Predicting Anchored Text from Translation Memories for Machine Translation Using Deep Learning Methods

Richard Yue yue.r@northeastern.edu
Northeastern University, San Jose, CA

John E. Ortega j.ortega@northeastern.edu
Institute for Experiential AI, Northeastern University, Boston, MA

Abstract

Translation memories (TMs) are the backbone for professional translation tools called computer-aided translation (CAT) tools. In order to perform a translation using a CAT tool, a translator uses the TM to gather translations similar to the desired segment to translate (s'). Many CAT tools offer a fuzzy-match algorithm to locate segments (s) in the TM that are close in distance to s'. After locating two similar segments, the CAT tool will present parallel segments (s, t) that contain one segment in the source language along with its translation in the target language. Additionally, CAT tools contain fuzzy-match repair (FMR) techniques that will automatically use the parallel segments from the TM to create new TM entries containing a modified version of the original with the idea in mind that it will be the translation of s'. Most FMR techniques use machine translation as a way of "repairing" those words that have to be modified. In this article, we show that for a large part of those words which are *anchored*, we can use other techniques that are based on machine learning approaches such as Word2Vec. BERT, and even ChatGPT. Specifically, we show that for anchored words that follow the continuous bag-of-words (CBOW) paradigm, Word2Vec, BERT, and GPT-4 can be used to achieve similar and, for some cases, better results than neural machine translation for translating anchored words from French to English.

1 Introduction

Professional translators use computer-aided translation (CAT) tools (Bowker, 2002) to translate text from one language called the source language (SL) to a target language (TL). Most CAT tools have an option known as *fuzzy-match repair* (FMR) (Kranias and Samiotou, 2004; Hewavitharana et al., 2005; Dandapat et al., 2011; Ortega et al., 2016; Bulté et al., 2018; Tezcan et al., 2021), which is backed by a parallel translation memory (TM) that contains sentences (called segments) in the SL and TL. Each pair, or unit, of parallel segments in the TM is known as a *translation unit* (TU). A TU contains a source segment (s) along with a target segment (t). When a professional translator attempts to translate a segment in the SL (denoted as s') a fuzzy-match lookup is performed using a word-based Levenshtein distance (Levenshtein, 1966) between s' and s where a 100% match means that the words from s' are identical to the words in s. It is often the case that a professional translator uses matches from FMR to only translate a few words (called sub-segments) from the entire segment. In this article, we focus on improving those cases where there exists only one word to translate, known as an *anchored* word, whose position is in between two words that are already captured. In our studies, the anchored word is a common case that professional translators often use. We experiment with four techniques to translate the anchored word: (1) Neural Machine Translation, (2) a BERT-based (Sanh et al., 2019) implementation, (3) Word2Vec (Mikolov et al., 2013) and (4) OpenAI GPT-4 prompting (Achiam et al., 2023).

The prediction of an anchored word has been presented in many contexts and can be considered the main objective of a language model. Several models based on attention allow a weight to be assigned to certain words within a context window so that surrounding words that strongly influence the overall context can have a greater impact on the prediction made. This could potentially be used in order to improve predictions made for anchored text by taking longer contexts into account than the surrounding words. We discuss this approach in the context of generative models, where such systems could be harnessed to generate highly accurate predictions.

The rest of this article is structured as follows. The next section discusses related work by accentuating the differences between FMR based on MT and anchored-word prediction. Section 3 then presents the BERT, Word2Vec, and GPT-4 approaches used for translating anchored words. In Section 4, we describe the corpus and configurations used for our experiments whose results are reflected in Section 5, followed by concluding remarks in Section 6.

2 Related Work

For the majority of FMR approaches, MT is used to translate mismatches, regardless if they are anchored words or not. Generally, MT techniques for FMR are focused on the decoding process where statistical-based systems (Biçici and Dymetman, 2008; Simard and Isabelle, 2009; Zhechev and Genabith, 2010; Koehn and Senellart, 2010; Li et al., 2016; Liu et al., 2019) or neural-based systems (Ortega et al., 2014, 2016; Gu et al., 2018; Bulté et al., 2018; Bulte and Tezcan, 2019) are used in such a manner to "repair" either the MT system or the mismatched sub-segments between s' and s. This article is focused on repairing the mismatched subsegments in specific situations where sub-segments of s are common in s' with the exception of one word (e.g. s='the **brown** dog' and s'='the **red** dog').

Previous work (Ortega et al., 2016; Bulté et al., 2018) can be considered identical to this article as it uses FMR to first find mismatches between s' and s and then translates the missing words with different MT systems. However, their system uses context around *all* mismatches where we only consider mismatches with anchored words, similar to Kranias and Samiotou (2004). While other techniques (Hewavitharana et al., 2005; Dandapat et al., 2011) are based on probabilistic MT models or employ different algorithms for aligning s' and s, we use a word-based edit distance (Levenshtein, 1966; Wagner and Fischer, 1974) that marks the mismatched sub-segments and discards non-anchored words.

Tezcan and Bulté (2022) investigate a wide range of automatic quality estimation (QE) metrics in order to assess what effect integrating fuzzy matches can have on a number of aspects of translation quality, in addition to performing manual MT error analysis. They further evaluate what influence fuzzy matches have on a translation and how further quality improvements can be made by quantitative analyses that focus on the specific characteristics of a retrieved fuzzy match. Neural Fuzzy Repair (NFR) outperforms baselines in all automated evaluation metrics. There was not a discernable difference between NFR and Neural Machine Translation (NMT) error in manual evaluation, but different error profiles emerged in this study, highlighting some of the strengths and weaknesses of each method. Namely, NFR produced more errors in the category of "semantically unrelated", whereas the baseline NMT system produced more errors in the categories of "word sense" and "multi-word expression". The NFR system made more accuracy errors, but producing fluent output was its strong suit. Meanwhile, in terms of lexical choices, NMT produced more "non-existing/foreign" errors, which was not an issue for NFR. The baseline system performed better on grammar and syntax. Our study differs in that it focuses specifically on anchored text and on leveraging the strengths of language models in next word prediction in order to fill in single-word gaps.

Esplà-Gomis et al. (2011) attempt to improve CAT via the TM using pre-computed word alignments between source and target TUs in the TM. When a user is translating s' with a fuzzy match score greater than or equal to 60%, the proposed system marks the words that need to change as well as those that must remain the same in order to obtain t'. Alignments are obtained from GIZA++ (Brown et al., 1993; Vogel et al., 1996) and take both a statistical and syntactic approach to detecting where changes need to occur. The experiments offer insight into how human decisions to keep/change text during translation can be integrated into FMR. Our approach differs in that we specifically locate anchored text and, following that, continue on to a prediction step, providing the content needed to perform fuzzy match repair in the translation step.

İrsoy et al. (2020) compare performance of pre-trained word embeddings in use in language models such as BERT with continuous bag of words (CBOW) embeddings trained with Word2Vec. The authors claim that, while BERT embeddings are useful and effective, they often offer only marginal gains as compared to Word2Vec embeddings trained using Gensim (Řehřek et al., 2011). The latter are much less computationally expensive to obtain; 768-dimensional vectors were trained in one epoch in 1.61 days on a 16-CPU machine. CBOW embeddings are trained by using surrounding context to predict a center word. While training via CBOW has often shown inferior performance to training via skipgram (SG), this paper shows that with a proper implementation, performance of CBOW embeddings can be on par with SG. Our work puts the CBOW prediction objective to good use, harnessing it to predict anchored text in source language segments.

3 Methodology

Neural MT systems have been shown by previous work (Bulte and Tezcan, 2019) to be the state-of-the-art for FMR. In this article, we experiment on the one hand with word-based language models that are trained using context around a word, like those that use the continuous bag of words (CBOW) model (Mikolov et al., 2013) (Word2Vec) or masked language modeling (Sanh et al., 2019) (BERT). On the other hand, it is our belief also that generative language modeling techniques may be a good candidate for accomplishing this task. To explore this avenue, we also compare output from these models with predictions obtained from prompting GPT-4 and find it to be competitive with the other methods. An example of a source sentence and the output from each method is provided in Table 1 with predicted (or reference) word in bold. In our experiment, the two language modeling techniques as well as the generative approach are compared against machine translation and measured using character rate and accuracy against sets of anchored words from the test set. A prediction or translation was deemed correct when the center word from a tri-gram of anchored words was correctly found. In the following sub-sections, we discuss each approach. In Section 4 we provide further details about the corpora and configuration.

3.1 Machine Translation

We train the neural MT system with Open-NMT (Klein et al., 2020) using the default transformer configuration. In order to get a wider range of difference with the MT system, we translate using two methods: (1) the translation of the s'_{en} segment to t^*_{fr} then translation from t^*_{fr} to s^*_{en}; and, (2) the translation of the three-word sub-segment only (i.e. the anchored tri-gram with the center word to be translated) from $s'_{trigram\text{-}en}$ to $t^*_{trigram\text{-}fr}$ then translation from $t^*_{trigram\text{-}fr}$ to $s^*_{trigram\text{-}en}$. For both methods, correctly translated center words from tri-grams were counted in the overall evaluation. Predictions by the other two methods were scored similarly. Further details on parameters and configuration are found in Section 4.2.

3.2 Word2Vec

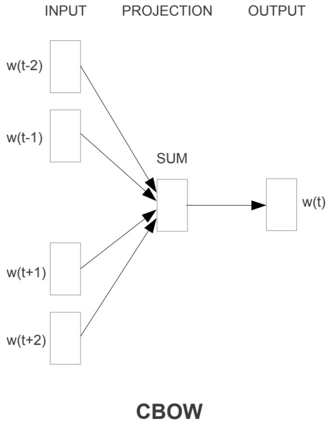

Figure 1: An illustration of predicting a word given the context around it (denoted as *anchored words in this article)*, called Continuos Bag of Words (CBOW) by Mikolov et al. (2013).

We used a pre-trained language model (PLM) for experimentation with Word2Vec (Mikolov et al.,

[1] We use the pre-trained word news vectors from Google found here:https://github.com/mmihaltz/word2vec-GoogleNews-vectors?tab=readme-ov-file.

Method	Output
Original French	“Afin d’évaluer si l’établissement identifie toutes les situations qui doivent être considérées comme des défauts, conformément à l’article 178, paragraphes 1 à 5, du règlement (UE) no 575/2013...”
Reference Translation	“In order to assess whether the institution identifies all situations which are to be considered defaults in accordance with Article 178 (1) to (5) of Regulation (EU) No 575/2013...”
Reference tri-gram	assess **whether** the
BERT	assess **whether** the
Word2Vec	assess **commission** the
MT	assess **obligatory** the
GPT-4	assess **and** the

Table 1: Anchored tri-gram reference and predictions (predicted word in bold)

2013).[1] The hope is that through the use of a PLM we can capture context in several different domains, specifically the corpus that we use which is parliamentary in nature.

The PLM weights from Word2Vec were used as a manner to predict anchored words due to the fact that the training method for them is based on a CBOW model. CBOW was selected because, as shown in Figure 1, its training objective most closely resembles the task we are trying to accomplish—the prediction of a word surrounded by anchored text (one word on the left and one word on the right).

As a first step, the PLM was downloaded and experimented as-is in its out-of-the-box state which consists of 300 dimensions and a default vocabulary. Then, in order to fine-tune the model, we adapted it to our parliamentary corpus. After the fine-tuning of the model, anchored tri-grams were extracted from s' and used as input to the PLM where the center word is used for prediction and the left and right “anchors” are used as input one-hot encoded embeddings, similar to the training exercise from Mikolov et al. (2013). Further details on parameters and configuration are found in Section 4.3.

3.3 BERT

Models based on the BERT (Kenton and Toutanova, 2019) algorithm are used frequently in modern times. They use an attention mechanism (Vaswani et al., 2017) and are known to be capable of capturing information better than previous implementations such as Word2vec. Therefore, in order to compare both algorithms to MT for predicting anchored words, we experiment with DistilBERT (Sanh et al., 2019), a BERT-based model that uses masked language modeling that in theory captures more parameters than the Word2Vec CBOW model.

Similar to the Word2Vec method, we fine-tune our DistilBERT model on the parliamentary corpus with a masked language modeling objective. We chose the masked language modeling objective as it is the most similar objective to CBOW. Further details on parameters and configuration are found in Section 4.4.

3.4 GPT-4

We experiment with prompting GPT-4 to predict anchored text using a temperature of 0 and the following prompt: *“You are an expert lexicographer and natural language processing assistant. Additionally, you are highly specialized in parliamentary proceedings. Given a trigram I provide with a ’?’ character in the center word, I need you to predict the ’?’ character with the most likely single-word token. Please return one predicted token without any text except the predicted token in your response. Do not provide the surrounding text or any additional information. Do not include the text ’predicting’, ’predict’, ’prediction’, ’predicted’ ’the predicted token is’ or ’The predicted token is’ in your response. Do not include any extra characters such as apostrophes, commas, colons, or semicolons in your re-*

sponse. Do not include any newline characters in your response.”.

4 Experimental Settings

4.1 Corpus

The corpus consists of 393,371 SL-TL pairs of European parliamentary proceedings, a freely available translation memory (Steinberger et al., 2012) obtained from the European Commission DGT-Translation Memory repository.[2] The corpus is divided randomly with a random state of 42. We divide the corpus up into 70% train, 20% dev, and 10% test sets as shown in in Table 2.

4.2 Machine Translation

As mentioned previously, we use the Open-NMT (Klein et al., 2020) framework to build our French to English (FR–EN) and English to French (EN–FR) MT system. The system is based on a transformer architecture model with the following hyperparameters: A maximum sequence length of 500, an early stopping parameter of 4, 7,800 train steps, 1,000 validation steps, a bucket size of 262,144, a batch size of 4,096, and a validation batch size of 2,048. The optimizer is an Adam (beta2 of 0.998) optimizer with with fp16 activated, a learning rate of 2, noam decay, label smoothing of 0.1, a hidden size of 512, word vector size of 512, 8 attention heads, a dropout of 0.1, and an attention dropout of 0.1. The choice of parameter selection is inspired by previous work from Yasmin Moslem.[3]

In order to verify that the NMT system is on-par with the latest MT systems for FR–EN and EN–FR, we first test the system in both directions on the test set. During test, we achieved a BLEU score of 55.84 for FR–EN and 62.60 for EN-FR. Nonetheless, as we show in Section 5, the translation of anchored words as measured by character rate and accuracy was not remarkable.

4.3 Word2Vec

The CBOW algorithm for Word2vec is a well-known algorithm performed as a way of capturing semantics via a language model (Mikolov et al., 2013). We describe our Word2Vec CBOW implemenation. Before fine-tuning, the Word2Vec model has 300 dimensions with a window size of 2 and a minimum word count of 1. Additionally, predefined vocabulary is used in the Google News Vectors that contains billions of words. The model is fine-tuned with our training set which is tokenized using the NLTK RegexpTokenizer[4]. The embeddings created from the training set use lockf at 1.0 and a window size of 3, similar to Zarrar Shehzad.[5]

4.4 BERT

Our BERT model is based on a PLM called DistilBERT[6]. (Sanh et al., 2019) We train DistilBERT using the HuggingFace PyTorch Trainer with 10 training epochs, a learning rate of 2e-5, weight decay of 0.01, and FP16 mixed precision set to true. Hyperparameters are inspired by HuggingFace.[7]

4.5 GPT-4

GPT-4 was prompted using the gpt-4-turbo variant and queried repetitively through the OpenAI API. Due to newline mismatches that occurred during batch processing, we opted to run an API call for every line in the dataset.

5 Results

In this section, we compare the results obtained from running four approaches for predicting the anchored word: (1) Neural Machine Translation (NMT) (2) Word2Vec (3) BERT and (4) GPT-4. NMT is divided into the two approaches mentioned in Section 3.1 (sentence-level and tri-grams). Accuracy measurements are performed and reported for all holes[8]. Additionally, we report on character matches for each approach after dividing the segments into fuzzy-match thresholds, common practice for FMR work (see (Ortega et al., 2016; Bulte and Tezcan,

[2] https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory_en

[3] https://github.com/ymoslem/OpenNMT-Tutorial

[4] https://www.nltk.org/_modules/nltk/tokenize/regexp.html

[5] https://czarrar.github.io/Gensim-Word2Vec/

[6] https://github.com/huggingface/transformers/tree/main/examples/research_projects/distillation

[7] https://huggingface.co/learn/nlp-course/en/chapter7/3

[8] A hole is a span of a tri-gram where the center word is predicted.

Data set	**Segment Size**
Train	275,317
Development	77,877
Test	40,117
Total	393,371

Table 2: Experimental sets from the European Commission DGT Translation memory used for creating and evaluating the three approaches.

	60–69%	**70–79%**	**80–89%**	**90–100%**
BERT	**8.97**	**9.61**	**7.98**	**7.87**
GPT	4.82	5.58	3.85	2.74
W2V	3.39	3.46	2.89	3.02
NMT-1	0.15	0.14	0.28	0.19
NMT-2	3.75	4.36	4.16	6.35

Table 3: Accuracy scores for various fuzzy-match threshold on five deep-learning approaches.

2019)).

First, we report on character match rates for the three approaches. Character match is defined as the number of characters in the output token that correspond to characters in the desired string. In Figure 2, we report the average character match for GPT-4, BERT, Word2Vec, NMT-1 (segment-level MT) and NMT-2 (tri-gram MT). We observe a marked improvement in average character match with language modeling approaches (BERT and Word2Vec) and GPT-4 performs competitively in most cases. BERT outperforms all approaches across all fuzzy-match thresholds. From an MT standpoint, the secondary approach (called NMT-2 in Figure 2) outperforms the primary approach; it appears that in our experiments the translation of anchored tri-grams is better than translating the entire segment.

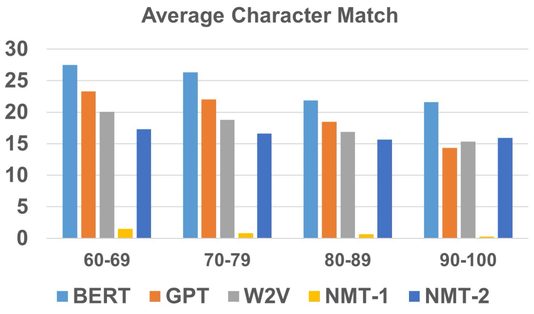

Figure 2: Average character match (y-axis) by fuzzy-match rate percentage (x-axis) by segment of the four experimental approaches: BERT, GPT, Word2Vec, Neural Machine Translation 1 and Neural Machine Translation 2 systems for different segment-level fuzzy-match thresholds.

Additionally, we measured the accuracy for the three approaches in order to better understand the hole span coverage. For accuracy, we measure only if prediction was correct or not; we do not take into account other predictions like blank, extra words, or others. To this end, we present accuracy scores in Table 3.

In our experiments, we notice that the NMT systems perform better on stop words and digits such as the phrase: “beyond **90** ghz”. Both the BERT and NMT systems were found to perform well in those situations. However, the MT system oftentimes did not replace one word only—in several cases it aggregated several words more. BERT performed well on average when compared with the other approaches. GPT remains competitive on all fuzzy match ranges except 90–100.

6 Conclusion

In this article, we have illustrated that via the use of a language model, predicting anchored words performed better in our experiments. The BERT model outperforms other approaches including neural machine translation (with two approaches) when measured via character match and tri-gram anchored word coverage.

We also demonstrate how generative models might be prompted to aid in predicting anchored text. It is our belief that this work could assist CAT tools backed by TMs and MT systems.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Biçici, E. and Dymetman, M. (2008). Dynamic translation memory: Using statistical machine translation to improve translation memory fuzzy matches. *Computational Linguistics and Intelligent Text Processing*, pages 454–465.

Bowker, L. (2002). *Computer-aided translation technology: a practical introduction*. University of Ottawa Press.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Bulte, B. and Tezcan, A. (2019). Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy.

Bulté, B., Vanallemeersch, T., and Vandeghinste, V. (2018). M3TRA: integrating TM and MT for professional translators. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 69–78, Alacant, Spain.

Dandapat, S., Morrissey, S., Way, A., and Forcada, M. L. (2011). Using example-based MT to support statistical MT when translating homogeneous data in a resource-poor setting. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation*, pages 201–208, Leuven, Belgium.

Esplà-Gomis, M., Sánchez-Martínez, F., and Forcada, M. L. (2011). Using word alignments to assist computer-aided translation users by marking which target-side words to change or keep unedited. In *European Association for Machine Translation Conferences/Workshops*.

Gu, J., Wang, Y., Cho, K., and Li, V. O. (2018). Search engine guided neural machine translation. In *Proceedings of the 32 AAAI Conference on Artificial Intelligence*, pages 5133–5140, New Orleans, USA.

Hewavitharana, S., Vogel, S., and Waibel, A. (2005). Augmenting a statistical translation system with a translation memory. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, pages 126–132, Budapest, Hungary.

İrsoy, O., Benton, A., and Stratos, K. (2020). Corrected cbow performs as well as skip-gram. *arXiv preprint arXiv:2012.15332*.

Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Klein, G., Hernandez, F., Nguyen, V., and Senellart, J. (2020). The opennmt neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 102–109.

Koehn, P. and Senellart, J. (2010). Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31, Denver, USA.

Kranias, L. and Samiotou, A. (2004). Automatic translation memory fuzzy match post-editing: a step beyond traditional TM/MT integration. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, pages 331–334, Lisbon, Portugal.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady.*, 10(8):707–710.

Li, L., Parra Escartín, C., and Liu, Q. (2016). Combining translation memories and syntax-based SMT. *Baltic Journal of Modern Computing*, 4:165–177.

Liu, Y., Wang, K., Zong, C., and Su, K.-Y. (2019). A unified framework and models for integrating translation memory into phrase-based statistical machine translation. *Computer Speech & Language*, 54:176–206.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ortega, J. E., Sánchez-Martínez, F., and Forcada, M. L. (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, vol. 1: MT Rsearchers)*, pages 42–53, Vancouver, BC, Canada.

Ortega, J. E., Sánchez-Martínez, F., and Forcada, M. L. (2016). Fuzzy-match repair using black-box 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, USA.

Řehřek, R., Sojka, P., et al. (2011). Gensim—statistical semantics in python. *Retrieved from genism. org*.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Simard, M. and Isabelle, P. (2009). Phrase-based machine translation in a computer-assisted translation environment. In *Proceeding of the 12th Machine Translation Summit (MT Summit XII)*, pages 120–127, Quebec, Canada.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., and Schlüter, P. (2012). DGT-TM: A freely available translation memory in 22 languages. In Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 454–459, Istanbul, Turkey. European Language Resources Association (ELRA).

Tezcan, A. and Bulté, B. (2022). Evaluating the impact of integrating similar translations into neural machine translation. *Inf.*, 13:19.

Tezcan, A., Bulté, B., and Vanroy, B. (2021). Towards a better integration of fuzzy matches in neural machine translation through data augmentation. In *Informatics*, volume 8, page 7. MDPI.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based word alignment in statistical translation. pages 836–841, Copenhagen.

Wagner, R. A. and Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173.

Zhechev, V. and Genabith, J. V. (2010). Seeding statistical machine translation with translation memory output through tree-based structural alignment. In *Proceedings of SSST-4 - 4th Workshop on Syntax and Structure in Statistical Translation*, pages 43–49, Dublin, Ireland.