Neural Machine Translation with Inflected Lexicon

Nowakowski Artur artur.nowakowski@amu.edu.pl
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznan, 61-614, Poland

Jassem Krzysztof jassem@amu.edu.pl
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznan, 61-614, Poland

Abstract

The paper presents experiments in Neural Machine Translation with lexical constraints into a morphologically rich language. In particular, we introduce a method, based on constrained decoding, which handles the inflected forms of lexical entries and does not require any modification to the training data or model architecture. To evaluate its effectiveness, we carry out experiments in two different scenarios: general and domain-specific. We compare our method with baseline translation, i.e. translation without lexical constraints, in terms of translation speed and translation quality. To evaluate how well the method handles the constraints, we propose new evaluation metrics which take into account the presence, placement, duplication and inflectional correctness of lexical terms in the output sentence.

1 Introduction

The incorporation of an inflected lexicon into Neural Machine Translation (NMT) enables system developers to adapt the translation to specific domains, and users to adjust translations of phrases generated by the translation system.

Phrase-Based Statistical Machine Translation (PB-SMT; Setiawan et al., 2005) provided control over system output, e.g. by using a domain-specific lexicon. The shift from phrase tables in PB-SMT to a continuous-valued representation of text in NMT has made it more difficult to incorporate lexical constraints into the translation process. The task of integrating the lexicon and a neural translator is even more challenging for highly morphological languages, when the lexical items should be correctly inflected in the output text.

We carry out experiments for translation with inflected lexical constraints. As the target language of the translation we choose Polish, whose inflection is typical of the Slavic languages. The number of declination cases is six, and the verbal groups are inflected by tense, number, and person. In terms of correct inflection of the output, translation from English to Polish seems to be a more challenging task than translation in the other direction.

Unlike in some preceding experiments, we require that the lexicon may be modified after the model training has been completed. We believe that in post-editing mode users expect the translation engine to immediately mirror their adjustments to the lexicon.

2 Related Work

One of the first papers that addressed the incorporation of a lexicon into an NMT system was Arthur et al. (2016). The authors noticed that NMT systems tend to produce unexpected output for low-frequency words (such as names of countries). The solution proposed there consisted in designing probability lexicons and combining them with probabilities calculated by an NMT model. Let us note that the motivation for that research was the avoidance of major translation errors, rather than domain adaptation.

Anderson et al. (2017) introduced the concept of a Constrained Beam Search (CBS) in the task of picture captioning. The proposed algorithm forces the inclusion of selected tag words in the output. The solution makes it possible to apply, in the caption, words that were never present in the training data. The method yields the desired results provided that these out-of-vocabulary tags are based on "ground truth", such as labels obtained by reliable object detectors.

The application of CBS for lexical interference in the process of neural text generation was investigated in Hokamp and Liu (2017). In the decoding phase, the beam is limited only to hypotheses, which include predefined phrases or words. The algorithm called the Grid Beam Search (GBS) may be used for various text-generation tasks where auxiliary knowledge is expected to be incorporated into the text output. If applied to translation, the solution searches for lexical items in the source text and, in positive cases, imposes the presence of their equivalents on the beam.

Hasler et al. (2018) pointed out a danger in the CBS method resulting from the lack of correspondence between constraints and the source words they cover – the placement of the constraint translation in the output may not be correct. To avoid this undesirable effect, the authors "employ alignment information between target-side constraints and their corresponding source words."

The downside of the above algorithms is their complexity: exponential (CBS) or linear (GBS) in the number of constraints. Post and Vilar (2018) introduce an improvement of the GBS algorithm, called Dynamic Beam Allocation (DBA), which divides the fixed-size beam into "banks": sets of hypotheses that satisfy the same number of constraints. The algorithm depends only on the sentence length and the beam size, being independent of the number of constraints.

Hu et al. (2019) notice that the use of positive (specific tokens must be present in the output) or negative (specific tokens must not be generated) constraints may be useful in rewriting tasks other than translation. Rewriting (see e.g. Napoles et al., 2016) consists in generating an output sentence in the same language and similar in meaning to the input. Examples of such tasks are paraphrasing, question answering and natural language inference. Hu et al. (2019) regard it as crucial to focus on complexity issues to speed up the process of constrained text generation. They develop a "vectorized DBA algorithm with trie representation", which speeds up the computations fivefold compared with the standard DBA algorithm.

Further complexity improvements to constrained NMT are suggested in Song et al. (2019). They apply the idea of so-called "code-switching", which consists in injecting the target terms to the source side of the training data. The idea is similar to that of using placeholder tags to stand for rare names (Luong et al., 2015) or named entities (Deng et al., 2017). The difference is that the direct translations of terms are placed in the source text instead of tags. The output text is then left untouched. The authors claim that the idea improves translation because it "does not hurt unconstrained words." We believe, however, that in some (not rare) cases the replacement of the constrained word(s) should have an impact on the choice of unconstrained words.

Dinu et al. (2019) apply the idea of "code-switching" in two different scenarios. Depending on the experimental setup the target terms are placed either beside or in place of their source equivalents.

The code-switching method is faster than the previous implementations based on constrained decoding (the presence of constraints need not be verified in the beam). The downside is that it requires interference with the training data.

Exel et al. (2020) verify the efficiency of the code-switching method in an industrial scenario. They inject the terminology of the SAP company into two translation pairs, English–German and English–Russian, and provide both automatic and human evaluation.

From our point of view, the English–Russian case is more interesting because it addresses the problem of inflected forms of lexical constraints. There are two questions of interest to us:

1. How to ensure that the terms are inserted into the target sentence in the correct inflected form?

2. How to evaluate the correctness of term inflection in the translation?

We could not find answers to the above questions in the paper. Therefore, we investigated other solutions, such as the Levenshtein Transformer, introduced in Gu et al. (2019). The method uses "dual policy learning", which consists in using two adversary policies during learning: when training one policy, the output from its adversary at the previous iteration is used as input. In the Levenshtein Transformer the two policies are deletion and insertion of a token in the generated text. The idea is supposed to resemble human intelligence, which sometimes chooses to delete an item from the text intended as output.

In Susanto et al. (2020) the Levenshtein Transformer was used to incorporate lexical constraints in NMT. The idea seemed more appealing to us than code-switching because it does not interfere with the training procedure. However, our initial experiments with the methodology did not succeed – the inflected forms of lexicon entries were not generated correctly. Finally, we decided to carry out our experiments with the base Transformer model, as introduced by Vaswani et al. (2017), and design an algorithm that handles inflected forms of lexical constraints based on the GBS algorithm.

3 Experiments

The purpose of our experiments was to find an efficient solution that applies lexical constraints in interactive-mode translation into a morphologically rich language. To be more specific, we aimed to develop a method that would satisfy the following conditions:

- The translation takes into account inflection of lexical items;

- The training data need not be modified.

3.1 Evaluation metrics

We used the standard BLEU metric for translation quality evaluation on the untokenized reference sentences. We also wanted to verify whether the following conditions are satisfied:

1. The target term is present in the output sentence;

2. The target term is properly placed;

3. The target term is not duplicated;

4. The target term is correctly inflected.

Following Exel et al. (2020), we used the Term Rate (TR) to evaluate condition 1. We define Placement Rate (PR) to evaluate condition 2, Duplication Rate (DR) to evaluate condition 3, and Inflection Rate (IR) to evaluate condition 4.

$$TR = \frac{count(terms\ generated\ in\ output)}{count(terms\ that\ appeared\ in\ input)}$$

$$PR = \frac{count(terms\ placed\ properly\ in\ output)}{count(terms\ generated\ \ in\ output)}$$

$$DR = \frac{count(terms\ not\ duplicated\ in\ output)}{count(terms\ generated\ in\ output)}$$

$$IR = \frac{count(terms\ inflected\ properly\ )}{count(terms\ generated\ in\ output)}$$

3.2 Lexical constraints

The lexical constraints were extracted from Paterson (2015), a compendium of Polish and English accounting forms, available under a Creative Commons license. The number of extracted term pairs was 1197.

We used the Google search engine to obtain inflected forms of Polish terms. Specifically, we queried the search engine with the base forms of terms and scraped snippets from the first 20 pages of query results. We then limited the number of inflected variants to those that covered 95% of cases (we found out that 5% rare cases were more often than not erroneous). The most frequent number of inflected forms for one term was between two and five.

This language-agnostic approach allowed us to obtain the most widely used inflected forms of multi-word phrases, which are not present in Polish vocabularies such as *SGJP*,[1] which only include inflected forms of single words.

3.3 Data preparation

The direction of translation was from English into Polish. The training corpus consisted of the *Europarl v8*, *EUBookshop v2*, *JRC-Acquis v3.0*, *TildeMODEL v2018* and *Wikipedia v1.0* corpora and most of *DGT v2019*. All corpora were downloaded from the *OPUS*[2] collection (Tiedemann, 2012) and filtered using the *Bicleaner*[3] and *Bifixer*[4] (Ramírez-Sánchez et al., 2020) tools. The size of the training corpus after filtering was 3,103,819 segments.

For the validation set, we used 2000 sentences from the *DGT* corpus, removing them from the training set.

For the test sets, for two experiments, we extracted respectively 1000 and 1104 segment pairs from the *DGT* corpus, making sure that they did not overlap with either the training set or the validation set. The first test set contained randomly selected segments in which at least one lexical term appeared in the source-side segment, regardless of the presence of target lexical equivalents. We further refer to this experiment as the general scenario. The second test set contained all segments from the corpus in which, for each lexical term in the source-side segment, one of the inflected forms of its lexical equivalent appeared in the target-side segment. We refer to this as the domain-specific scenario.

All of the sets were processed by the BPE algorithm (Sennrich et al., 2016) with the SentencePiece tool[5] (Kudo and Richardson, 2018).

3.4 Experimental setup

We carried out our experiments using *fairseq*[6] (Ott et al., 2019), a PyTorch-based open-source sequence modeling toolkit.

We designed a lexicon where for each entry in the source language we provided multiple inflected forms of the corresponding entry in the target language, as described in 3.2. In order to use constrained decoding, we trained the Transformer model with a base configuration of six encoding and decoding layers, as introduced by Vaswani et al. (2017).

[1] http://sgjp.pl

[2] https://opus.nlpl.eu/

[3] https://github.com/bitextor/bicleaner

[4] https://github.com/bitextor/bifixer

[5] https://github.com/google/sentencepiece

[6] https://github.com/pytorch/fairseq

To obtain translations with correct inflected forms of lexical constraints, we introduced the following algorithm, which applies constrained decoding:

1. Translate the input sentence without any lexical constraints; calculate its average log-likelihood score.

2. Use the fuzzy search (see below) to check whether all lexical constraints are satisfied in the translation; end if the answer is positive.

3. For each unsatisfied lexical constraint:

    (a) Take all inflected forms of its lexical equivalent from the lexicon.

    (b) For each inflected form:
    Use lexically constrained decoding to translate the input sentence with the inflected form required to be present in the output.

    (c) Select the inflected form for which the translation has the highest average log-likelihood score.

4. Use lexically constrained decoding to generate the translation with the list of constraints selected in step 3.

5. Mark the translation as "ok" if the score of the selected translation is not worse than half of the score of the unconstrained translation; otherwise mark it as "warning".

Marking translation output as "warning" allowed us to detect potential errors in the constrained translation (mismatched context, a missing morphological form), thus making it possible to revert to the unconstrained translation if an error was detected.

In the fuzzy search (step 2 of the algorithm) we applied the Token Sort Ratio method, as implemented in the *spaczz*[7] library. The Token Sort Ratio algorithm splits the compared strings into tokens, sorts each list of tokens alphabetically and compares the corresponding elements of the lists using the Levenshtein distance on the level of characters. We considered the found term to match the search term if the similarity ratio, calculated by the algorithm, was not lower than 90%.

We used a beam size of 5 for decoding in step 3(b) of the above algorithm. We used a beam size of 12 in steps 1 and 4.

3.5 Evaluation

The baseline for our solution is the translation without lexical constraints. To assess the effectiveness of our method, we compared it with the baseline in the general and domain-specific scenarios and verified the following aspects of its performance:

1. translation quality (BLEU score);

2. translation speed (measured in seconds);

3. Term Rate;

4. Placement Rate;

5. Duplication Rate;

6. Inflection Rate.

[7] https://github.com/gandersen101/spaczz

We performed a manual check to calculate the Term Rate, Placement Rate, Duplication Rate and Inflection Rate. The BLEU scores were calculated using the *SacreBLEU*[8] tool (Post, 2018).

We calculated separate BLEU scores for the entire test sets and for the set of sentences for which the constrained decoding was actually used (i.e. sentences for which the result of unconstrained translation did not satisfy all of the lexical constraints). Additionally, we calculated the BLEU score for the scenario where "warning" translations are reverted to the unconstrained translations. Manual evaluation metrics were calculated for the entire test sets.

The speed tests were performed on a single NVIDIA RTX 2070 GPU and the AMD Ryzen 7 3700X 8-core processor, using the entire test sets. When translating with the lexicon, the first (unconstrained) and last (with all selected inflected forms) translations were performed with a batch size of 1, while the search for the correct inflected forms was performed as a single batch with the size depending on the number of constraints and their inflected forms. The time spent on the search for the appearance of lexicon entries was also included. When translating without a lexicon, we used a batch size of 1.

In the tables of results, we refer to the unconstrained translation as *base*, the translation using the lexicon as *lexicon*, and the translation using the lexicon with reversion to the original in case of "warning" as *lexicon-revert*.

3.5.1 Experiment 1: general scenario

In this scenario the test set consisted of sentences which contained lexical terms in the source text, independently of the presence of their equivalents in the target text.

Constrained decoding was used in the translation of 622 out of 1000 sentences, which corresponds to 62.20% of the entire test set. In these 622 translated sentences, 404 were marked as "ok" and 218 as "warning". In the 378 sentences where constrained decoding was not used, the unconstrained translation satisfied all lexical constraints.

The BLEU results for the experiment are presented in Table 1, the manual evaluation results for the *lexicon* translation type are presented in Table 2, and translation speed results are presented in Table 3.

Table 1: BLEU scores obtained in the general scenario

Translation type	Entire set	Constrained sentences
base	**42.21**	**41.67**
lexicon	39.91	37.59
lexicon-revert	40.97	39.68

Table 2: Results of manual evaluation of *lexicon* translation type in the general scenario

Metric	Result
Term Rate	98.90
Placement Rate	90.79
Duplication Rate	97.00
Inflection Rate	76.48

[8] https://github.com/mjpost/sacrebleu

Table 3: Translation speed in the general scenario

Translation type	Time result (s)
base	**273.88**
lexicon	1200.26

Unsurprisingly, the BLEU results are higher for translation without using the lexicon. This is consistent with the intuition that in the general scenario using the lexicon to correct the neural translation leads to a decrease in the BLEU score. The reversion to the unconstrained translation in situations where the output was marked "warning" may mitigate this effect to some extent. The reversion was particularly helpful in situations where the output from translation with the lexicon was corrupted; for instance, when constraints were placed at the end of the generated sentence or in the wrong inflected form, due to mismatched context or absence of the correct inflected form of the term in the lexicon.

The manual evaluation results indicate that the constraint accuracy in the general scenario is high for three metrics: Term Rate, Placement Rate and Duplicate Rate. Inflection Rate, however, is rather low because of the missing relevant inflected forms of the terms in the lexicon.

Term Rate is lower than 100% because in a few cases the lexical equivalent was generated in a different inflected form than any of the forms present in the lexicon. This is due to the fact that constraints are also divided into subwords (by the BPE algorithm) before the constrained decoding. In some rare cases this may lead to the proper generation of constraint subword units in the output sentence, but to a different constraint form than is required after the sentence is "de-BPEed".

Translation speed results show that constrained decoding significantly slows down the translation process. The decrease in speed is dependent on the number of constraints and the number of inflected forms of target lexical terms.

3.5.2 Experiment 2: domain-specific scenario

In Scenario 2 we evaluated the effectiveness of lexically constrained translation for the sentences where all lexical constraints were satisfied in the reference translation.

Constrained decoding was used in the translation of 150 out of 1104 sentences, which corresponds to 13.59% of the entire test set. In these 150 translated sentences, 143 were marked as "ok" and 7 as "warning". In the 954 sentences where constrained decoding was not used, all lexical constraints were satisfied in the unconstrained translation.

The BLEU results for the experiment are presented in Table 4, the manual evaluation results for the *lexicon* translation type are presented in Table 5, and translation speed results are presented in Table 6.

Table 4: BLEU scores obtained in the domain-specific scenario

Translation type	Entire set	Constrained sentences
base	42.30	36.17
lexicon	**42.76**	**39.80**
lexicon-revert	42.73	39.54

Table 5: Results of manual evaluation of *lexicon* translation type in the domain-specific scenario

Metric	Result
Term Rate	99.37
Placement Rate	98.37
Duplication Rate	99.09
Inflection Rate	97.28

Table 6: Translation speed in the domain-specific scenario

Translation type	Time result (s)
base	**316.79**
lexicon	540.56

The BLEU metric results show that translation with the lexicon leads to an increase in translation quality when the context of the input sentences matches the context of the lexicon and when the relevant inflected forms are present in the lexicon. Reverting to the translation without constraints in situations where the output was marked as “warning” resulted in a very slight decrease in the BLEU score. This is probably due to the fact that such cases were too rare for the results to be reliable.

The manual evaluation results indicate that our method is very effective in selecting a correct inflected form of the constraint in the domain-specific scenario. All of the metrics returned high scores, including the Inflection Rate.

In this scenario, lexical constraints were not satisfied in the unconstrained translation only in 13.59% of cases. This shows that the neural translation model itself is capable of generating translations with the correct terminology given adequate context. It is concluded that the use of lexical constraints in NMT improves translation quality only in scenarios where the lexicon is highly specific for the translation context.

3.6 Examples of translation with inflected lexicon

Table 7 shows two examples of sentences translated with and without the use of inflected lexicon. The lexicon entries consist of a term in English language with the equivalent in Polish language along with its comma-separated list of inflectional forms.

4 Conclusions

We have examined a new approach to terminology translation into a morphologically rich language with the use of lexicons. We verified that our method, based on constrained decoding, enables the selection of accurate inflected forms of lexical constraints. The method yields an increase in the BLEU metric score provided that appropriate lexical variants of terms are present in the lexicon and the input sentence context is consistent with the lexicon entries. The cost of the algorithm is a decrease in the translation speed. We proposed new metrics for the evaluation of terminology translations: Placement Rate, Duplication Rate and Inflection Rate. The manual evaluation results show that our method ensures terminological adequacy and consistency when translating into a morphologically rich language in domain-specific scenarios.

5 Future Work

We believe that there is still much to explore in the field of terminology translation. In future experiments, we plan to compare our solution with the code-switching approach (Dinu et al.,

Table 7: Examples of translation with inflected lexicon

Lexicon entry	audit committee -> komisja rewizyjna, komisji rewizyjnej, komisją rewizyjną, komisję rewizyjną
Source sentence	The **audit committee** should be composed exclusively of non-executive or supervisory directors.
Translation without lexicon	Komitet ds. audytu powinien składać się wyłącznie z dyrektorów niewykonawczych lub będących członkami rady nadzorczej.
Translation with lexicon	W skład **komisji rewizyjnej** powinni wchodzić wyłącznie dyrektorzy niewykonawczy lub będący członkami rady nadzorczej.
Lexicon entry	outlay -> nakład, nakładu, nakłady, nakładów
Source sentence	The statement of the beneficiary's outlay shall be produced in support of any request for a new payment.
Translation without lexicon	Deklarację wydatków beneficjenta przedstawia się na poparcie każdego wniosku o nową płatność.
Translation with lexicon	Deklarację **nakładów** beneficjenta przedstawia się na poparcie każdego wniosku o nową płatność.

2019), (Song et al., 2019) and to investigate methods which do not have such a negative impact on translation speed as constrained decoding. Another potential direction for improvement is to design a method that does not require the presence of multiple inflected forms in the lexicon before translation.

References

Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2017). Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Arthur, P., Neubig, G., and Nakamura, S. (2016). Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas. Association for Computational Linguistics.

Deng, Y., Kim, J., Klein, G., Kobus, C., Segal, N., Servan, C., Wang, B., Zhang, D., Crego, J. M., and Senellart, J. (2017). SYSTRAN purely neural MT engines for WMT2017. In Bojar, O., Buck, C., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno-Yepes, A., Koehn, P., and Kreutzer, J., editors, *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 265–270. Association for Computational Linguistics.

Dinu, G., Mathur, P., Federico, M., and Al-Onaizan, Y. (2019). Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Exel, M., Buschbeck, B., Brandt, L., and Doneva, S. (2020). Terminology-constrained neural machine translation at SAP. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 271–280, Lisboa, Portugal. European Association for Machine Translation.

Gu, J., Wang, C., and Zhao, J. (2019). Levenshtein transformer. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hasler, E., de Gispert, A., Iglesias, G., and Byrne, B. (2018). Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Hokamp, C. and Liu, Q. (2017). Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Hu, J. E., Khayrallah, H., Culkin, R., Xia, P., Chen, T., Post, M., and Van Durme, B. (2019). Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., and Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Napoles, C., Callison-Burch, C., and Post, M. (2016). Sentential paraphrasing as black-box machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 62–66, San Diego, California. Association for Computational Linguistics.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Paterson, R. (2015). In *Compendium of Acounting in Polish & English*. Ministerstwo Finansów by arrangement with the Swiss-Polish Cooperation Program – Financial Reporting Technical Assistance Program (FRTAP).

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Post, M. and Vilar, D. (2018). Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Ramírez-Sánchez, G., Zaragoza-Bernabeu, J., Bañón, M., and Ortiz-Rojas, S. (2020). Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Sennrich, R., Haddow, B., and Birch, A. (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.

Setiawan, H., Li, H., Zhang, M., and Ooi, B. C. (2005). Phrase-based statistical machine translation: A level of detail approach. In *Second International Joint Conference on Natural Language Processing: Full Papers*.

Song, K., Zhang, Y., Yu, H., Luo, W., Wang, K., and Zhang, M. (2019). Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Susanto, R. H., Chollampatt, S., and Tan, L. (2020). Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In Calzolari, N., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.