Highlighting Matched and Mismatched Segments in Translation Memory Output through Sub-Tree Alignment

Ventsislav Zhechev

EuroMatrixPlus, School of Computing, Dublin City University, Ireland

Abstract

In recent years, it is becoming more and more clear that the localisation industry does not have the necessary manpower to satisfy the increasing demand for high-quality translation. This has fuelled the search new and existing technologies that would increase translator throughput. As Translation Memory (TM) systems are the most commonly employed tool by translators, a number of enhancements are available to assist them in their job. One such enhancement would be to show the translator which parts of the sentence that needs to be translated match which parts of the fuzzy match suggested by the TM. For this information to be used, however, the translators have to carry it over to the TM translation themselves.

In this paper, we present a novel methodology that can automatically detect and highlight the segments that need to be modified in a TM-suggested translation. We base it on state-of-the-art sub-tree alignment technology (Zhechev, 2010) that can produce aligned phrase-based-tree pairs from unannotated data. Our system operates in a three-step process. First, the fuzzy match selected by the TM and its translation are aligned. This lets us know which segments of the source-language sentence correspond to which segments in its translation. In the second step, the fuzzy match is aligned to the input sentence that is currently being translated. This tells us which parts of the input sentence are available in the fuzzy match and which still need to be translated. In the third step, the fuzzy match is used as an intermediary, through which the alignments between the input sentence and the TM translation are established. In this way, we can detect with precision the segments in the suggested translation that the translator needs to edit and highlight them appropriately to set them apart from the segments that are already good translations for parts of the input sentence. Additionally, we can show the alignments—as detected by our system—between the input and the translation, which will make it even easier for the translator to post-edit the TM suggestion. This alignment information can additionally be used to pre-translate the mismatched segments, further reducing the post-editing load.

1. Introduction

As the world becomes increasingly interconnected, ideas, products and services need to be communicated to the widest audience possible. This requires localisation for as many languages, cultures and locales as possible, with translation being one of the main parts of the localisation process. Because of this, the amount of data that needs professional high-quality translation is continuing to increase well beyond the capacity of the world's human translators.

Current efforts in the localisation industry are mostly directed at the reduction of the amount of data that needs to be translated manually from scratch. Such efforts mainly include the use of Translation Memory (TM) systems, where earlier translations are stored in a database and offered as suggestions when new data needs to be translated. As TM systems were originally limited to providing transla-

techniques is often seen as the only feasible development that has the potential to significantly reduce the amount of manual translation.

The system that we present in this paper, however, takes a different approach in that it aims to aid the translators in the process of post-editing TM matches. In particular, the system isolates and marks-up the parts of the TM-suggested translation (henceforth *the TM output*) that can be judged as good based on automatic alignment between the segment that needs to be translated (henceforth *the input*) and the TM fuzzy match. The parts of the TM output that are leftover after this process are the ones that need editing and the system highlights them so that they can be easily spotted by the translator without having to search through the whole TM output. The system can further be augmented with a Statistical Machine Translation (SMT) backend to pre-translate the mismatched parts of the TM output, before presenting them to the translator for post-editing, thus hopefully reducing post-editing effort.

With recent advances in the performance and quality of Statistical Machine Translation (SMT) systems, many commercial TM systems offer the user the option to obtain SMT-generated translations for new data.[1] Such translations, however, are usually only obtained for cases where the TM system could not produce a good-enough translation (cf. Heyn, 1996). Given that the SMT system used is presented with the "hard" translation cases (strings not seen in the TM) and is usually trained only on the data available in the TM, it tends to have only few examples from which to construct the translation, thus often producing fairly low quality output. Because of this, and since translators are used to TMs as an integral part of their working environment but less so to MT, SMT output is still often scorned upon by professional translators. We hope that the system described here presents a use case for SMT in a TM context in which translators may see the benefits this technology can bring.

In Section 2, we present the technical details of the design of our system, together with motivation for the particular design choices we took. Section 3 details the experiments we performed with pre-translating the mismatched segments of the TM output and the results we achieved. In Section 4, we present out future development plans and conclude.

2. System Framework

We present a system that uses state-of-the-art sub-tree alignment techniques to mark-up Translation Memory output highlighting the parts of it that need to be edited manually.

[1] At the time of writing, the only commercial system to include technology similar to the framework described in this paper, rather than simply allow the translation of full sentences via an SMT plugin, was the upcoming Déjà Vu X2 (`http://www.atril.com`).

2.1. Translation Memory Backend

Although the intention is to integrate the methodology outlined here into a full-scale TM system, to have complete control over the process for this initial research we decided to build a simple prototype TM backend ourselves.

We employ a database setup using the PostgreSQL v.8.4.3[2] relational database management (RDBM) system. The segment pairs from a given TM are stored in this database and assigned unique IDs for further reference. When a new sentence is supplied for translation, the database is searched for (near) matches, using a Fuzzy Match Score (FMS) based on character-based Levenshtein edit distance (Levenshtein, 1965). To speedup the computation, we use a recursive wrapper around the PostgreSQL-internal implementation of the `levenstein()` function, taken from the TinyTM project.[3]

In this way, for each input segment, from the database we obtain the matching segment with the highest FMS, its translation and the score itself.

2.2. Sub-Tree Alignment

The system presented in this paper uses sub-tree alignment (Zhechev, 2010) to discover parts of the input sentence that correspond to parts of the suggested translation extracted from the TM database. This is done in a three-step process. First, the plain TM match and the TM output are aligned, which produces a sub-tree aligned phrase-based tree pair. We call this step bilingual alignment.

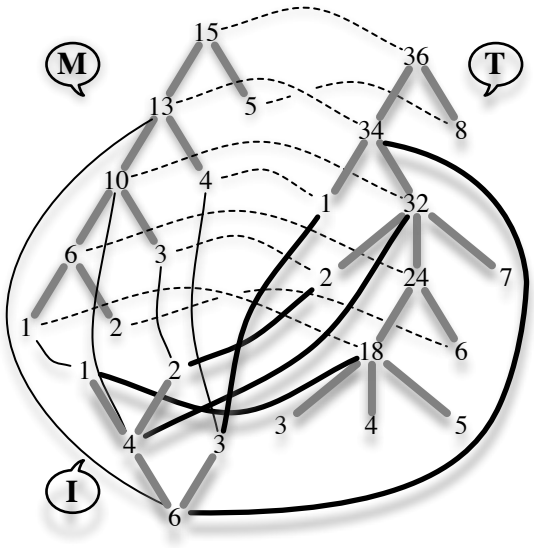
In the second step, called monolingual alignment, the phrase-based tree-annotated version of the TM match is aligned to the plain-text input sentence. The reuse of the structure for the TM match allows us to use it in the third step as an intermediary to establish the available sub-tree alignments between the input sentence and the TM output.

During this final alignment, we identify matched and mismatched portions of the input sentence and their possible translations in the TM output and, thus, this step is called matching.

The alignment process is exemplified in Figure 1. The tree marked '**I**' corresponds to the input sentence, the one marked '**M**' to the TM match and the one marked '**T**' to the TM output. We only display the node ID numbers of the non-terminal nodes in the phrase-structure trees—in reality all nodes carry the label 'X'. These IDs are used to identify the sub-sentential alignment links. The lexical items corresponding to the leaves of the trees are presented in the table below the graph.

[2] `http://www.postgresql.org`

[3] `http://tinytm.sourceforge.net/en/technology/fuzzymatch.html`



I	1	2	3					
input	sender	email	address					
M	1	2	3	4	5			
match	sender	's	email	address	.			
T	1	2	3	4	5	6	7	8
translation	adresse	électronique	de	l'	expéditeur	du	message	.

Figure 1. Example of sub-tree alignment between an input sentence, TM match and TM output

The alignment process can be visually represented as starting at a linked node in the **I** tree and following the link to the **M** tree. Then, if available, we follow the link to the **T** tree and this leads us to the **T**-tree node corresponding to the **I**-tree node we started from. In Figure 1, this results in the **I–T** alignments **I1–T18**, **I2–T2**, **I3–T1**, **I4–T32** and **I6–T34**. The first three links are matches, because the lexical items covered by the **I** nodes correspond exactly to the lexical items covered by their **M** node counterparts. Such alignments provide us the direct TM translations for our input. The last two links in the group are mismatched, because there is no lexical correspondence between the **I** and **M** nodes (node **I4** corresponds to the phrase *sender email*, while the linked node **M10** corresponds to *sender 's email*). Such alignments can only be used to infer reordering information for the experiments presented in Section 3. In particular in this case, we can infer that the target word order for the input sentence is *address email sender*, which corresponds to the translation *adresse électronique de l' expéditeur*.

The correspondence between the input and the TM output could be presented to the user as shown in Figure 2.[4] The alignments representing correct translations as inferred from the TM are highlighted in green and alignment lines are drawn to

[4] A prototype system implementing the proposed techniques is currently still being developed. All UI elements presented here may change in the final product.

the translator what the system believes the translational correspondences are. The final words in the French segment are faded out, as an indication from the system that they should probably be deleted. If desired, the system can also present the TM match as the intermediary between the input and the TM output.

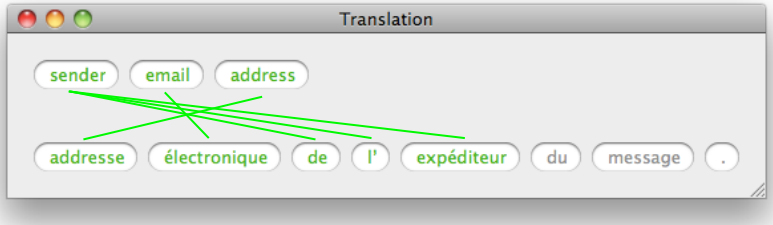


Figure 2. Suggested UI representation for the correspondence in Figure 1.

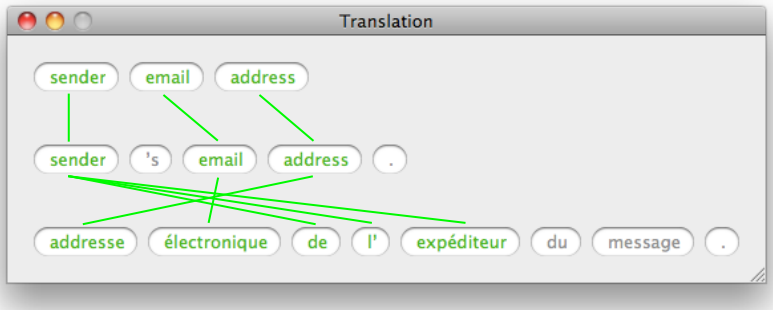


Figure 3. Suggested UI representation for the correspondence in Figure 1. (including the TM match)

We decided to use sub-tree-based alignment, rather than plain word alignment (e.g. GIZA++ – Och and Ney, 2003), due to a number of factors. The goal of sub-tree alignment methods is not to align as many lexical items as possible, but to represent structurally the best translational equivalences in the sentences that are being aligned. This allows for the encoding of long-distance translational dependency by means of links between nodes higher up in the tree structures.

The alignments produced by a sub-tree alignment model are also precision-oriented, rather than recall-oriented (cf. Tinsley, 2010). This is important in our case, where we want to only extract those parts of the translation suggested by the TM for which we are most certain that they are good translations.

Out of the three currently available open-source sub-tree alignment systems, two can only operate when at least one language-side of the data that needs to be aligned is pre-parsed (Ambati et al., 2009, Tiedemann, 2010) and one of them needs a hand-crafted parallel treebank as training data (Tiedemann, 2010).

As these requirements necessitate the acquisition of human-annotated data besides the data available in the TM, we decided to use the system described in (Zhechev, 2010) instead. It can produce aligned phrase-based-tree pairs from unannotated (i.e. unparsed) data. It can also function fully automatically without the need for any training data.

The only resource necessary for the operation of this system is a probabilistic bilingual dictionary covering the data that needs to be aligned. For the bilingual alignment step, such a bilingual dictionary—if not already available—can be generated automatically using a tool like GIZA++ (Och and Ney, 2003). For the monolingual alignment step, the required probabilistic dictionary is generated by simply listing each unique token seen in the source-language data in the TM as translating only as itself with probability 1.

3. Experimenting with Pre-Translation

As stated earlier, the design of our system allows for the pre-translation of the mismatched parts of the TM output using an SMT system. We explore two approaches to handling the translation of these outstanding fragments.

The first approach is extremely straightforward, in that the non-translated segments of the input sentence are sent severally to the SMT backend for translation without any context information. The segments translated using TM data and the ones translated using the SMT backend are then simply concatenated in the target-language word order, as determined implicitly by the sub-tree alignment information. The most serious drawback of this approach is that translating the individual segments out of context might often lead to improper lexical choice by the SMT backend, which could have been properly resolved given the context of the whole input sentence. Also, for certain cases (particularly with low FMS) the target-language word order may not be discernible for all input-sentence segments and the translations of the segments with undetermined placement are simply appended to the end of the generated translation. Still, the simplicity of this approach makes it a good baseline benchmark against which to evaluate improvements. This approach is referred to as *comb* below.

The second approach to handling non-translated input-sentence segments relies on a specific feature of the SMT backend we use, namely the Moses system (Koehn et al., 2007). We decided to use this particular system as it is the most widely adopted open-source SMT system, both for academic and commercial purposes. In this approach, we annotate the segments of the input sentence for which translations have been found from the TM suggestion using XML tags with the translation corresponding to each segment given as an attribute to the encapsulating XML tag. The SMT backend is supplied with a string consisting of the concatenation of the XML-enclosed translated segments and the plain non-translated segments in the target-language word order, as established by the alignment process. The SMT backend is instructed to translate the string as a whole, while keeping the translations supplied via the XML annotation. This mode of operation provides the SMT backend with the necessary context information to come up with proper lexical choice for the non-translated fragments and allows it to introduce reordering on its own, based on the SMT reordering models derived during training. We refer to this approach as *xml*. A drawback present with this approach is that we need to perform additional alignment and matching steps to reestablish the alignments between the input and the newly generated translation.

Further, we present experiments on pre-translating the mismatched parts of the TM output using the methods described above (*comb* and *xml*).

3.1. Experimental Data

We use real-life TM data provided by Symantec Ireland, an industrial partner of CNGL. The TM was generated during the translation of RTF-formatted customer support documentation. The data is in TMX format and originally contains 108 967 English–French translation segments, out of which 14 segments either have an empty language side or have an extreme discrepancy in the number of tokens for each language side and were therefore discarded.

A particular real-life trait of the data is the presence of a large number of XML tags. Running the tag-mapping tool described in Section 2.5, we gathered 2 049 distinct tags for the English side of the data and 2 653 for the French side. Still, there were certain XML tags that included a label argument whose value was translated from one language to the other. These XML tags were left intact so that our system could handle the translation correctly.

The TM data also contain a large number of file paths, e-mail addresses, URLs and others, which makes bespoke tokenisation of the data necessary. Our tokenisation tool ensures that none of these elements are tokenised, keeps RTF formatting sequences non-tokenised and properly handles non-masked XML tags, minimising their fragmentation.

Due to the nature of TM data, translation segments rarely occur more than once in the data set. This explains the high number of unique tokens (measured after pre-processing) that we observe for the two languages — 41 379 for English and 49 971 for French—out of 108 953 segment pairs. The average sentence length is 13.2 for English and 15.0 for French.

For evaluation, we use a data set of 4 977 English–French segments that were obtained from a different set of documents than the ones, for whose translation the TM presented above was used. The sentences in the test set—with average length 9.2 tokens for English and 10.9 for French — are significantly shorter compared to the TM.

It must be noted that we used SMT models with maximum phrase length of 3 tokens, rather than the standard 5 tokens, and for decoding we used a 3-gram language model. This results in much smaller models than the ones usually used in mainstream SMT applications, thus making the system more accessible by lowering the system requirements for running it. (The standard for some tools goes as far as 7-token phase-length limit and 7-gram language models.)

3.2. Evaluation Results

For the evaluation of our system, we used a number of widely accepted automatic MT-quality metrics, namely BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2006) and inverse F-Score based on token-level precision and recall.

We setup our system to only fully process input sentences for which a TM match with an FMS over 50% was found, although all sentences were translated directly using the SMT backend for control purposes (marked as *direct*). The TM output was also evaluated unmodified (*tm*). *comb* and *xml* refer to the two setups described in the beginning of Section 3.

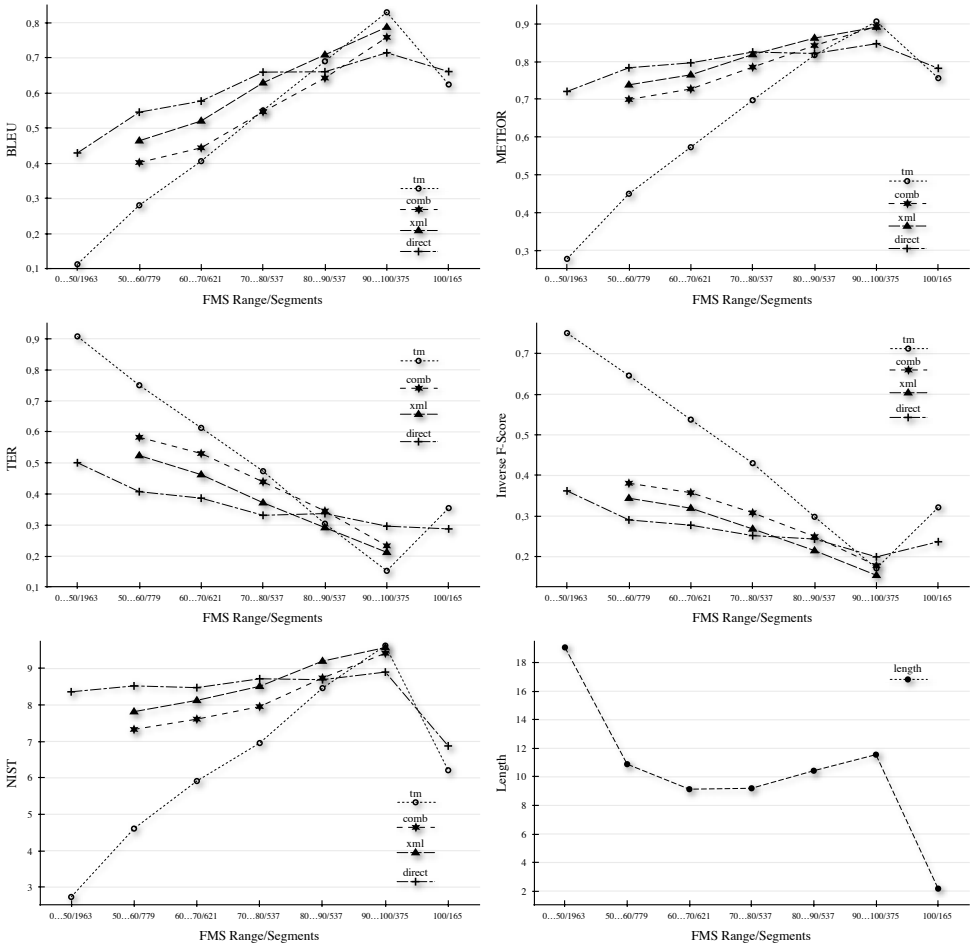


Figure 4. Evaluation results for English-to-French translation, by FMS range

The results of the evaluation are given in Figure 4, where the *tm* and *direct* scores are also given for FMS between 0% and 50% and FMS 100%. Across all metrics we see a uniform drop in the quality of TM-suggested translations, which is what we expected, given that these translations contain one or more incorrect words. We believe that the relatively high scores recorded for the TM-suggested translations at the high end of the FMS scale are a result of the otherwise perfect word order and lexical choice. For n-gram-match-based metrics like the ones we used, such a result is expected and predictable. Although the inverse F-score results show the potential of our setup to translate the outstanding tokens in a 90%–100% TM match, it appears that the SMT system produces word order that does not correspond to the reference translation and because of this receives lower scores on the

The inverse F-score results also confirm our prediction that the *comb* translation approach is prone to lexical-choice errors due to the lack of context during translation. These errors seem to be the major factor leading to significantly worse performance compared to the *xml* approach.

The unexpected drop in scores for perfect TM matches is due to discrepancies between the reference translations in our test set and the translations stored in the TM. We believe that this issue affects all FMS ranges, albeit to a lower extent for non-perfect matches. Unfortunately, the exact impact cannot be ascertained without human evaluation.

We observe a significant drop-off in translation quality for the direct output below FMS 50%. This suggests that sentences with such low FMS should be translated either by a human translator from scratch, or by an SMT system trained on different/more data.

The *xml* setup of our system clearly outperforms the *direct* SMT translation for FMS between 80% and 100% and has comparable performance between FMS 70% and 80%. Below FMS 70%, the SMT backend has the best performance. Although these results are positive, we still need to investigate why our system has poor performance at lower FMS ranges. Theoretically, it should outperform the SMT backend across all ranges, as its output is generated by supplying the SMT backend with good pre-translated fragments. The Inverse F-Score graph suggests that this is due to worse lexical choice, but only manual evaluation can provide us with clues for solving the issue.

The discrepancy between the results in the Inverse F-Score graph and the other metrics suggests that the biggest problem for our pre-translation system is producing output in the expected word-order.

4. Future Work and Conclusions

First, our main goal is to integrate the presented methodology in a standalone commercial or open-source TM system so that it can become a part of a fully integrated localisation workflow.

The pre-translation functionality needs to first be evaluated on a small, but representative, set of data to establish the FMS level at which the system performs at its best and set the appropriate thresholds accordingly for the further use of the system. This can be linked to a translation-quality estimator, when such tools become more widely available.

Finally, a user study evaluating the effect of the use of our system on post-editing speeds should be performed. We expect the findings of such a study to show a significant increase of throughput that will significantly reduce the costs of translation for large-scale projects.

The system we developed uses precise sub-tree-based alignments to reliably determine correspondences between an input sentence and a TM output and presents them to a translator to facilitate the post-editing process. It can employ an SMT backend to translate the mismatched parts of the input sentence and produce a complete translation with higher quality than the original TM output.

Our evaluation of the pre-translation functionality shows that it significantly improves the quality of the pure SMT output when using TM matches with FMS above 80% and produces results on par with the pure SMT output for FMS between 70% and 80%. Still, further investigation is needed to properly diagnose the drop in quality for FMS below 70%.

Acknowledgements

This research is funded under FP7 of the EC within the EuroMatrix+ project (grant №231720). The data used was generously provided by Symantec Ireland.

References

Ambati, Vamshi, Alon Lavie and Jaime Carbonell. 2009. Extraction of Syntactic Translation Models from Parallel Data using Syntax from Source and Target Languages. In *The Twelfth Machine Translation Summit (MT Summit XII)*. Ottawa, ON, Canada.

Banerjee, Satanjeev and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgements. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pp. 65–72. Ann Arbor, MI.

Doddington, George. 2002. Automatic Evaluation of Machine Translation Quality Using N-Gram Co-Occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research (HLT '02)*, ed. Mitchell Marcus, pp. 138–145. San Diego, CA: DARPA.

Heyn, Matthias. 1996. Integrating Machine Translation into Translation Memory Systems. In *Proceedings of the EAMT Machine Translation Workshop, TKE '96*, pp. 113–126. Vienna, Austria.

Koehn, Philipp, 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 Demo and Poster Sessions of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pp. 177–180. Prague, Czech Republic.

Levenshtein, Vladimir I. 1965. Двоичные коды с исправлением выпадений, вставок и замещений символов (Binary Codes Capable of Correcting Deletions, Insertions, and Reversals). *Доклады Академий Наук СССР*, 163 (4): 845–848. [reprinted in: Soviet Physics Doklady, 10: 707–710.].

Och, Franz Josef and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29 (1): 19–51.

Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (ACL '02)*, pp. 311–318. Philadelphia, PA.

Snover, Matthew, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA '06)*, pp. 223–231. Cambridge, MA.

Tiedemann, Jörg. 2010. Lingua-Align: An Experimental Toolbox for Automatic Tree-to-Tree Alignment. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC '10)*. Valletta, Malta.

Tinsley, John. 2010. Resourcing Machine Translation with Parallel Treebanks. School of Computing, Dublin City Univercity: PhD Thesis. Dublin, Ireland.

Zhechev, Ventsislav. 2010. Automatic Generation of Parallel Treebanks. An Efficient Unsupervised System: Lambert Academic Publishing.