about the relative performance of NLP systems are only possible if the gold standard employed in a given evaluation is both valid and reliable. In this paper, we explore the validity of human annotations currently employed in the evaluation of document-level quality estimation for machine translation (MT). We demonstrate the degree to which MT system rankings are dependent on weights employed in the construction of the gold standard, before proposing direct human assessment as a valid alternative. Experiments show direct assessment (DA) scores for documents to be highly reliable, achieving a correlation of above 0.9 in a self-replication experiment, in addition to a substantial estimated cost reduction through quality controlled crowdsourcing. The original gold standard based on post-edits incurs a 10–20 times greater cost than DA.

1 Introduction

Evaluation of NLP systems commonly takes the form of comparison of system-generated outputs with a corresponding human-sourced gold standard. The suitability of the employed gold standard representation greatly impacts the reliability and validity of conclusions drawn in any such evaluation. With respect to reliability, measures such as inter-annotator agreement (IAA) enable the likelihood of replicability to be taken into account, were an evaluation to be repeated with a distinct set of human annotators. One approach to achieving high IAA is through the development of a strict set of annotation guidelines, while for machine translation (MT), human assessment is more subjective, making high IAA difficult to achieve. For example, in past large-scale human evaluations of MT, low IAA levels have been highlighted as a cause of concern (Callison-Burch et al., 2007; Bojar et al., 2016). Such problems cause challenges not only for evaluation of MT systems, but also for MT quality estimation (QE), where the ideal gold standard comprises human assessment.

Although concern surrounding the reliability of human annotations is by far the most common complaint with respect to human evaluation of MT, the validity of the particular gold standard representation used in a given evaluation is also highly important. When it comes to validity, conventionally speaking, the very fact that human annotators manually generate the gold standard provides reassurance of its validity, as results at least reflect the judgment of one or more members of the target audience, i.e., human users. In the case of there being some “interpretation” of the human annotations, tuned to the particulars of a given task, validity becomes a concern. In recent document-level QE shared tasks, for example, the gold standard is generated through a linear combination of two separate human evaluation components, with weights tuned to optimize mean absolute error (MAE) and variance with respect to gold label distributions. In this paper, we explore the validity of the gold standard, and investigate to what degree tuning the gold standard impacts the validity of the resultant system performance estimates. Our contribution shows the method used to generate the gold standard has a substantial impact on the resultant system ranking, and propose an alternate gold standard representation for document-level quality estimation that is both more reliable and more valid as a gold standard.
2 Background

Document-level QE (Soricut and Echihabi, 2010) is a relatively new area, with only two shared tasks taking place to date (Bojar et al., 2015; Bojar et al., 2016).

In WMT-15, gold standard labels took the form of automatic metric scores for documents (specifically Meteor scores (Denkowski and Lavie, 2011)), and system predictions were compared to gold labels via MAE. A conclusion that emerged from the initial shared task was that automatic metric scores were not adequate, based on the following observation: if the average of the training set scores is used as a prediction value for all data points in the test set, this results in a system as good as the baseline system when evaluated with MAE. The fact that average scores are good predictors is more likely a consequence of the applied evaluation measure, MAE, however, as outlined in Graham (2015). When evaluated with the Pearson correlation, such a set of predictions would not be a reasonable entry to the shared task since the prediction distribution would effectively be a constant and its correlation with anything is therefore undefined. Regardless of the predictability of automatic metric scores when evaluated with MAE, they unfortunately do not provide a suitable gold standard, simply because they are known to provide an insufficient substitute for human assessment, often unfairly penalizing translations that happen to be superficially dissimilar to reference translations (Callison-Burch et al., 2006).

Consequently, for WMT-16, the gold standard was modified to take the form of a linear combination of two human-targeted translation edit rate (HTER) (Snover et al., 2006) scores assigned to a given document. Scores were produced via two human post-editing steps: firstly, sentences within a given MT-output document were post-edited independent of other sentences in that document, producing post-edition 1 (PE1). Secondly, PE1 sentences were concatenated to form a document-level translation, and post-edited a second time by the same annotator, with the aim of isolating errors only identifiable when more context is available, to produce post-edition 2 (PE2). Next, two translation edit rate (TER) scores were computed by: (1) comparing the document-level MT output with PE1, TER(PE1, MT); and TER between PE2 and PE1, TER(PE2, PE1). Finally, these two scores were combined into a single gold standard label, G, as follows:

\[
G = W_1 \text{TER}(PE_1, MT) + W_2 \text{TER}(PE_2, PE_1)
\]

where weights, \(W_1\) and \(W_2\), are decided by the outcome of the following tuning process: \(W_1\) is held static at 1; \(W_2\) is increased by 1 from a starting value of 0 until either of the following stopping criteria is reached: (i) the ratio between the standard deviation and the mean is 0.5 for the official baseline QE system predictions, or (ii) a baseline prediction distribution is constructed by assigning to all prediction labels the expected value of the training set labels. This second case is designed to deal with the degenerate behaviour described above of assigning to each test item the average over the training data, with the stopping criteria being such that the difference between the MAE achieved by such a system and the official baseline MAE is at least 0.1. The final values used to produce official results were \(W_1 = 1\) and \(W_2 = 13\).

The way in which the gold standard is constructed deviates to quite a degree from conventional gold standards, therefore, which raises some important questions. Firstly, it appears that the optimization process is carried out with direct reference to the test set. If so, does such a process overly blur the lines with respect to what is considered true unseen test data?

Secondly, neither of the two TER scores corresponds to a straightforward human assessment, putting into doubt the conventional validity attributed to human-generated gold standards. For example, the component assigned most weight in the final evaluation is \(\text{TER}(PE_2; PE_1)\), and this unfortunately corresponds more closely to a measure of the dependence of the meaning of the sentences within a given document on other sentences in that document, as opposed to the overall quality of the MT output document.

Finally, and most importantly, assigning weights to components of the human evaluation through a somewhat arbitrary optimization process deviates from the expected interpretation of each reported correlation, i.e. the correlation between system predictions of translation quality and the actual quality of translated documents. Including such weights in the construction of a gold standard potentially invalidates the human evaluation, and is unfortunately very likely to exaggerate the apparent performance of some systems while under-rewarding others.
Figure 1: System performance as the weight of the TER(PE\textsubscript{2}, PE\textsubscript{1}) human evaluation component is increased to 13, as in official evaluation, and beyond (WMT-16 document-level QE English to Spanish shared task systems).

To demonstrate to what degree this could be the case, since post-editions employed in the creation of the actual gold standard used to produce results in the shared task are unavailable, we simulate a possible set of TER(PE\textsubscript{1}, MT) and TER(PE\textsubscript{2}, PE\textsubscript{1}) labels for test documents in the following way: A possible set of TER(PE\textsubscript{1}, MT) labels are simulated by relocation of the TER score distribution (of the MT output document with reference translations as opposed to post-edits) to more closely resemble scores of our later human evaluation, before rescaling that score distribution according to the mean and standard deviations (provided in the QE task findings paper) of TER(PE\textsubscript{1}, MT). TER(PE\textsubscript{2}, PE\textsubscript{1}) scores were then reverse-engineered from the correspondence between TER(PE\textsubscript{2}, MT) and gold labels. Final gold labels arrived at through our simulation of TER(PE\textsubscript{1}, MT) and TER(PE\textsubscript{2}, PE\textsubscript{1}) are identical to the original evaluation for $W_1 = 1$ and $W_2 = 13$.

Figure 1 shows correlations achieved by all systems in the shared task when the weight of our simulated TER(PE\textsubscript{2}, PE\textsubscript{1}) component is varied from 1 up towards the original weight of 13 and beyond. The correlation achieved by all systems varies dramatically with $W_2$, demonstrating how correlations achieved by QE systems are highly dependent on the chosen weights.

3 Alternate Human Gold Standard

A recent development in human evaluation of MT is direct assessment ("DA"), a human assessment shown to yield highly replicable segment-level scores, by combination of a minimum of 15 repeat human assessments per translation into mean scores (Graham et al., 2015).

Human adequacy assessments are collected via a 0–100 rating scale that facilitates reliable quality control of crowd-sourcing. Document-level DA scores are computed by repeat assessment of the individual segments within a given document, computation of the mean score for each segment (micro-average), and finally, combination of the mean segment scores into an overall mean document score (macro-average).\textsuperscript{3}

DA assessments are carried out by comparison of a given MT output segment (rendered in black) with a human-generated reference translation (in gray), and human annotators rate the degree to which they agree with the statement: The black text adequately expresses the meaning of the gray text in Spanish.\textsuperscript{3}

Reference translations employed in DA are manually translated by an expert with reference to the entire source document, thus ensuring individual reference segments retain any elements needed to stay faithful to the meaning of the source document as a whole. Since in creation of a test set in general in MT, the professional human translator will have access to and make use of the entire source document, reference translations found in standard MT test sets can directly be employed.

3.1 Self-replication Experiment

Although DA has been shown to produce highly reliable human scores for translations on the segment level, achieving a correlation of above 0.9 between scores for segments collected in separate data collection runs (Graham et al., 2015), the reliability of DA on the document level has yet to be tested. Similar to Graham et al.

\textsuperscript{3}Micro-averaging before macro-averaging avoids weighting segments by the number of times they are assessed.

\textsuperscript{3}Instructions are translated into the target language.

---

\textsuperscript{1}All data employed in this work is available at http://github.com/ygraham/eacl2017

\textsuperscript{2}Instructions are translated into the target language.
Quality Translation 21
D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

Table 1: Numbers of DA human assessments collected per data collection run on Mechanical Turk before (“Total”) and after quality control filtering (“Post QC”) for WMT-16 Document-level QE task (English to Spanish; 62 documents in total).

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Post QC</th>
<th>Mean Assess. per Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run A</td>
<td>14,600</td>
<td>6,640</td>
<td>107</td>
</tr>
<tr>
<td>Run B</td>
<td>10,050</td>
<td>7,700</td>
<td>124</td>
</tr>
</tbody>
</table>

(2015), we therefore assess the reliability of DA for document-level human evaluation by quality-controlled crowd-sourcing in two separate data collection runs (Runs A and B) on Mechanical Turk, and compare scores for individual documents collected in each run.

Quality control is carried out by inclusion of pairs of genuine MT outputs and automatically degraded versions of them (bad references) within 100-translation HITs, before a difference of means significance test is applied to the ratings belonging to a given worker. The resulting p-value is employed as an estimate of the reliability of a given human assessor to accurately distinguish between the quality of translations (Graham et al., 2013; Graham et al., 2014). Table 1 shows numbers of judgments collected in total for each data collection run on Mechanical Turk, including numbers of assessments before and after quality control filtering, where only data belonging to workers with a p-value below 0.05 were retained.

Figure 2 shows the correlation between document-level DA scores collected in Run A with scores produced in Run B, where, for Run B, repeat assessments are down-sampled to show the increasing correspondence between scores as ever-increasing numbers of repeat assessments are collected for a given document. Correlation between scores collected in the two separate data collection runs reaches \( r = 0.901 \) by a minimum of 27 repeat assessments of the sentences of a given document, or by an average 107 sentence assessments per document.

Since DA scores achieve a correlation of \( r > 0.9 \) in our self-replication experiment, we now know that DA provides reliable human evaluation scores for not only segments but also documents. The validity of DA is superior to the existing gold standard employed for document-level QE as it avoids arbitrary weighting or tuning of component scores to reach final gold standard labels. It is therefore highly unlikely to ever unfairly exaggerate (or under-reward) the performance of any QE system in a given evaluation.

With regard to resources required to construct each gold standard, a single DA data collection run cost USD$109 on average, while the cost estimate provided to us by a professional post-editor for the same test set came between USD$1,422 and USD$2,728. In other words, the cost of producing the gold standard is 10–20 times greater for post-editing than DA.\(^5\)

3.2 Re-evaluating Doc-level QE WMT-16

In order to demonstrate DA’s potential as a gold standard, Table 2 shows correlations for WMT-16 document-level QE shared task systems when evaluated with DA and the original gold standard. Results show system rankings that diverge from the original, as the original gold standard exaggerated the performance of three participating sys-

---

\(^3\)Variance in numbers of repeat assessments per document is due to sentences of all documents being sampled without preference for documents made up of larger numbers of sentences.

\(^5\)Post-editing cost estimates are based on 0.06 and 0.12 Euro per source document word converted to USD$. Further details provided by the post-editor in relation to estimates can be found at [https://github.com/ygraham/eacl2017](https://github.com/ygraham/eacl2017)
D3.5: Quality Estimation Metrics and Analysis of 2nd Annot. Round and Error Profiles

Table 2: Correlation ($r$) of system predictions with direct assessment (DA) and original gold standard (WMT-16 QE English to Spanish)

<table>
<thead>
<tr>
<th>System</th>
<th>DA</th>
<th>WMT-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTM-FS+PLS-TREE</td>
<td>0.38</td>
<td>0.36</td>
</tr>
<tr>
<td>GRAPH-DISC</td>
<td>0.32</td>
<td>0.26</td>
</tr>
<tr>
<td>BASE-EMB-GP</td>
<td>0.31</td>
<td>0.39</td>
</tr>
<tr>
<td>BASELINE</td>
<td>0.26</td>
<td>0.29</td>
</tr>
<tr>
<td>RTM-FS-SVR</td>
<td>0.23</td>
<td>0.29</td>
</tr>
</tbody>
</table>

systems, while under-rewarding two other systems. Notably, system GRAPH-DISC, which includes discourse features learned from document-level features, achieves a higher correlation when evaluated with DA compared to the original gold standard.

Differences in correlations are small, however, and can’t be interpreted as differences in performance without significance testing. Differences in dependent correlations showed no significant difference for all pairs of competing systems according to Williams test (Williams, 1959; Graham and Baldwin, 2014).

3.3 Discussion of DA Fluency Omission

In development of the newly proposed variant of DA for document-level QE, the question arose if the assessment should also include an assessment of the fluency of documents (in addition to adequacy), as in Graham et al. (2016b). Besides the several other design criteria in DA aimed at avoiding possible sources of bias in general, the motivation for including a separate fluency assessment was originally to counter any bias resulting from comparison of the MT output with a reference translation in the adequacy assessment, similar to the reference bias encountered in automatic metrics scores. Although genuine human assessors of MT are unlikely to be biased by the reference by anything close to the degree to which automatic metrics scores are, there still exists the possibility that reference bias could impact the accuracy of DA scores to some degree. Inclusion of fluency does of course have a trade-off; however, requiring additional resources, resources that could otherwise be employed to increase the number of translations in the test set, for example. It is important to investigate the degree to which reference bias may or may not be a problem for DA before including it in document-level QE evaluation therefore.

Graham et al. (2016a) provide an investigation into reference bias in monolingual evaluation of MT and despite the risk of reference bias that DA adequacy could potentially encounter, experiment results show no evidence of reference bias. Human assessors of MT appear to genuinely read and compare the meaning of the reference translation and the MT output, as requested with DA, applying their human intelligence to the task in a reliable way, and are not overly influenced by the generic reference.

Although DA fluency could still have its own applications, for the purpose of evaluating MT or MT QE, this additional insight into the lack of reference bias encountered by DA adequacy means that there is no longer any real motivation for including DA fluency when resources are constrained. Given the choice of inclusion of DA fluency in evaluation of document-level QE or expanding the test set (with respect to adequacy), there is no question that the latter is now the more sensible choice.

4 Conclusion

Methodological concerns were raised with respect to optimization of weights employed in construction of document-level QE gold standards in WMT-16. We demonstrated the degree to which MT system rankings are dependent on weights employed in the construction of the gold standard. Experiments showed with respect to the alternate gold standard we propose, direct assessment (DA), scores for documents are highly reliable, achieving a correlation of above 0.9 in a self-replication experiment. Finally, DA resulted in a substantial estimated cost reduction, with the original post-editing gold standard incurring a 10–20 times greater cost than that of DA.

Acknowledgments

This project has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement 645452 (QT21) and Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.
References


