A Comparison of Machine Translation Paradigms for Use in Black-Box Fuzzy-Match Repair

Rebecca Knowles
Department of Computer Science
Johns Hopkins University
rknowles@jhu.edu

John E. Ortega
Dept. de Llenguatges
i Sistemes Informàtics
Universitat d'Alacant
jeo10@alu.ua.es

Philipp Koehn
Department of Computer Science
Johns Hopkins University
phi@jhu.edu

Abstract

Fuzzy-match repair (FMR), which combines a human-generated translation memory (TM) with the flexibility of machine translation (MT), is one way of using MT to augment resources available to translators. We evaluate rule-based, phrase-based, and neural MT systems as black-box sources of bilingual information for FMR. We show that FMR success varies based on both the quality of the MT system and the type of MT system being used.

1 Introduction

Translation memories (TM) play a key role in computer-aided translation (CAT) tools: helping translators to reuse past work (i.e. when translating highly-repetitive texts) by showing them parallel language resources similar to the text at hand (Bowker, 2002). A TM consists of pairs of segments in the source and target language that were produced by past human translation work. In this work, we focus on the *fuzzy-match repair* (FMR)[1] task: automatically modifying target-language TM text before providing it to the human translator, a task similar to automatic post-editing.

Given a new source segment s' to translate, a CAT tool can provide the translator with the best fuzzy-match segment s found in the TM and its corresponding validated translation segment t. The translator can modify mismatched sub-segments[2] of t to produce a correct translation of the new segment s', rather than translating it from scratch. The goal of FMR is to use a source of bilingual information (for example, a dictionary, MT system, phrase table, etc.) to translate the mismatched sub-segments and correctly combine them with the target segment prior to presenting it to the translator. Delivering a correctly repaired segment should save the human translator time, by decreasing the number of changes they need to make in order to complete the translation. A "perfectly" repaired segment would require no changes from the translator.

Ortega et al. (2014) and Ortega et al. (2016) present an algorithm for fuzzy-match repair (FMR) using any source of bilingual information (SBI) as a black-box. Using Apertium (Forcada et al., 2011) as their black-box machine translation (MT) system, they find that the best fuzzy-match repaired segments are closer to the reference translations than either MT or TM alone. We extend that work by comparing three types of MT systems (rule-based, phrase-based, and neural) as the source of bilingual information and by examining the way that both MT system quality and type impact performance.

We begin with a discussion of related work. In Sections 3 and 4, we describe the algorithm used in FMR and the MT systems we tested as sources of bilingual information, respectively. Then, in Section 5 we show that while phrase-based statistical machine translation (henceforth SMT) and neural MT (henceforth NMT) systems both outperform a rule-based (RB) system, these two types of systems perform in markedly different ways as black-box input to the FMR system.

2 Related Work

Attempting to "repair" and propose translations that are closer to the desired translation is a common approach to combining TMs and MT. Simard and Isabelle (2009); He et al. (2010); Koehn and Senellart (2010) all combine TMs and statistical MT in ways that require either a glass-box or explicitly modified MT.

Our work focuses on ways of applying *any* MT system to the task of FMR, without requiring knowledge of the system's inner workings. We use the approach from Ortega et al. (2016) (described in more detail in Section 3). That particular fuzzy-match repair system allows the CAT tool to use any source of bilingual information, but in their publications, they focus only on Apertium (Forcada et al., 2011) as the source of bilingual information. Their work, as well as ours in this paper, depends on an oracle evaluation. In order to be truly useful in a live system, FMR will require some form of quality estimation in order to select the best repaired segment. Research in that area is ongoing.

[1] Or fuzzy-match post-editing (Kranias and Samiotou, 2004). The use of the term "fuzzy-match" references the fuzzy-match score used to find similar source sentences.

[2] Throughout this work, we refer to a complete line of text as a *segment* (rather than a sentence, as a number of the lines of text in the data we use do not constitute full grammatical sentences, but may include things like titles). Sequences of one or more tokens within the segment are *sub-segments*.

In the trade-off between adequacy (translations with the same meaning as the source) and fluency (translations that sound fluid or natural), neural machine translation systems, tend towards greater fluency, while sometimes producing fluent-sounding but semantically inappropriate output (Bojar et al., 2016; Koehn and Knowles, 2017; Toral and Sánchez-Cartagena, 2017). In the FMR application, the full segment from the translation memory may already provide the (fluent) backbone for the translation, while only containing a few subsegment mismatches (such as numbers, names, noun phrases, and so on). This differs from automatic post-editing, where there may be structural issues to repair as a result of errors in the machine translation output. All of this naturally raises the question of how rule-based MT (which may provide greater adequacy for individual subsegments) will compare to neural MT systems (which may provide greater fluency) or phrase-based statistical MT systems (which may fall between the two) for the task of FMR. We also address the question of how NMT systems, which are particularly sensitive to changes in domain or style (Koehn and Knowles, 2017) will perform when used to translate sub-segments rather than full sentences.

Neural MT systems have recently produced state-of-the-art performance across a number of language pairs (Bojar et al., 2017). While NMT has been applied to other CAT applications, namely interactive translation prediction, (Knowles and Koehn, 2016; Wuebker et al., 2016) and neural approaches have been used for automatic post-editing (Pal et al., 2016; Junczys-Dowmunt and Grundkiewicz, 2016; Hokamp, 2017), this is the first work we are aware of that uses NMT for FMR.

3 Black-Box MT for FMR

Here we provide an overview of an algorithm for using black-box MT for FMR. For full details, see Ortega et al. (2016) (Sections 2 and 3), whose algorithm we follow. Black-box approaches allow one system to be used for many tasks, rather than requiring specially-tailored MT systems for every task.

Given a new source-language sentence s' to translate, the FMR system selects (by fuzzy-match score, or FMS) the source-target pair of segments (s, t) from the TM that most closely matches s'. The FMS takes on values between 0% (entire segment requires edits) to 100% (segments identical). A common definition of FMS[3] is given by:

$$\mathrm{FMS}(s, s') = \left(1 - \frac{\mathrm{ED}(s, s')}{\max(|s|, |s'|)}\right) \times 100\% \quad (1)$$

where $\mathrm{ED}(s, s')$ is the (word-based) *edit distance* or Levenshtein distance (Wagner and Fischer, 1974) and $|s|$ and $|s'|$ are the lengths (in tokens) of s and s'. Edit distance is used to find mismatches between s' and s. Sub-segment pairs (σ, σ') containing at least a mismatched word are extracted (via phrase-pair extraction (Koehn, 2009)) from s and s' respectively. The (σ, σ') are passed to the black-box MT system for translation, producing output translations (μ, μ'). To constrain the set which can be used for repairs, any pair (μ, μ') for which μ is not found in t is discarded. The remaining (μ, μ') pairs are then used to "patch" or repair t, by swapping the μ found in t for the new μ' in the hopes of editing t into an accurate translation of s'. More than one such patching action can be applied in the process of forming the final repaired segment, and the system may output multiple unique final repaired segments (using different subsets of the set of available (μ, μ') pairs).

4 Data and Machine Translation Systems

We compare representatives of three MT paradigms: Apertium (rule-based, or RB), Moses (phrase-based SMT) and Nematus (NMT with attention).[4] Test data for the FMR experiments is drawn from the 2015 DGT-TM data set which is composed of highly-repetitive and formal official legal acts and is lowercased in post-processing (Steinberger et al., 2012). We choose English to Spanish as the language pair and translation direction.[5]

4.1 Rule-Based MT (Apertium)

Apertium (Forcada et al., 2011) is a rule-based (RB) machine translation system, which performs translation using a pipeline of components: a morphological analyzer, a part of speech tagger, a lexical transfer module (which uses a bilingual dictionary to translate lexical forms from source language to target), and a structural transfer module (which performs syntactic operations). We use a recent version[6] as a baseline.

4.2 Neural MT (Nematus)

We use the attention-based encoder-decoder Nematus (Sennrich et al., 2017) and the compatible AmuNMT decoder[7] (Junczys-Dowmunt et al., 2016).

Initial model training is done using Europarl v7 (Koehn, 2005) and News Commentary v10 data[8] (WMT13 training data for English–Spanish), with 2012 News Test data for validation. Following the domain adaptation method described in Luong and Manning (2015) and Freitag and Al-Onaizan (2016), we continue training on DGT-TM 2011–2013, with 3000

[3] CAT providers often use proprietary variations of FMS.

[4] Due to limited space, we present the best system trained for each MT type. Other systems trained, which included ones trained on more directly comparable training data, showed the same trends.

[5] In Ortega et al. (2016), Apertium's Spanish–English was the lowest-performing language pair (as compared to Spanish–Portuguese and Spanish–French); we choose it here to demonstrate the range of improvement possible.

[6] http://apertium.org (en–es, SVN rev. 83165)

[7] Now part of Marian (https://github.com/marian-nmt/marian).

[8] http://www.casmacat.eu/corpus/news-commentary.html

lines from the 2014 release as validation data.[9]

We use these training parameters: vocabulary of size 50,000, word embedding layer size of 500, hidden layer size of 1000, batch size of 80, Adadelta (Zeiler, 2012) as the optimizer, maximum sentence length of 50, and default learning rate of 0.0001. All other parameters are set to Nematus defaults. Data is preproccessed with the standard preprocessing scripts: tokenization, truecasing, and byte pair encoding (Sennrich et al., 2016). We report scores with a beam size of 12.

4.3 Phrase-Based SMT (Moses)

We use Moses (Koehn et al., 2007) to train our phrase-based statistical MT (SMT) system using the same parallel text as the NMT model, with the addition of Common Crawl,[10] for phrase extraction. Europarl v7, News Commentary v10, monolingual News Crawl from 2007–2011, Spanish Gigaword v3 (Mendonça et al., 2011), and target side DGT-TM data were used to build a 5-gram interpolated language model.

We use an operation sequence model (Durrani et al., 2015) with order 5, Good-Turing discounting of phrase translation probabilities, binning of phrase pair counts, pruning of low-probability phrase pairs, and sparse features for target word insertion, deletion, and translation, and phrase length. Tuning is run on the same DGT-TM data used for NMT model validation.

5 Experiments and Results

5.1 MT System Quality

We first compare the MT systems in terms of both BLEU score and word error rate (WER)[11] on the task of translating the full segments from the 1993 segments of the 2015 DGT-TM test set used for evaluating FMR.[12] Results are shown in the right two columns of Table 1, under the heading “MT Output”. Both the SMT and NMT systems report higher BLEU scores and lower WER than the RB system. The best performing system by these metrics is the SMT system, with a BLEU score of 57.2 and a WER of 35.2.

5.2 Oracle Fuzzy-Match Repair Results

At times the FMR system fails to repair a segment (e.g. if no set of sub-segment translations match the target-side TM segment) and at others it produces multiple patched segments. To handle the latter, we use the oracle evaluation approach from Ortega et al. (2016), which, given a fuzzy-match score threshold θ (we use 60%, 70%, and 80% as values of θ), consists of:

1. For each segment s' in the test set, find the best segment pair (s, t) from the translation memory such that $\text{FMS}(s', s) \geq \theta$, if such a pair exists.[13]
2. If there exists such a pair (s, t), produce all possible FMR segments using that pair. Select the repaired segment with the lowest edit distance to the reference t' (oracle evaluation). If no repaired segment was produced through the FMR process (or no satisfactory pair (s, t) was found), produce a translation of s' using the MT system.

This would not be possible in a real use setting, as it requires access to the reference translation to determine which repaired segment has the lowest WER (with respect to the reference). Thus the oracle results represent the most optimistic case for fuzzy-match repair (the case where we can always select the optimal repaired segment when more than one is produced) possible within this fixed framework; quality estimation and ranking of hypotheses for a more real-world setting has been left for future work by Ortega et al. (2016). The challenge of combining several such CAT options is far from trivial (Forcada and Sánchez-Martínez, 2015); for example, we found that for high-quality MT systems, MT output can (under certain FMS thresholds) outperform the best FMR output upwards of 15% of the time.

Table 2 shows example segments: source and reference (s', t'), the best fuzzy-match from the TM (s, t), and the best output from the three MT systems. In this example, the SMT system produces the best repair, with a WER of 25.0% (as compared to the TM WER of 37.5%). The SMT system successfully inserts the desired translation (*formación*) of the mismatched word *training*, replacing *desarrollo*, but fails to add the token *los*, and doesn’t change the translation of *promote*. This latter error is to be expected, since *promote* is a matching word across the source and TM source, so the system does not try to repair it.

Table 1 reports word error rate[14] over several subsets of the test set. In the *Match* columns, the score is computed based on a subset of the full data: for each fuzzy-match threshold θ (60%, 70%, and 80%) we select the segments for which a fuzzy-match could be found in the TM (such that fuzzy-match score $\geq \theta$%), and apply FMR (in the event that FMR does not successfully produce a repair, we instead back off to the unmodified

[9] As the fuzzy-match repair scenario assumes that no sentences from that test set have been observed in the TM, we remove exact test set matches from DGT-TM training data.

[10] Available at http://www.statmt.org/wmt13/translation-task.html

[11] Computed over the full corpus as $\frac{\sum_i ED(t_i,r_i)}{\sum_i |r_i|}$, where ED is the Levenshtein edit distance and r_i is the i^{th} reference in the corpus.

[12] The initial set consisted of 2000 segments, of which 7 were discarded for being longer than 100 tokens.

[13] Note that we use the fuzzy-match score solely on the source side. Esplà-Gomis et al. (2015) propose using an additional threshold of $|FMS(s', s) - FMS(t', t)| < \phi$ to lessen the incidence of correct repairs being marked as incorrect due to inconsistencies resulting from free translations (e.g. two different but equally appropriate translations of the same phrase appearing in s and s', respectively).

[14] The WER is again computed at the document level, as before, over the particular set of sentences as defined by the column of the table.

	60% FMT		**70% FMT**		**80% FMT**		**MT Output**	
Sys.	*Match*	*Full*	*Match*	*Full*	*Match*	*Full*	*WER, Full*	*BLEU, Full*
TM	20.8	-	16.7	-	13.4	-	-	-
RB	18.5	37.5	15.0	39.6	12.2	43.7	60.8	19.2
SMT	15.6	26.7	12.7	27.3	10.4	27.9	35.2	57.2
NMT	15.2	27.1	12.0	26.8	9.4	28.5	36.8	52.6

Table 1: The left section of the table contains word error rates for fuzzy-match repair. In the *Match* columns, the score is computed based on a subset of the full data: 60% Fuzzy-Match Threshold (1184 segments for which a fuzzy-match could be found in the TM with fuzzy-match score $\geq$ 60%), 70% Fuzzy-Match Threshold (828 segments), and 80% Fuzzy-Match Threshold (660 segments), with the oracle best fuzzy-match repaired segment scored, backing off to the TM if no repair was successful. In the *Full* column, the data from the corresponding *Match* column backs off to MT output when no TM segment with a sufficiently high FMS is available.The rightmost sections (MT Output) contain BLEU scores and WER for machine translation output of the full data set.

s':src	promote human resources training;
t':ref	promover la formación de los recursos humanos;
s:TM	promote human resources development;
t:TM	fomentar el desarrollo de los recursos humanos;
RB	fomentar el desarrollo de los los recursos humanos que entrenan;
SMT	fomentar la formación de recursos humanos;
NMT	fomentar los recursos humanos;

Table 2: Example segments, showing the best fuzzy-match repaired segments for three MT systems.

TM segment). In the *Full* column, the data from the corresponding *Match* column backs off to MT output when no TM segment with a sufficiently high FMS is available. The *WER, Full* column under the MT Output heading in Table 1 can be compared directly to any of the *Full* columns. We see that FMR with either SMT or NMT outperforms all pure MT utput (across all three system types). The worst FMR performance between those two systems is the NMT at the 80% fuzzy-match threshold with a WER of 28.5 on *Full* data, yet this still outperforms even the best MT output with its WER of 35.2. This underscores the potential usefulness of FMR.

Interestingly, despite having worse BLEU scores and WER on full-sentence translations, the NMT system actually outperformed the SMT system as a source of bilingual information for FMR on the subsets of data for which TM matches were found. The better full-data performance of the SMT system can be attributed to backing off to (better) MT output when no TM best-match was available. All of the MT systems outperform the no-repair TM baseline WER (in which we simply computed WER for the best fuzzy-matches from the TM, without any repairs).

5.3 Analysis

The NMT system performs best for FMR on matches and it also is more often successful at repairing segments. This raises two questions: Are the improvements solely or primarily due to successfully repairing more sentences? (Section 5.3.1) Why do the neural systems succeed in repairing more sentences? (Section 5.3.2) We focus on comparing SMT and NMT, due to their stronger performance over the RB system.

5.3.1 Direct Comparison

At the 60% FMT level, the SMT system successfully produced repairs for 788 segments, while the NMT system successfully produced repairs for 957 segments (out of a possible 1184 segments).[15] Since those are two distinct sets of segments, we cannot directly compare WER. We first examine the intersection of those sets (the subset of segments for which both systems successfully performed FMR).

A total of 754 segments were successfully repaired by both systems. There were 34 segments which the SMT system repaired and the NMT system did not, and 203 segments for which the opposite was true. Of the 754 segments repaired by both, 212 were repaired better by the NMT system, 139 were repaired better by the SMT system, and 403 were repaired equally well by the two MT systems (all in terms of WER). Computing the WER over this shared 754 segment set, we find that the WER of the SMT system (14.4%) is quite close to that of the NMT system (14.3%). This suggests that the NMT system's ability to patch more sentences plays a major role in its better FMR results.

The NMT system produced an average of 1.92 possible repaired segments per source segment (standard deviation: 1.29, maximum: 9). Using the SMT system, an average of 1.68 possible repaired segments were produced per source segment (standard deviation: 0.92, maximum: 7). In a real-world setting, the system would need to choose between more repaired options for the NMT system than the SMT system.

To see how important it is to select the best repaired segment, we compare the optimistic oracle approach to a pessimistic one, where for each of the 754 segments, we select the repaired segment with the highest WER (the *worst* possible outcome). For this set of segments, the TM baseline WER is 20.6%. When we choose the

[15] Professional translators typically use higher fuzzy-match thresholds, but we select 60% in this section to provide the greatest amount of data for direct comparison of repairs.

worst repaired segments produced by the NMT system, the WER is 20.5%, which is very close to the TM baseline. The WER for the SMT system appears slightly better, at 19.0%. Both represent a large drop from the optimistic oracle, but the drop is greater for the NMT system.

5.3.2 Analysis of Sub-Segment Translations

We examine the sub-segment translations produced by the NMT and SMT systems to gain insight about what allows that NMT system to repair more segments and produce more possible repaired versions per segment.

Without gold references for the sub-segment translations, we cannot evaluate them in terms of WER or BLEU, so we examine them quantitatively and qualitatively. First, we look at the lengths of the translations of the sub-segments. For both the SMT and NMT systems, the translations tend to be longer than the source sub-segments (64% of the time for the SMT system and 58% of the time for the NMT system). The NMT system produces translations that are shorter than the source 23% of the time, while the SMT system does so 18% of the time. They also differ in the range of lengths; the NMT system has more extreme values, sometimes producing no translation at all and even occasionally producing translations more than three times the length of the longest source segments. On average, the SMT translations are 2.37 tokens longer than the source sub-segments (SD.: 3.95). The NMT translations average 2.71 tokens longer than the source, with a much greater standard deviation of 10.26. The very long NMT translations may be more likely to be discarded (due to not matching), but the very short translations may be easier to find matches for in the TM target side, contributing to the larger number of sentences the NMT system patches.

We also note a qualitative difference: the SMT systems often add additional punctuation that was not included in the source, as well as determiners. These spurious tokens could make it harder to find matches in the TM target segments, resulting in fewer opportunities for fuzzy-match repair. This could be caused by the language model providing higher scores to the phrases that include those tokens.

5.4 Discussion

The sub-segments which need to be translated for fuzzy-match repair are not complete always segments, but often sub-segments which could be taken from any point in the original segment. Each sub-segment is then translated using the MT system, without full context (though Ortega et al. (2014) do note that the context provided by using "anchored" subsegments—those that have overlap with the matching subsegments—improves performance over non-anchored subsegments).[16] This poses a potential challenge for any MT system which is trained on full segments. In the case of the SMT system, the language model may prefer sub-segment translations that include, for example, determiners or additional punctuation, as we observed. NMT systems have been observed to do a poor job of handling data that differs from the original training data, often producing fluent-seeming text that has little to do with the source. While this mismatch does not seem to have had a strong negative impact on the overall results, it is possible that the results could still improve if the sub-segmental input were better matched to the training data. There would be several ways to do this. The first would be to produce parallel sub-segment data (using phrase alignments) and use this instead of the full sentences for domain adaptation. Another alternative (though it would require changes to the MT system, violating the goal of a black-box system) would be to always provide the MT system with access to the full context surrounding or preceding the segment to be translated, which it could use as a better starting state to generate the segment's translation.

6 Conclusions

We show that three very different types of machine translation can successfully be used in the black-box fuzzy-match repair approach described in Ortega et al. (2016). We find that despite lower BLEU scores on full-sentence translations, in the oracle evaluation, NMT systems outperform phrase-based SMT systems as sources of bilingual information for fuzzy-match repair (potentially surprising, given that the task requires translation of sub-segments). However, the greater variance in NMT results suggests a need for caution when deciding what type of MT system to use as a black-box, and underscores the need for work on quality estimation for real-world use in CAT tools.

Acknowledgments

This work was partially supported by a National Science Foundation Graduate Research Fellowship under Grant No. DGE-1232825 (to the first author) and by the Spanish government through the EFFORTUNE (TIN2015-69632-R) project (the second author). We thank the reviewers for their comments and suggestions, and Mikel Forcada and Felipe Martínez for useful comments on an early version of this work.

[16] We ran a brief set of experiments on the XML markup method described in Koehn and Senellart (2010), for which we omit detail due to space constraints. We found that on the sentences whose TM best-matches met or exceeded the 60% threshold, the XML method improved slightly over the TM baseline. This is in contrast to what Koehn and Senellart (2010) observed in their original work (namely, that the XML method only improved over the TM and MT output in terms of BLEU score for higher fuzzy-match thresholds).

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Lynne Bowker. 2002. *Computer-Aided Translation Technology: A Practical Introduction*. University of Ottawa Press.

Nadir Durrani, Hassan Sajjad, Shafiq Joty, Ahmed Abdelali, and Stephan Vogel. 2015. Using joint models for domain adaptation in statistical machine translation. *Proceedings of MT Summit XV*, page 117.

Miquel Esplà-Gomis, Felipe Sánchez-Martínez, and Mikel L. Forcada. 2015. Using machine translation to provide target-language edit hints in computer aided translation based on translation memories. *J. Artif. Int. Res.*, 53(1):169–222.

Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Mikel L Forcada and Felipe Sánchez-Martínez. 2015. A general framework for minimizing translation effort: towards a principled combination of translation technologies in computer-aided translation. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, pages 27–34.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Y. He, Y. Ma, J. van Genabith, and A. Way. 2010. Bridging SMT and TM with translation recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630, Uppsala, Sweden.

Chris Hokamp. 2017. Ensembling factored neural machine translation models for automatic post-editing and quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 647–654.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? A case study on 30 translation directions. *CoRR*, abs/1610.01108.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn and Jean Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

Lambros Kranias and Anna Samiotou. 2004. Automatic translation memory fuzzy match post-editing: A step beyond traditional TM/MT integration. In *LREC*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*.

Ângelo Mendonça, Daniel Jaquette, David Graff, and Denise DiPersio. 2011. Spanish gigaword third edition ldc2011t12. Web Download. Philadelphia: Linguistic Data Consortium.

John E Ortega, Felipe Sánchez-Martınez, and Mikel L Forcada. 2014. Using any machine translation source for fuzzy-match repair in a computer-aided translation setting. In *Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2014*, volume 1, pages 42–53.

John E. Ortega, Felipe Sánchez-Martínez, and Mikel L. Forcada. 2016. Fuzzy-match repair using blackbox machine translation systems: what can be expected? In *Proceedings of the 12th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2016, vol. 1: MT Researchers' Track)*, pages 27–39, Austin, TX, USA.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016. A neural network based approach to automatic post-editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel L"aubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a Toolkit for Neural Machine Translation. In *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Michel Simard and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. *Proceeding of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 120–127.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2012. DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, Istanbul.

Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1063–1073, Valencia, Spain. Association for Computational Linguistics.

Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

Joern Wuebker, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and inference for prefix-constrained machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Berlin, Germany. Association for Computational Linguistics.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.