informatics

MDPI

Article

Towards a Better Integration of Fuzzy Matches in Neural Machine Translation through Data Augmentation

Arda Tezcan [1,*], Bram Bulté [2] and Bram Vanroy [1]

[1] Language and Translation Technology Team (LT3), Ghent University, B-9000 Ghent, Belgium; Bram.Vanroy@UGent.be
[2] Centre for Computational Linguistics, KU Leuven, B-3000 Leuven, Belgium; bram.bulte@ccl.kuleuven.be
* Correspondence: Arda.Tezcan@UGent.be

Abstract: We identify a number of aspects that can boost the performance of Neural Fuzzy Repair (NFR), an easy-to-implement method to integrate translation memory matches and neural machine translation (NMT). We explore various ways of maximising the added value of retrieved matches within the NFR paradigm for eight language combinations, using Transformer NMT systems. In particular, we test the impact of different fuzzy matching techniques, sub-word-level segmentation methods and alignment-based features on overall translation quality. Furthermore, we propose a fuzzy match combination technique that aims to maximise the coverage of source words. This is supplemented with an analysis of how translation quality is affected by input sentence length and fuzzy match score. The results show that applying a combination of the tested modifications leads to a significant increase in estimated translation quality over all baselines for all language combinations.

Keywords: translation memories; data augmentation; fuzzy matching; NMT; sub-word units

check for updates

Citation: Tezcan, A.; Bulté, B.; Vanroy, B. Towards a Better Integration of Fuzzy Matches in Neural Machine Translation through Data Augmentation. *Informatics* **2021**, *8*, 7. https://doi.org/10.3390/informatics8010007

Received: 30 December 2020
Accepted: 24 January 2021
Published: 29 January 2021

1. Introduction

Recent advances in machine translation (MT), most notably linked to the introduction of deep neural networks in combination with large data sets and matching computational capacity [1], have resulted in a significant increase in the quality of MT output, especially for specialised, technical and/or domain-specific translations [2]. The increase in quality has been such that more and more professional translators, translation services, and language service providers have integrated MT systems in their workflows [3]. This is exemplified by recent developments in the translation service of the European Commission, one of the world's largest translation departments, where MT is increasingly used by translators, also driven by increased demands on productivity [4–6]. Post-editing MT output has been shown to, under specific circumstances, increase the speed of translation, or translators' productivity, compared to translation from scratch [5–7]. Specifically, higher-quality MT output, as assessed by automated evaluation metrics, has been shown to lead to shorter post-editing times for translators [8].

MT is currently most often used by translators alongside translation memories (TMs), a computer-assisted translation (CAT) tool that by now is a well-established part of many translation workflows [9–11]. MT tends to be used as a 'back-off' solution to TMs in cases where no sufficiently similar source sentence is found in the TM [12,13], since post-editing MT output in many cases takes more time than correcting (close) TM matches. This is, for example, due to inconsistencies in translation and a lack of overlap between MT output and the desired translation [14]. The level of similarity between the sentence to translate and the sentence found in the TM, as calculated by a match metric [15,16], thus plays an important role. Whereas translations retrieved from a TM offer the advantage of being produced by a translator, in the case of partial, or fuzzy matches, they are the translation of a sentence that is similar, but not identical to the sentence to be translated. In contrast, MT produces a translation of any input sentence, but in spite of the recent increase in MT quality, this output is still not always completely error-free. Moreover, the perception of translators is that MT errors are often not predictable or coherent, which results in a lower confidence for MT output in comparison to TM segments [14,17].

At least since the turn of the century, researchers have attempted to combine the advantages of TMs and MT, for example by better integrating information retrieved from a TM into MT systems [18–20]. In addition, in the context of neural machine translation (NMT), several approaches to TM-MT integration have shown that NMT systems do not fully exploit the available information in the large parallel corpora (or TMs) that are used to train them [21–23]. Whereas most of these TM-NMT approaches require modifications to the NMT architectures or decoding algorithms, an easy-to-implement method, neural fuzzy repair (NFR), was proposed that only involves data pre-processing and augmentation [24]. This method, based on concatenating translations of similar sentences retrieved from a TM to the source sentence, has been shown to increase the quality of the MT output considerably, also when state-of-the-art NMT systems are used [25]. However, this approach has only been implemented recently, and several important aspects remain to be explored. Amongst other things, fuzzy match retrieval in the context of NFR has so far used word-level information only, and combinations of fuzzy matches relied exclusively on the criterion of match score.

In this paper, we identify a number of adaptations that can boost the performance of the NFR method. We explore sentence segmentation methods using sub-word units, employ vector-based sentence similarity metrics for retrieving TM matches in combination with alignment information added to the retrieved matches, and attempt to increase the source sentence coverage when multiple TM matches are combined. We evaluate the proposed adaptations on eight different language combinations. In addition, we analyse the impact on translation quality of the source sentence length as well as the estimated similarity of the TM match. The code for fuzzy match extraction and data augmentation is available at https://github.com/lt3/nfr.

This paper is structured as follows: in the next part, we discuss relevant previous research (Section 2). This is followed by an overview of the tested NFR configurations (Section 3) and the experimental setup (Section 4). Section 5 presents the results, which are subsequently discussed (Section 6). In the final Section 7, conclusions are drawn up.

2. Related Research

We first briefly describe TMs and fuzzy matching techniques (Section 2.1), before discussing previous attempts to integrate TM and MT (Section 2.2). We then focus on TM-based data augmentation methods within the NMT framework (Section 2.3), the approach that is also followed in this paper. Finally, we list other related studies that followed a similar general approach (Section 2.4).

2.1. Translation Memories and Fuzzy Matching

Proposed in the 1970s [26], TMs were integrated in commercial translation software in the 1980s–1990s [11]. They have since become an indispensable tool for professional translators, especially for the translation of specialised texts in the context of larger translation projects with a considerable amount of repetition, and for the translation of updates or new versions of previously translated documents [10]. TMs are particularly useful when consistency (e.g., with regard to terminology) is important. A TM consists of a set of sentences (or 'segments') in a source language with their corresponding translation in a target language. As such, a TM can be considered to be a specific case of a bilingual parallel corpus. TM maintenance efforts can help to ensure that translation pairs are of high quality, and that the resulting parallel corpus is particularly 'clean'.

Full or partial matches of sentences to be translated are retrieved from the TM using a range of possible matching metrics in a process referred to as fuzzy matching. Fuzzy matching techniques use different approaches to estimate the degree of overlap or similarity between two sentences, such as calculating:

- the percentage of tokens (or characters) that appear in both segments [15], potentially allowing for synonyms and paraphrase [27],
- the length of the longest matching sequence of tokens, or n-gram matching [25],
- the edit distance between segments [28], generally believed to be the most commonly used metric in CAT tools [16],
- automated MT evaluation metrics such as translation edit rate (TER) [29,30],
- the amount of overlap in syntactic parse trees [31], or
- the (cosine) distance between continuous sentence representations [32], a more recently proposed method.

To facilitate the correction of partial matches, fuzzy matching is usually combined with the identification and highlighting of parts in the corresponding target segment that either can be kept unchanged or need editing.

2.2. Approaches to TM-MT Integration

Besides the fact that TMs are valuable resources for training in-domain MT systems, researchers working within different MT frameworks have attempted to improve the quality of automatically generated translations by harnessing the potential of similar translations retrieved from a TM. Such translations offer the advantage of being produced by translators, and with a well-managed TM their quality is assumed to be high [33]. If the retrieved source sentence is sufficiently similar to the sentence to be translated, even using its retrieved translation as such can result in better output than that generated by MT systems, both as judged by automatic evaluation metrics and in terms of post-edit effort needed by translators [13,34]. This most basic option for TM-MT integration, i.e., favouring TM matches over MT output based on a fixed (or tunable) threshold for a chosen similarity metric, can be combined with systems designed to automatically edit such matches to bring them closer to the sentence to be translated [35,36], an approach that is sometimes referred to as 'fuzzy match repair' [37]. Recent implementations of this approach have been demonstrated to outperform both unaltered fuzzy matches and state-of-the-art NMT systems [38,39].

Next to this 'dual' approach, different methods were developed to achieve a closer integration of TMs and different types of MT systems. It can be argued that the principles underlying the example-based MT paradigm as a whole are closely related to TM-MT integration and fuzzy match repair [40,41]. Within this framework, researchers have focused on different ways of using sub-segmental TM data for the purpose of MT [20,42]. To this end, example-based MT systems have also been combined with phrase-based statistical MT (PBSMT) systems [43], which were generally considered to be the state-of-the-art in MT before the advent of NMT [44]. Other TM-PBSMT integration methods have been proposed as well, for example by constraining the MT output to contain (preferably large) parts of translations retrieved from a TM [34,45]. This process involves identifying continuous strings of translated tokens to be 'blocked' in the MT output on the basis of statistical information about the alignment between tokens in the source and target sentences. Alternatively, the phrase tables used in PBSMT systems can be enriched with information retrieved from TM matches [13,46], or the decoding algorithms can be adapted to take into consideration matches retrieved from a TM [47,48]. All of these approaches were shown to lead to a significantly increased quality of MT output, especially in contexts where the amount of high-scoring TM matches retrieved is high.

Within the NMT paradigm various modifications to the NMT architecture and search algorithms were proposed to leverage information retrieved from a TM. For example, an additional encoder can be added to the NMT architecture specifically for TM matches [49] or, alternatively, a lexical memory [21]. The NMT system has also been adapted so it can have access to a selection of TM matches at the decoding stage [22]. Decoding algorithms have further been modified to incorporate lexical constraints [50] or to take into account rewards attached to retrieved strings of target language tokens, or 'translation pieces' [23], both of which can be informed by retrieving matches from a TM. More recently, a method has been proposed that augments the decoder of an MT system using token-level retrieval based on a nearest neighbour method, labeled *k*-nearest-neighbor machine translation (*k*NN-MT) [51]. A potential advantage of this method is that it does not rely on the identification of useful sentence-level matches, but rather finds relevant matches for each individual word, resulting in a wider coverage. On the other hand, preserving the syntactic and terminological integrity of close matches is one of the presumed advantages of many other TM-MT integration methods, which may, at least in part, be lost in this approach.

2.3. Integrating Fuzzy Matches into NMT through Data Augmentation

An easy-to-implement TM-NMT integration approach, labelled neural fuzzy repair (NFR), was proposed by [24]. Drawing on work in the field of automatic post-editing [52] and multi-source translation [53], this method consists of concatenating the target-language side of matches retrieved from a TM to the original sentence to be translated. As such, it only involves data pre-processing and augmentation, and is compatible with different NMT architectures.

In the original paper, the method was tested using a single fuzzy match metric (i.e., token-based edit distance) and seq2seq bidirectional RNN models with global attention. Different options were explored with regard to the number of concatenated matches, the amount of training data generated and the choice between training a 'dedicated' model for sentences for which high-scoring matches were retrieved only and a 'unified' model that deals with sentences with and without concatenated matches.

Tests carried out on two language combinations (English-Dutch, EN-NL, and English-Hungarian, EN-HU) using the TM of the European Commission's translation service [33] showed large gains in terms of various automated quality metrics (e.g., around +7.5 BLEU points overall for both language combinations when compared to the NMT baseline). It was observed that the increase in quality is almost linearly related to the similarity of the retrieved and concatenated matches, with quality gains in the highest match range (i.e., 90% match or higher) of around 22 BLEU points compared to the NMT baseline for both language combinations, compared to an increase of not even one BLEU point for EN-NL and around three BLEU points for EN-HU in the match range between 50% and 59%. The more matches that are retrieved, and the higher the similarity of the matches to the sentence to be translated, the more beneficial NFR is for translation quality.

The data augmentation approach to TM-NMT integration was further explored by [25], who tested the impact of different matching metrics, using Transformer models [54] for one language combination, English-French, and nine data sets. In addition to token-based edit distance, they tested n-gram matching and cosine similarity of sentence embedding vectors generated using sent2vec [55], as well as different combinations of matches retrieved using these matching methods. In addition, they incorporated information about the alignment of the target tokens for token-based edit distance and for n-gram matching by either removing unmatched tokens from the input or by additionally providing this information to the NMT architecture using word-level features. Their results are in line with those of [24], showing important increases in translation quality for all of the tested NFR methods in comparison to a baseline NMT system. Moreover, they confirm that the NFR approach is compatible with the Transformer NMT architecture. Their best-performing model concatenates both the best edit-distance match and the best sentence-embedding match to the input sentence, and adds information about the provenance and alignment of the tokens using factors (indicating whether tokens belong to the original source sentence, are aligned or unaligned edit-distance tokens, or belong to the sentence-embedding match). Finally, their results demonstrate that the NFR approach also leads to considerable quality gains in a domain adaptation context.

2.4. Other Related Research

The NFR approach is somewhat similar to the approach simultaneously proposed by [56] to incorporate terminology constraints into NMT, in that this method also introduces tokens in the target language in the source sentences, leading to bilingual inputs. In addition, in the context of this method, NMT models were shown to be flexible in dealing with such bilingual information and incorporating constraints, leading to improved MT output.

In addition, the so-called Levenshtein Transformer [57] is relevant in this context. This promising neural model architecture is aimed at learning basic edit operations (i.e., insertion and deletion), which makes it, in theory at least, especially suited for a task such as fuzzy match repair. Researchers have already used this architecture successfully to impose lexical constraints on NMT output [58].

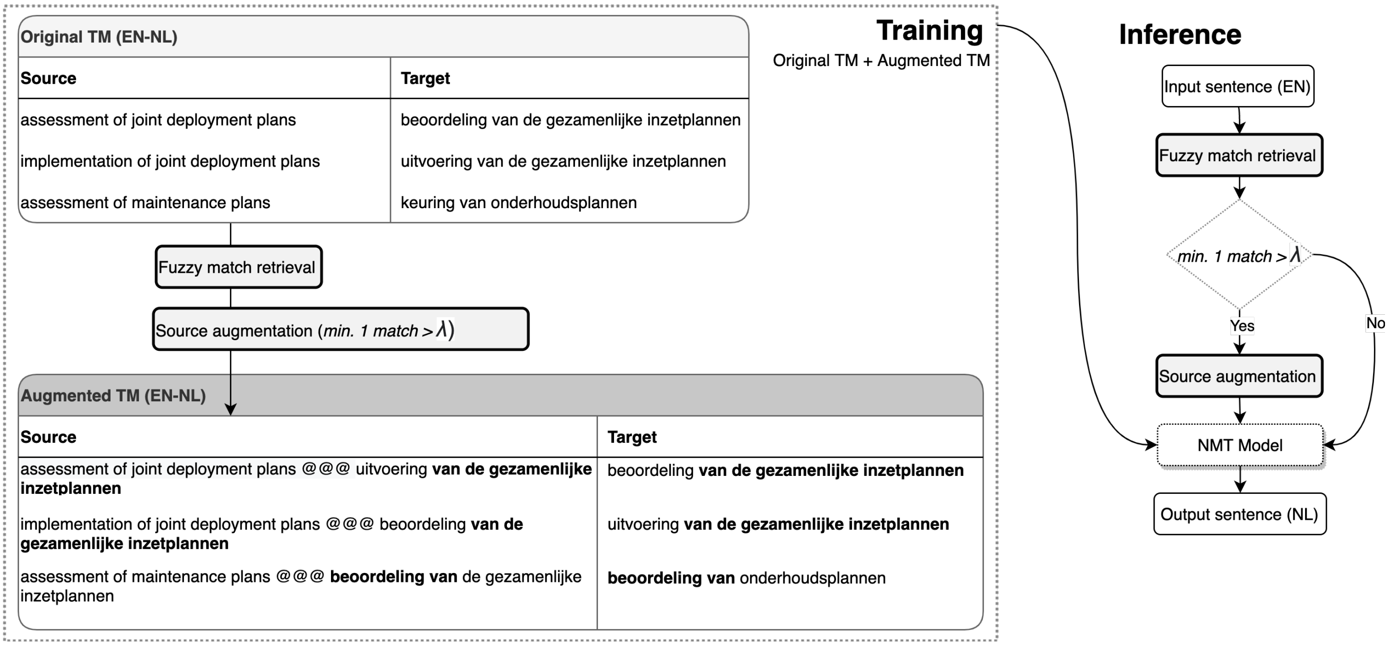
With regard to the incorporation of alignment information in NMT models, it should be noted that this has been attempted explicitly before, relatively quickly after the rise of NMT [59]. This integration of information from more traditional word alignment models was meant to combine the perceived advantages of the PBSMT and NMT paradigms, and thus differs from the incorporation of alignment information in the NFR paradigm.

Fuzzy match retrieval in combination with source–target concatenation has been shown to also be useful for improving the robustness of NMT models in the case of noisy data [60]. In addition, it was demonstrated to be a powerful method in the field of text generation as well [61].

3. Neural Fuzzy Repair: Methodology

In NFR, for a given TM consisting of source/target sentence pairs S, T, each source sentence $s_i \in S$ is augmented with the translations $\{t_1, \ldots, t_n\} \in T$ of n fuzzy matches $\{s_1, \ldots, s_n\} \in S$, where $s_i \notin \{s_1, \ldots, s_n\}$, given that the fuzzy match score is sufficiently high (i.e., above a given threshold λ). Previous research reported comparable NFR performance using 1, 2 and 3 concatenated fuzzy match targets [24]. In the experiments in this study, we use the translations of maximally two fuzzy matches for source augmentation ($n = 2$). We use "@@@" as the boundary between each sentence in the augmented source. The NMT model is then trained using the combination of the original TM, which consists of the original source/target sentence pairs S, T and the augmented TM, consisting of augmented-source/target sentence pairs S', T. At inference, each source sentence is augmented using the same method. If no fuzzy matches are found with a match score above λ, the non-augmented (i.e., original) source sentence is used as input. Figure 1 illustrates the NFR method for training and inference.

With the aim of improving the NFR method further, this paper explores a number of adaptations that involve sub-word level segmentation methods for fuzzy match retrieval (Section 3.1), the integration of (sub-)word alignment information (Section 3.2), and combinations of multiple fuzzy matches that extend source coverage (Section 3.3).



Original TM (EN-NL)

Source	Target
assessment of joint deployment plans	beoordeling van de gezamenlijke inzetplannen
implementation of joint deployment plans	uitvoering van de gezamenlijke inzetplannen
assessment of maintenance plans	keuring van onderhoudsplannen

Augmented TM (EN-NL)

Source	Target
assessment of joint deployment plans @@@ uitvoering **van de gezamenlijke inzetplannen**	beoordeling **van de gezamenlijke inzetplannen**
implementation of joint deployment plans @@@ beoordeling **van de gezamenlijke inzetplannen**	uitvoering **van de gezamenlijke inzetplannen**
assessment of maintenance plans @@@ **beoordeling van** de gezamenlijke inzetplannen	**beoordeling van** onderhoudsplannen

Figure 1. Neural fuzzy repair: training and inference.

3.1. Fuzzy Matching Using Sub-Word Units

Fuzzy matching is a key functionality in NFR, as the quality of the generated translations is determined by the similarity level of the retrieved fuzzy match(es) [24]. The original NFR method used token-based edit distance for fuzzy matching. With this similarity metric, the fuzzy match score $ED(s_i, s_j)$ between two sentences s_i and s_j is defined as:

$$ED(s_i, s_j) = 1 - \frac{EditDistance(s_i, s_j)}{max(|s_i|, |s_j|)} \tag{1}$$

where $|s|$ is the length of s. Following [24], candidates for high fuzzy matches are identified using `SetSimilaritySearch` (https://github.com/ardate/SetSimilaritySearch) before calculating edit distance.

It has been shown that matches extracted by measuring the similarity of distributed representations, in the form of sentence embeddings, can complement matching based on edit distance and lead to improvements in translation quality [25]. The sentence similarity score $SE(s_i, s_j)$ between two sentences s_i and s_j is defined as the cosine similarity of their sentence embeddings e_i and e_j:

$$SE(s_i, s_j) = \frac{e_i \cdot e_j}{\|e_i\| \times \|e_j\|} \tag{2}$$

where $\|e\|$ is the magnitude of vector e. Similar to [25], we use sent2vec [55] for training models from in-domain data, and build a FAISS index [62] containing the vector representation of each sentence for each NFR method and language. FAISS is a library specifically designed for efficient similarity search and vector clustering, and is compatible with the large data sets used in this study.

In a previous study, it was hypothesised that, while edit distance provides lexicalised matches from which the model learns to copy tokens to the MT output, matches obtained using sentence embeddings help the model to further contextualise translations, meaning both methods complement each other in the NFR context [25]. At the same time, however, it can be argued that both sentence similarity metrics rely on the information provided by surface forms of word tokens to measure similarity. This means that complex (inflectional) morphology can pose challenges to the retrieval of useful matches, especially when using edit distance as match metric. Using sub-word units might provide a useful way to mitigate such challenges. Even though sentence embeddings are already effective in capturing semantic similarities between vector representations of sentences using word tokens, sub-word units have been proven to be useful for building multilingual sentence embeddings [63], and were successfully utilised for the task of measuring sentence similarity [64].

In NLP, different sub-word segmentation approaches have been proposed with the aim of reducing data sparsity caused by infrequent words and morphological complexity, such as byte-pair encoding (BPE) [65], WordPiece [66], and linguistically motivated vocabulary reduction (LMVR) [67]. BPE was originally a data compression algorithm [68] before being used in the context of MT. It seeks an optimal representation of the vocabulary by iteratively merging the most frequent character sequences [65]. WordPiece uses the same approach for vocabulary reduction as BPE, with the difference that the merge choice is based on likelihood maximisation rather than frequency [69]. LMVR, on the other hand, is based on an unsupervised morphological segmentation algorithm that predicts sub-word units in a corpus by a prior morphology model, while reducing the vocabulary size to fit a given constraint [67].

Our hypothesis is that sub-word units can enable us to extract more relevant fuzzy matches, both when using exact, string-based matching algorithms (such as edit distance) and algorithms that utilise sentence embeddings. In this study, we first adapt both types of fuzzy matching approaches ED_{tok} and SE_{tok}, and replace word tokens with two types of sub-word units, namely, byte-pairs (ED_{bpe}, SE_{bpe}) and LMVR tokens (ED_{lmvr}, SE_{lmvr}). In our experiments, we use 32K merged vocabulary for source and target languages combined for the BPE implementation [65] (https://github.com/rsennrich/subword-nmt). As LMVR is based on language-specific morphological segmentation, we use 32K vocabulary for source and target languages separately for the LMVR implementation [67] (https://github.com/d-ataman/lmvr). Table 1 provides an example of fuzzy match retrieval using edit distance with word tokens, byte-pairs and LMVR tokens.

Table 1. Best fuzzy matches retrieved with edit distance using word tokens, byte-pairs and LMVR tokens for the Hungarian input sentence 'a görbületi sugarak közötti eltérések:', with the English translation 'differences between the radii of curvature:'. Matching (sub)word tokens between the input source (s_i) and the best fuzzy match source/target pair (s_j, t_j) are underlined.

	Score	$\boldsymbol{ED_{tok}}$
s_i		a görbületi sugarak közötti eltérések:
s_j	0.5	a nemek közötti egyenlőség:
t_j		gender equality:
		$\boldsymbol{ED_{bpe}}$
s_i		a görb@@ ületi sugar@@ ak közötti eltérések :
s_j	0.5	a tük@@ rö@@ k görb@@ ületi sugar@@ ai közötti eltérések
t_j		differences between the radii of curvature of mirrors
		$\boldsymbol{ED_{lmvr}}$
s_i		a gör @@b @@ület @@i sugar @@ak köz @@ött @@i eltérés @@ek:
s_j	0.71	a tükr @@ök gör @@b @@ület @@i sugar @@ai köz @@ött @@i eltérés @@ek
t_j		differences between the radii of curvature of mirrors

In the example in Table 1, both ED_{bpe} and ED_{lmvr} retrieve the same best fuzzy match with the translation 'differences between the radii of curvature of mirrors', which is different from the fuzzy match retrieved by ED_{tok}, with the translation 'gender equality:'. In this example, by utilising sub-word units, both ED_{bpe} and ED_{lmvr} retrieve a fuzzy match that is arguably more informative for the correct translation of the source sentence than ED_{tok}.

Due to the difference in the way sub-words are generated by ED_{bpe} and ED_{lmvr}, however, the two methods retrieve the same fuzzy match with different match scores (0.5 and 0.71, respectively).

3.2. Marking Relevant Words in Fuzzy Matches

Previous work shows that fuzzy matches with lower similarity scores may cause the NFR system to copy incorrect/unrelated text fragments to the output, leading to translation errors [24]. To mitigate this problem, source-side features have been utilised to mark relevant (and irrelevant) words in fuzzy target sentences extracted using ED_{tok} [25]. In a given TM consisting of source/target sentence pairs S, T, for a given source sentence $s_i \in S$ and fuzzy match source–target pair $s_j \in S$, $t_j \in T$, they first utilise LCS (Longest Common Sub-sequence) to mark the words in s_j that also exist in s_i. The marked words in s_j are then mapped to t_j using word alignment information between s_j and t_j. In this study, we follow this general idea and mark relevant (and irrelevant) words in fuzzy target segments t_j used to augment source sentences. In contrast to [25], we use the optimal path of edit distance to find overlapping, identical source and fuzzy source tokens (as seen in [70] (Chapter 3.11)) instead of LCS. We also use GIZA++ [71] for word alignment rather than fast_align [72]. Figure 2 illustrates how tokens are aligned between the input sentence, the fuzzy source and the fuzzy target.

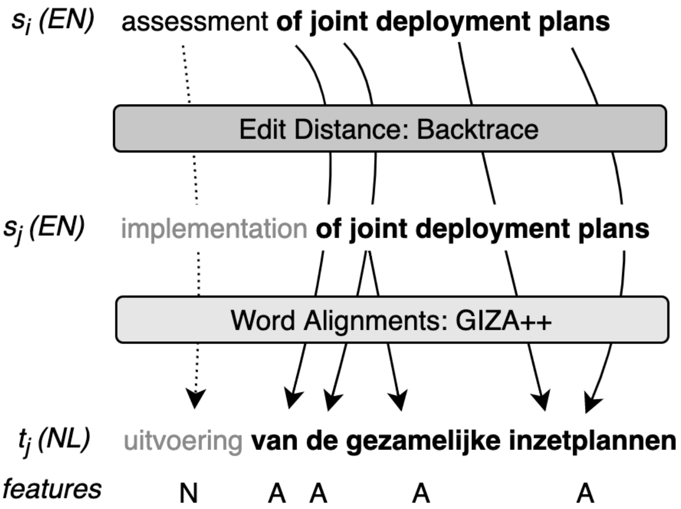

s_i (EN)
assessment of joint deployment plans
Edit Distance: Backtrace
s_j (EN)
implementation of joint deployment plans
Word Alignments: GIZA++
t_j (NL)
uitvoering van de gezamelijke inzetplannen
features
N A A A A

Figure 2. Aligning tokens of a source sentence (s_i) with tokens of source and target sentences for a fuzzy match (s_j, t_j). The features "*A*" and "*N*" stand for "*aligned*" (i.e., relevant) and "*non-aligned*" (i.e., irrelevant), respectively.

Reasoning that the similar sentences obtained with distributed representations do not necessarily present any lexical overlap, in a previous study this type of information was not added to t_j obtained by SE [25]. Rather, they add a dummy feature "E" to all words in the fuzzy target. We, however, hypothesise that such source side features can still assist the model with making better lexical choices, in case relevant words are present in similar sentences. Therefore, we also mark the relevant (and irrelevant) words in t_j obtained by SE. In this scenario, the focus is on high precision rather than on finding all semantically related words: by marking the words that occur in both s_j and t_j, we hope that the model learns to recognise these words as high-fidelity candidates to copy to the target side.

Besides word-level tokens, we use BPE and LMVR units for fuzzy matching (as described in Section 3.1). In this case, we follow the approach described above to mark relevant (and irrelevant) sub-word units, but use GIZA++ on sub-word-level data, which has been shown to lead to an improved alignment quality [73].

3.3. Maximum Source Coverage

As discussed in the previous section, features that mark relevant words in fuzzy match target sentences can provide additional information to the NFR model. Moreover, these features allow us to better combine fuzzy matches so that they provide complementary information about the source sentence. Existing methods on combining multiple fuzzy matches in the context of NFR, which lead to improvements over using a single fuzzy match for data augmentation, either use n-best fuzzy matches obtained by a single fuzzy matching method or the best matches obtained by n different fuzzy matching methods. Neither of these approaches guarantee that more source information (i.e., words) is covered by the combined fuzzy matches.

In this study, we propose an alternative fuzzy match combination method, *max_coverage*. To combine two fuzzy target sentences for data augmentation, this method first retrieves the best fuzzy source–target pair s_1, t_1 obtained for a given source sentence s_i. As a second match, this method seeks a source–target pair s_j, t_j where s_j maximises the coverage of source words in s_i when combined with s_1. We limit the search for the second match to the best 40 matches with match score above 0.5. If no such match is found, the algorithm falls back to using 2-best matches. To calculate the source coverage of a given t_j, we use the methodology described in Section 3.2. Algorithm 1 shows the pseudo-code for *max_coverage*. Table 2 illustrates the different approaches to using features on fuzzy target sentences and combining fuzzy matches, including *max_coverage*.

Algorithm 1: Pseudo-code for *max_coverage*.

```
function max_coverage ($s_i, S, T$);
Input  : Source sentence $s_i$, list of fuzzy match source–target pairs $S, T$ with fuzzy match score above the
         threshold $\lambda$, where $s_i \notin S$
Output : List of fuzzy match source–target pairs $S_{maxc}, T_{maxc}$
$S_{maxc} \leftarrow [s_1]$, where $s_1 \in S$ is the source segment of the highest-scoring fuzzy match;
$T_{maxc} \leftarrow [t_1]$, where $t_1 \in T$ is the target segment of the highest-scoring fuzzy match;
$C \leftarrow$ list of token IDs in $s_i$, which are aligned with tokens in $s_1$;
$extra\_coverage \leftarrow 0$;
$s_{new} \leftarrow None$;
$t_{new} \leftarrow None$;
for $(s_j, t_j)$ in $(S, T)$ do
    $C_{cand} \leftarrow$ list of token IDs in $s_i$, which are aligned with tokens in $s_j$;
    $C_{union} \leftarrow C \cup C_{cand}$;
    if $|C_{union}| - |C| > extra\_coverage$ then
        $s_{new} \leftarrow s_j$;
        $t_{new} \leftarrow t_j$ ;
        $extra\_coverage \leftarrow |C_{union}| - |C|$;
    end
end
if $s_{new} \neq None$ and $t_{new} \neq None$ then
    $S_{maxc}$.append($s_{new}$);
    $T_{maxc}$.append($t_{new}$);
else
    $S_{maxc}$.append($s_2$), where $s_2 \in S$ is the source segment of the fuzzy match with the second highest match score;
    $T_{maxc}$.append($t_2$), where $t_2 \in T$ is the target segment of the fuzzy match with the second highest match score;
end
return $S_{maxc}, T_{maxc}$
```

Table 2. Adding feature labels to best fuzzy match targets retrieved for the Hungarian input source 'orvosi fizikus szakértők .' (EN: 'medical physics experts .'). The feature labels "*A*" and "*N*" indicate target tokens that are aligned/not aligned to tokens in the input source. Label "*E*" is a dummy feature used for correct formatting. Target tokens that are aligned to the input source through features are marked in bold.

Match Rank/Type	Score	Best Fuzzy Match Targets
		2-Best ED, without Features (Bulté & Tezcan, 2019)
$1 - ED_{tok}$	0.5	medical equipment.
$2 - ED_{tok}$	0.5	to receive medical treatment
		Best ED + Best SE (Xu et al., 2020)
$1 - ED_{tok}$	0.5	**medical | A** equipment | N **. | A**
$1 - SE_{tok}$	0.862	to | E receive | E medical | E treatment | E
		2-best SE
$1 - SE_{tok}$	0.862	to | N receive | N **medical | A** treatment | N
$2 - SE_{tok}$	0.829	**medical | A** equipment | N **. | A**
		Best SE + $max\_coverage$ SE
$1 - SE_{tok}$	0.862	to | N receive | N **medical | A** treatment | N
$2 - SE_{tok}$	0.765	**medical | A physics | A** expert | N

In Table 2, the only second fuzzy match target that increases the number of words covered in the source sentence is retrieved by *max_coverage* (*Best SE + max_coverage SE*). Despite its lower match score compared to the second best match (0.829 vs. 0.765), the English sentence *'medical physics expert'* provides additional information about the translation for the source word *'fizikus (physics)'*.

4. Experimental Setup

This section describes the data sets that are used in the experiments (Section 4.1), the NFR models, and the baselines they are compared to (Section 4.2), as well as the procedures used for evaluation (Section 4.3).

4.1. Data

As the data set, we use the TM of the European Commission's translation service [33]. All sentence pairs were truecased and tokenised using the Moses toolkit [74]. We run detailed tests of different NFR configurations on one language pair, English ↔ Hungarian (EN-HU, HU-EN). The best systems are then tested for three further language pairs: English ↔ Dutch (EN-NL, NL-EN), English ↔ French (EN-FR, FR-EN), and English ↔ Polish (EN-PL, PL-EN).

For each language combination, we used ∼2.4 M sentences as training, ∼3 K sentences as validation, and ∼3.2 K sentences as the test set. The validation and test sets did not contain any sentences that also occurred in the training set (i.e., 100% matches were removed). Our aim was to keep the validation and test sets as constant as possible across language combinations, but some sentence pairs needed to be removed from the test set when the translation direction was switched since the source sentences occurred in the corresponding training set, leading to slightly different test sets for the different language combinations. The exact number of sentences for each language combination, before data augmentation was applied, is provided in Appendix A.

4.2. Baseline and NFR Models

Table 3 provides an overview of the NFR system configurations that are tested in this study for English ↔ Hungarian, alongside the baselines they are compared to. All systems, including the four baselines, make use of the Transformer NMT architecture (see

Appendix B.1 for a detailed description). For training the NMT models, BPE is applied to all data, except when LMVR is used for sub-word segmentation prior to retrieving fuzzy matches. It should be noted that, for training the NMT models, the sub-word segmentation step is applied independently from the segmentation techniques used for fuzzy match retrieval (see Section 3.1). This distinction is also made in Table 3, where these steps are referred to as 'NMT unit' and 'Match unit', respectively. All NFR systems use source sentences augmented with two retrieved fuzzy matches, at most, and all but one of the NFR systems use 0.5 as a threshold for fuzzy match retrieval.

The first baseline, *Baseline Transformer*, is the only system not using the NFR method. As a second baseline, we apply the originally proposed NFR data augmentation method, involving token-based fuzzy matching using edit distance [24], but implemented using the Transformer architecture and BPE, instead of BRNNs and tokens (ED_{tok}). Finally, we report the performance of two systems that implement the best NFR configuration proposed by [25], involving the combination of one edit-distance and one sentence-embedding match using token-based matching, with alignment features added to the first match, but not to the second. The first of these two baselines uses the thresholds for fuzzy matching used by [25], namely 0.6 for edit distance and 0.8 for sentence embeddings (ED_{tok}/SE_{tok}), whereas the second uses the same threshold of 0.5 as the other NFR systems in this study ($ED_{tok}/SE_{tok}{}^{\ddagger}$). For both variants of this system, we used `SetSimilaritySearch` to extract high fuzzy match candidates prior to calculating ED as described in Section 3.1.

Table 3. Summary of the systems evaluated in this study for English ↔ Hungarian.

Baseline Systems	**Match Threshold**	**Match Method**	**Match Unit**	**NMT Unit**	**Alignment Features**	**Maximum Coverage**
Baseline Transformer				BPE		
ED_{tok}	0.5	ED	tok	BPE	−	−
ED_{tok}/SE_{tok}	0.6/0.8	ED/SE	tok	BPE	+/−	−
$ED_{tok}/SE_{tok}{}^{\ddagger}$	0.5	ED/SE	tok	BPE	+/−	−
Tested systems						
ED_{bpe}	0.5	ED	BPE	BPE	−	−
ED_{lmvr}	0.5	ED	LMVR	LMVR	−	−
SE_{tok}	0.5	SE	tok	BPE	−	−
SE_{bpe}	0.5	SE	BPE	BPE	−	−
SE_{lmvr}	0.5	SE	LMVR	LMVR	−	−
$SE_{tok}+$	0.5	SE	tok	BPE	+	−
$SE_{bpe}+$	0.5	SE	BPE	BPE	+	−
$SE_{lmvr}+$	0.5	SE	LMVR	LMVR	+	−
$SE_{tok}+M$	0.5	SE	tok	BPE	+	+
$SE_{bpe}+M$	0.5	SE	BPE	BPE	+	+
$SE_{lmvr}+M$	0.5	SE	LMVR	LMVR	+	+

The tested NFR systems are divided into three blocks: the first group of systems is implemented without alignment features or maximum source coverage, the second group uses alignment features only, and the third uses both alignment features and maximum source coverage. The labels that are used for each tested system represent the fuzzy matching method (either edit distance, *ED*, or sentence embeddings, *SE*) and the fuzzy match unit (word-level tokens, *tok*; byte-pair encoding, *bpe*; or linguistically motivated vocabulary reduction, *lmvr*), show whether source-side alignment features are used (indicated with +)

and whether maximum source coverage is applied (indicated with M). In the case of SE, 'match unit' refers to the (sub)word unit that was used to train the sent2vec model that, in turn, was used to extract sentence embeddings, which were used for fuzzy match retrieval. For the systems that utilise word-level tokens as a 'match unit' but sub-word units as the 'NMT unit', alignment features are mapped from tokens to their corresponding sub-word units.

The hyper-parameters and the training details of the NMT systems, sent2vec, FAISS, LMVR, and GIZA++ are provided in Appendix B.

4.3. Evaluation

We make use of automated evaluation metrics BLEU [75] (From Moses: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl), TER [30] (Version 0.7.25: https://github.com/snover/terp), and METEOR [76] (Version 1.5: https://www.cs.cmu.edu/~alavie/METEOR/) to assess the quality of the translations. Bootstrap resampling tests are performed to verify whether differences between the best baseline system and the NFR systems combining all tested modifications are statistically significant [77]. In addition to evaluations on the complete test set, we also carry out more fine-grained evaluations of subsets of the test set defined on the basis of input sentence length and fuzzy match score.

5. Results

First, we present a detailed analysis of the results for English ↔ Hungarian (Section 5.1). Section 5.2 shows the results for the three other language pairs, using the best NFR systems identified for English ↔ Hungarian. Finally, Section 5.3 focuses on the impact of fuzzy match score and sentence length on the quality of the generated translations.

5.1. Detailed Evaluation for English-Hungarian and Hungarian-English

Table 4 shows the results of the automated evaluations for the four baseline systems and all NFR systems that were tested for EN-HU and HU-EN. The table consists of four sections. From top to bottom, it shows the results for (a) the baseline systems, (b) the NFR systems using different match methods and match units, (c) the systems that incorporate alignment features, and (d) those that use both alignment features and maximum source coverage. The table reports BLEU, TER, and METEOR scores, but, in the text, we mainly focus on BLEU.

For both translation directions, *Baseline Transformer*, which does not make use of the NFR augmentation method, is outperformed by all NFR baselines by 4.5 to 6.67 BLEU points. For both EN-HU and HU-EN, the strongest baseline is the system combining the best ED and SE matches, with a threshold of 0.5 ($ED_{tok}/SE_{tok}{}^{\ddagger}$). This configuration also outperforms the same system that uses higher thresholds for ED and SE matches, 0.6 and 0.8, respectively (ED_{tok}/SE_{tok}).

The results of our first set of experiments, targeting different fuzzy matching methods and units, show that the systems using matches retrieved by SE outperform their counterparts that use ED as match metric (SE_{tok} vs. ED_{tok}, SE_{bpe} vs. ED_{bpe}, and SE_{lmvr} vs. ED_{lmvr}). In this context, we note that, with ED, the number of retrieved fuzzy matches above the 0.5 threshold was consistently and substantially lower than with SE. Moreover, for HU-EN, the system combining the two highest-scoring SE matches using byte-pairs (SE_{bpe}) already outperforms the baseline system that combines the best ED and SE matches ($ED_{tok}/SE_{tok}{}^{\ddagger}$), which additionally includes alignment features. Comparing the different fuzzy match units, for both translation directions, the systems that use LMVR-based matching perform worst (for both ED and SE). Token-based matching, in contrast, appeared to work better, but not as well as BPE. Especially for HU-EN, SE_{bpe} scored considerably higher than SE_{tok} (+1.05 BLEU points). For EN-HU, the increase in quality was less pronounced (+0.22 BLEU).

Table 4. Results of the automated evaluations for English ↔ Hungarian. Match% refers to the percentage of test sentences augmented with matches. Best scores are highlighted per table section (underlined for baselines, bold for tested systems).

	English-Hungarian				Hungarian-English			
	BLEU	**TER**	**MET.**	**Match%**	**BLEU**	**TER**	**MET.**	**Match%**
Baseline Transformer	46.78	40.01	63.51	-	57.63	30.74	44.93	-
ED_{tok}	52.22	36.44	67.17	58%	62.30	27.81	46.80	50%
ED_{tok}/SE_{tok}	52.83	35.77	67.66	65%	63.20	27.35	47.15	65%
$ED_{tok}/SE_{tok}{}^{\ddagger}$	<u>53.45</u>	35.29	68.19	99%	<u>64.06</u>	26.39	47.55	99%
ED_{bpe}	52.27	36.54	67.15	56%	62.19	28.12	46.67	47%
ED_{lmvr}	50.64	38.46	65.38	55%	60.53	29.05	45.79	48%
SE_{tok}	52.89	35.89	67.71	99%	64.17	26.53	47.49	99%
SE_{bpe}	**53.11**	**35.79**	**67.86**	96%	**65.22**	**25.86**	**48.04**	98%
SE_{lmvr}	51.72	36.92	66.77	99%	63.43	26.89	47.04	99%
SE_{tok}+	53.52	35.27	68.26	99%	64.41	26.48	47.76	99%
SE_{bpe}+	**53.63**	**35.16**	**68.34**	96%	**65.03**	**26.05**	**47.90**	98%
SE_{lmvr}+	52.09	36.28	67.06	99%	63.39	27.16	46.98	99%
SE_{tok}+M	**53.83**	**34.81**	**68.60**	99%	65.44	25.65	48.19	99%
SE_{bpe}+M	53.62	35.22	68.34	96%	**65.75**	**25.40**	**48.28**	98%
SE_{lmvr}+M	52.34	36.17	67.22	99%	63.97	26.78	47.23	99%

Given the consistent improvements yielded by SE over ED, for the next set of experiments, we focused on systems combining sentence embedding matches only. As the third section of Table 4 shows, adding features based on alignment information resulted in a small but consistent improvement in quality for EN-HU (between +0.37 and +0.63 BLEU points compared to the corresponding systems without features). For HU-EN, however, this was not the case. Whereas the performance was slightly better for the token-based system with added features (SE_{tok}+) compared to the system without (+0.24 BLEU), the opposite was the case for the systems using BPE and LMVR as match unit (−0.19 BLEU for SE_{bpe}+ and −0.04 for SE_{lmvr}+).

Next, we apply *max_coverage* to all three systems that use SE and alignment-based features. For EN-HU, this resulted in an improved performance in two out of three cases (+0.31 BLEU for SE_{tok}+M and +0.4 for SE_{lmvr}+M). For SE_{bpe}+M, the estimated quality remained virtually identical (−0.01 BLEU). This results in SE_{tok}+M scoring best overall for EN-HU, according to the three evaluation metrics. Compared to the best baseline ($ED_{tok}/SE_{tok}{}^{\ddagger}$), the difference is +0.38 BLEU ($p < 0.05$), −0.48 TER, and +0.41 METEOR.

For HU-EN, retrieving matches using *max_coverage* improved the system's performance on all occasions, both compared to the systems with and without alignment-based features. The differences compared to the systems with features were larger than for EN-HU (between +0.58 and +1.03 BLEU), but, at the same time, most HU-EN systems including features scored slightly worse than those without. This means that, for HU-EN, the best-scoring system overall was the system including fuzzy matching based on sentence embeddings for BPE, including alignment features and maximum source coverage (SE_{bpe}+M). Compared to the best baseline ($ED_{tok}/SE_{tok}{}^{\ddagger}$), its performance was estimated to be better on all evaluation metrics (+1.69 BLEU, $p < 0.001$; −0.99 TER; +0.73 METEOR).

In a final step, we retrained the two best-scoring systems (SE_{tok}+M and SE_{bpe}+M) without alignment features, considering that adding such features did not appear to be beneficial in HU-EN (SE vs. SE+ in Table 4). The results of these tests (see Table 5) show that, for both language combinations, both systems with added features score better than their counterparts without (between −0.20 and −0.95 BLEU).

Table 5. Results of the automated evaluations for two additional systems without alignment features and with maximum coverage for English ↔ Hungarian.

	English-Hungarian			**Hungarian-English**		
	BLEU	**TER**	**MET.**	**BLEU**	**TER**	**MET.**
SE_{tok}-M	53.58	35.33	68.15	64.49	26.59	47.55
SE_{bpe}-M	53.31	35.74	67.95	65.55	25.55	48.13

5.2. Evaluation on Three Additional Language Pairs

In this section, we present the results of the comparisons between the best NFR configurations identified for English ↔ Hungarian and the baseline systems on three additional language pairs: English ↔ Dutch, English ↔ French, and English ↔ Polish. We only report the three baselines using the same thresholds for fuzzy matching (i.e., 0.5). The best NFR systems for English ↔ Hungarian were those using fuzzy matching based on sentence embeddings, including alignment-based features and maximum source coverage. One system uses tokens as match unit (SE_{tok}+M), the other BPE (SE_{bpe}+M).

Table 6 shows the results for English ↔ Dutch. Compared to *Baseline Transformer*, the NFR system using token-based edit-distance matching (ED_{tok}) already performs considerably better for both translation directions (+4.96 BLEU for EN-NL, and +5.4 BLEU for NL-EN). In both cases, however, the best baseline is the system including one sentence-embedding and one edit-distance match ($ED_{tok}/SE_{tok}{}^{\ddagger}$), with an additional +1.92 BLEU for EN-NL and +0.90 for NL-EN. Compared to the best baseline, the two NFR configuration tested here score better according to all evaluation metrics for both EN-NL and NL-EN. For EN-NL, our best system (SE_{bpe}+M) scores +0.54 BLEU better than the best baseline ($p < 0.01$). For NL-EN, the best system (SE_{tok}+M) outperforms the best baseline by +0.86 BLEU ($p < 0.001$).

Table 6. Results for English ↔ Dutch. Statistical significance of the improvements in BLEU is tested against the strongest baseline scores, which are underlined (**: $p < 0.01$; ***: $p < 0.001$). Best scores for the tested systems are highlighted in bold.

Model	**English-Dutch**			**Dutch-English**		
	BLEU	**TER**	**MET.**	**BLEU**	**TER**	**MET.**
Baseline Transformer	56.04	32.54	73.21	61.38	27.39	46.85
ED_{tok}	61.00	29.17	76.06	66.78	24.05	49.06
$ED_{tok}/SE_{tok}{}^{\ddagger}$	<u>62.92</u>	27.41	77.46	<u>67.68</u>	23.04	49.43
SE_{tok}+M	63.21	27.36	77.57	**68.54** ***	**22.67**	49.76
SE_{bpe}+M	**63.46** **	**27.29**	**77.70**	68.47 **	22.69	**49.82**

The results for English ↔ French are presented in Table 7. The pattern that emerges is fairly similar to the one observed for English ↔ Dutch. ED_{tok} outperforms *Baseline Transformer* with +3.61 BLEU for EN-FR and +5.20 BLEU for FR-EN, but the best baseline is $ED_{tok}/SE_{tok}{}^{\ddagger}$, scoring +1.35 BLEU higher than the second-best baseline for EN-FR and +1.03 BLEU for FR-EN. For both translation directions, the best NFR system is SE_{bpe}+M. Compared to the best baseline, this system achieves +0.93 BLEU for EN-FR ($p < 0.001$) and +1.07 BLEU for FR-EN ($p < 0.001$).

Table 7. Results for English ↔ French. Statistical significance of the improvements in BLEU are tested against the strongest baseline scores, which are underlined (**: $p < 0.01$; ***: $p < 0.001$). Best scores for the tested systems are highlighted in bold.

Model	**English-French**			**French-English**		
	BLEU	**TER**	**MET.**	**BLEU**	**TER**	**MET.**
Baseline Transformer	60.80	29.53	73.64	61.81	26.91	47.08
ED_{tok}	64.41	27.18	75.93	67.01	23.44	49.26
$ED_{tok}/SE_{tok}{}^{\ddagger}$	<u>65.76</u>	26.06	76.90	<u>68.04</u>	22.82	49.68
SE_{tok}+M	66.45 **	25.72	77.39	68.83 **	22.31	50.09
SE_{bpe}+M	**66.69** ***	**25.48**	**77.51**	**69.11** ***	**22.08**	**50.24**

Finally, Table 8 shows the results for English ↔ Polish. Generally speaking, these results are in line with those observed for the other language combinations that were tested. In addition, here, $ED_{tok}/SE_{tok}{}^{\ddagger}$ is the strongest baseline, scoring +1.09 BLEU better than ED_{tok} for EN-PL and +2.04 for PL-EN. ED_{tok}, in turn, outperformed *Baseline Transformer* with +3.96 BLEU for EN-PL and +4.81 for PL-EN. For both translation directions, SE_{bpe}+M performs best, improving over the best baseline by a further +1.16 BLEU for EN-PL ($p < 0.001$) and +1.58 BLEU for PL-EN ($p < 0.001$).

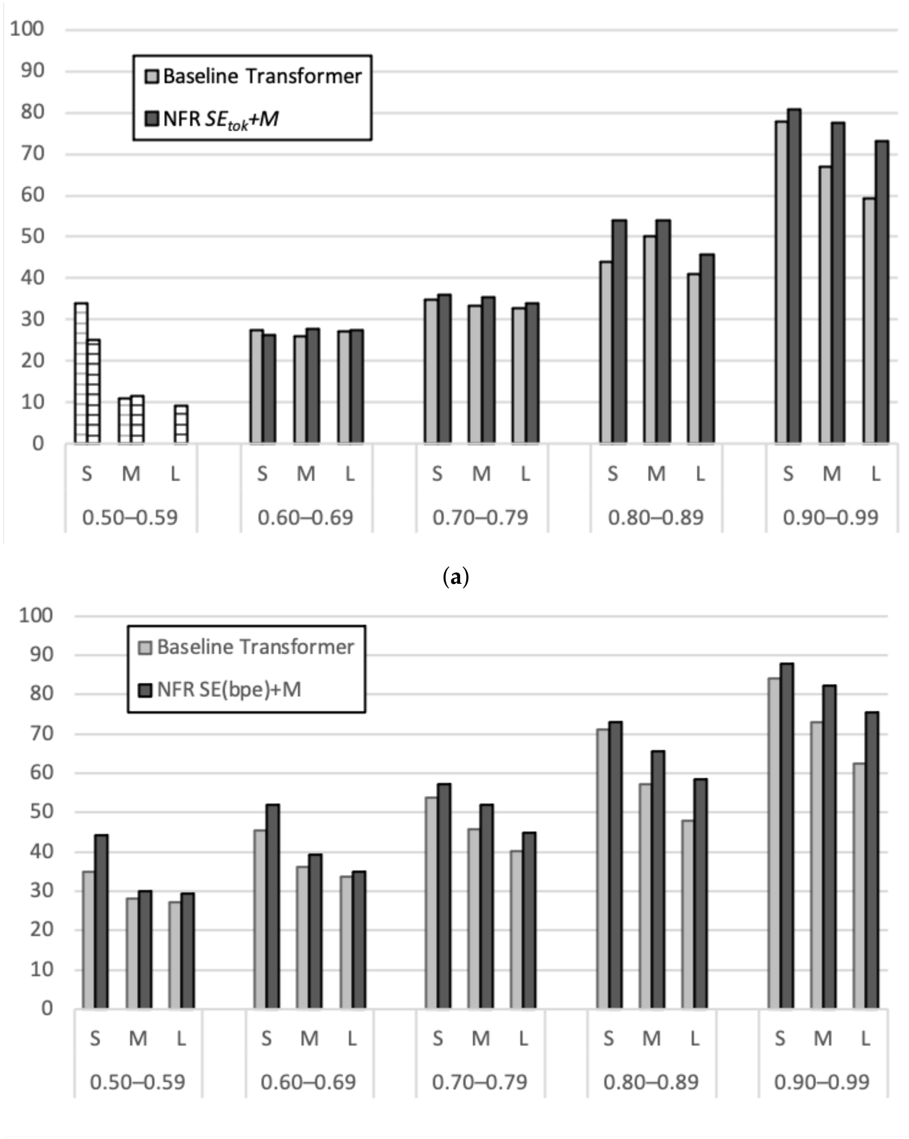
Table 8. Results for English ↔ Polish. Statistical significance of the improvements in BLEU are tested against the strongest baseline scores, which are underlined (**: $p < 0.01$; ***: $p < 0.001$). Best scores for the tested systems are highlighted in bold.

Model	**English-Polish**			**Polish-English**		
	BLEU	**TER**	**MET.**	**BLEU**	**TER**	**MET.**
Baseline Transformer	52.13	35.77	38.24	59.16	30.24	45.48
ED_{tok}	56.09	33.11	40.15	63.97	27.32	47.38
$ED_{tok}/SE_{tok}{}^{\ddagger}$	<u>57.18</u>	32.21	40.76	<u>66.01</u>	25.77	48.28
SE_{tok}+M	58.24 ***	31.68	41.32	66.86 **	25.21	48.67
SE_{bpe}+M	**58.34** ***	**31.50**	**41.36**	**67.59** ***	**24.55**	**49.05**

5.3. Impact of Match Score and Sentence Length on Translation Quality

To obtain more insight into the performance of the NFR systems, we evaluate the impact of two variables that can influence the quality of the generated translations: the length of the input sentence [78], and the degree of similarity between the retrieved fuzzy matches and the input sentence [24]. For the purpose of this analysis, we focus once more on the language pair English ↔ Hungarian. This type of analysis can be informative for determining the most appropriate value for the threshold λ (i.e., the lower bound for the fuzzy matching score), and it may tell us whether data augmentation is beneficial or not for input sentences of a certain length.

Figures 3 and 4 plot the BLEU scores on the test set for each of the bins defined by input sentence length (calculated on the basis of word tokens) and fuzzy match score, for (a) SE_{tok}+M and (b) SE_{bpe}+M, compared to the *Baseline Transformer*, for EN-HU and HU-EN, respectively. Appendix C contains an overview of the sizes of the corresponding bins for both language combinations. Considering that the bin sizes for SE_{tok}+M in the lowest match range are very small, we exclude these bins from the interpretation of the analysis. We also note that the distribution of similarity scores was markedly different for BPE and token-based matches, with the latter being concentrated in the higher match ranges.



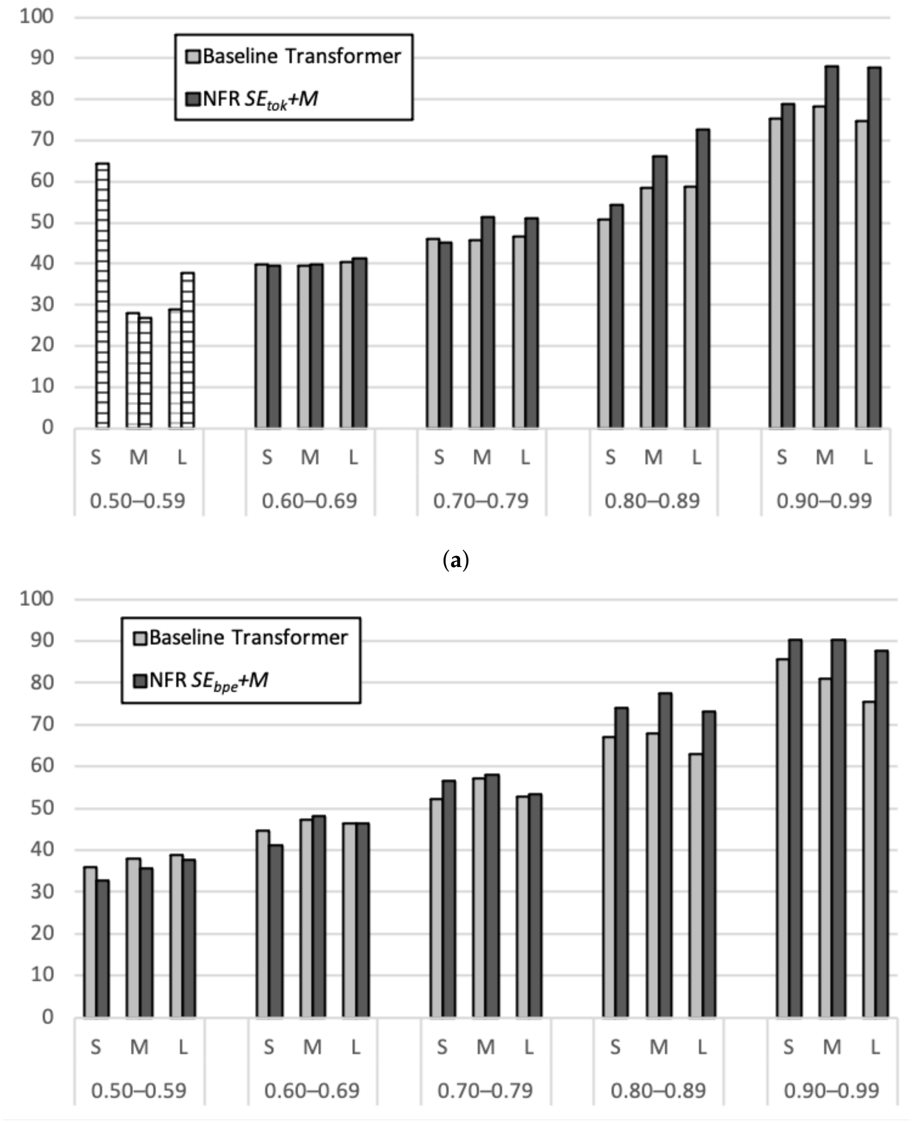
(**a**)

(**b**)

Figure 3. Comparison of (**a**) *SE_{tok}+M* and (**b**) *SE_{bpe}+M* with *Baseline Transformer* for different match ranges and sentence lengths, EN-HU. Note that in (**a**) the bin sizes in the lowest match range are very small (i.e., between 3 and 9 sentences), so the reported BLEU scores for these bins are not reliable. S refers to sentences of length 1–10, M 11–25, and L over 25.

Even though Figure 3 shows a slightly different picture for both systems using different match units, some trends can be observed: (a) BLEU scores increase with increasing match scores, (b) the added value of the NFR approach becomes greater in higher match ranges, (c) from a match score of 0.7 onward, for each comparison the NFR system outperforms *Baseline Transformer*, and (d) short sentences score, overall, higher than longer ones. Whereas *SE_{bpe}+M* outperforms *Baseline Transformer* in every bin (i.e., above 0.5), for *SE_{tok}+M*, this is not necessarily the case for matches scoring lower than 0.7.

The picture for HU-EN, shown in Figure 4, is in some respects similar to what we observed for EN-HU. In addition, we have (a) the scores for both systems increase with increasing match scores, and (b) the higher the match range, the greater the positive impact of the NFR approach. However, different from EN-HU, (c) the NFR systems only consistently outperform *Baseline Transformer* with match score of 0.8 or higher, and (d) shorter input sentences do not necessarily lead to higher translation quality. For *SE_{bpe}+M*, *Baseline Transformer* outperforms the NFR system for all bins in the lowest match range (i.e., 0.50–0.59). In the match range 0.60–0.69, this is the case for two out of three bins.



(**a**)

(**b**)

Figure 4. Comparison of (**a**) SE_{tok}+M and (**b**) SE_{bpe}+M with *Baseline Transformer* for different match ranges and sentence lengths, HU-EN. Note that, in (**a**), the bin sizes in the lowest match range are very small (i.e., between 3 and 5 sentences), so the reported BLEU scores for these bins are not reliable. S refers to sentences of length 1–10, M 11–25, and L over 25.

6. Discussion

Our detailed analyses for English ↔ Hungarian show that retrieving and ranking fuzzy matches using sentence embeddings leads to better results than edit distance or a combination of both methods. This confirms the usefulness of fuzzy matching based on distributed representations that capture semantic relationships between (sub-)words, in contrast to exact, string-based fuzzy matching [25,32]. One striking difference between *ED* and *SE* in our experiments is the percentage of input sentences for which a fuzzy match was retrieved: whereas almost all sentences in the training (as well as the test) set were augmented with a fuzzy match target using *SE*, with *ED*, this was only the case for around half of the sentences. In this study, we used a fuzzy match threshold of 0.5 for both *ED* and *SE*. Even though lowering the threshold for fuzzy matching in the case of *ED* could increase the proportion of input sentences for which a match is retrieved and data augmentation is performed, retrieved sentences that are (formally speaking) too dissimilar to or do not show enough overlap with the original sentence can lead to a decrease in translation quality [24]. It thus seems that *SE* is able to retrieve more matches that are informative in the NFR context. This may, in part, be related to the fact that matching based on SE is less likely to retrieve (semantically) unrelated sentences than matching based on ED, especially for short sentences (e.g., because of coincidental overlap in function words or punctuation). We return to the issue of the fuzzy matching threshold below.

We also found that, generally speaking, fuzzy matching based on BPE performed slightly better than token-based matching, but this was not always the case. For example, the best-scoring NFR system for HU-EN used tokens as match unit, which is why we also tested NFR systems using token-based matching for the other language pairs. If we look across language combinations, on six out of eight occasions, the NFR system using BPE as match unit outperformed the system relying on tokens (the two exceptions being EN-HU and NL-EN). However, the differences between systems that use tokens and BPE were relatively small, never exceeding 1 BLEU point. The systems using LMVR did not perform as well as the ones using tokens and BPE. Since sub-word units were both used as matching and NMT unit, we cannot claim with certainty whether this lower performance is due to the performance of LMVR as a match unit or as an NMT unit. In this context, it is worth noting that previous research has shown that LMVR could lead to better translation quality than BPE for morphologically rich source languages [67].

With regard to features based on alignment information, adding such features improved the performance of all NFR systems for EN-HU, but not for HU-EN. For HU-EN, adding features was only beneficial when tokens were used as match unit, and when features were added in combination with maximum source coverage. These findings show that adding alignment-based features to matches retrieved using SE is potentially beneficial, and that providing such information should maybe not be restricted to fuzzy matches retrieved using ED or other exact string-based similarity metrics [25].

Across EN-HU and HU-EN, applying $max\_coverage$ in addition to features led to improvements in estimated translation quality for five out of six tested systems. The increase in BLEU scores was, however, more pronounced with Hungarian as a source language. Moreover, $max\_coverage$ also led to improvements in BLEU scores for systems without alignment features. To better understand the impact of applying this algorithm, we verified the proportion of sentences in the training set that was affected by it. $max\_coverage$ only affects the second fuzzy match target that is used for source augmentation (i.e., not selecting the second highest-scoring match). For both EN-HU and HU-EN and for all systems, this was the case for between 68% and 77% of sentences for which data augmentation was performed. This means that, for a majority of sentences, the fuzzy match with the second highest match score was not the most informative for the NFR system, which shows that match score should not be considered to be the only criterion for selecting useful matches for data augmentation.

Even though the tested modifications to the NFR system individually only led to small and at times inconsistent improvements to the estimated translation quality for EN-HU and HU-EN, in combination, they resulted in significant improvements over the best baseline system for all tested language combinations. The reported improvements were also not equally large for each language combination, with the most substantial improvements recorded for HU-EN (+1.69 BLEU) and PL-EN (+1.58), and the least for EN-NL (+0.54) and EN-HU (+0.38). It thus seems that the proposed modifications work best for source languages that are morphologically rich (i.e., Hungarian and Polish in our experiments), but this tentative conclusion would need to be confirmed in subsequent studies.

On the topic of the threshold that is used for fuzzy match retrieval, our analyses show that finding the optimal value for this threshold not only depends on the match metric, but also on the type of fuzzy match unit that is used and on the (source) language. In this context, we also pointed out that there were considerable differences between the distribution of similarity scores for BPE and token-based matches. We therefore argue that this threshold should be considered a tunable parameter of the NFR system, rather than a fixed value. According to our preliminary tests, the optimal value for this parameter varies between 0.5 and 0.7, but in a previous study, for example, a threshold of 0.8 for

SE matching was applied [25]. Note that, for our data set and language combinations, however, a threshold of 0.5 led to better results for the system configuration used in that study. It thus seems plausible that the optimal threshold also varies between different data sets and domains. With regard to input sentence length, this factor had a clear impact on overall translation quality, but, according to our experiments, it did not seem beneficial to apply a filter to this parameter (in combination with fuzzy match score) for the purpose of NFR.

The experiments presented in this paper are limited in a number of ways. First, we only used one data set, albeit with multiple language pairs, and a single domain. Second, it was not possible to test all combinations of NFR configurations because of the high computational cost involved in fuzzy match retrieval for large data sets (when using sentence embeddings, with FAISS indexing, this operation takes approximately 48 hours for each training set on a single Tesla V100 GPU) and training Transformer models with training sets that are further enlarged (up to approximately twice the original size) through data augmentation. As a result, we did not, for example, test the impact of individual modifications to the NFR method for language combinations other than EN-HU and HU-EN, but only evaluated the combined impact of all modifications in comparison to the baselines. Finally, we relied on automated evaluation metrics only, and did not conduct any experiments involving human evaluation.

7. Conclusions

The experiments conducted in this study confirm previous findings [24,25] that applying the NFR data augmentation method leads to a considerable increase in estimated translation quality compared to baseline NMT models in a translation context in which a high-quality TM is available. Moreover, we identified a number of adaptations that can further improve the quality of the MT output generated by the NFR systems: retrieving fuzzy matches using cosine similarity for sentence embeddings obtained on the basis of sub-word units, adding features based on alignment information, and increasing the informativeness of retrieved matches by maximising source sentence coverage. When all proposed methods are combined, statistically significant improvements in BLEU scores were reported for all eight tested language combinations, EN↔{HU,NL,FR,PL}, compared to a baseline Transformer NMT model (up to +8.4 BLEU points) as well as an already strong NFR baseline [25] (up to +1.69 BLEU points).

We argue that TM-NMT integration is both useful in contexts where the generated automatic translation is used as a final product, and for integration in a professional translator's workflow. Not only does NFR increase the quality of the generated MT output, it may also help to overcome the lack of confidence in MT output on the part of translators. Both of these factors potentially have a positive impact on the required post-editing time.

There are several lines of research arising from this work that we intend to pursue. We would like to explore further adjustments to the NFR method involving (a) additional sub-word segmentation methods for fuzzy match retrieval, such as WordPiece [66] or SentencePiece [79], as well as fuzzy match retrieval using lemmas [80]; (b) the use of pre-trained, multilingual sentence embeddings for fuzzy matching [81]; (c) more recent neural word alignment methods [47]; (d) alternative fuzzy match combination methods, e.g., by weighting fuzzy match score and the amount of overlap between input sentence and fuzzy matches; and (e) combinations with techniques for automatic post-editing [38,82]. A second line of research is to investigate the factors that influence the optimal fuzzy match threshold further, with the aim of better informing the selection of this threshold. It could also be interesting to supplement our experiments with an analysis of attention or saliency [83] to gain more insight into how the NMT system deals with augmented input sentences, for example to better study the impact of alignment features. Finally, despite the improvements achieved in estimated translation quality, the usefulness of the translations generated by the NFR system is yet to be confirmed by human judgements in the context of CAT workflows. In future work, we would also like to perform human evaluations, both in terms of perceived quality and post-editing time.

Author Contributions: Conceptualization, A.T., B.B. and B.V.; methodology, A.T., B.B. and B.V.; software, A.T., B.B. and B.V.; validation, A.T., B.B. and B.V.; formal analysis, A.T., B.B. and B.V.; investigation, A.T., B.B. and B.V.; resources, A.T., B.B. and B.V.; data curation, A.T., B.B. and B.V.; writing—original draft preparation, A.T. and B.B.; writing—review and editing, A.T., B.B. and B.V.; visualization, A.T., B.B. and B.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Research Foundation—Flanders (FWO).

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

BLEU	Bilingual Evaluation Understudy
BPE	Byte-Pair Encoding
CAT	Computer-Assisted Translation
ED	Fuzzy Matching using Edit Distance
LMVR	Linguistically Motivated Vocabulary Reduction
GPU	Graphics Processing Unit
NFR	Neural Fuzzy Repair
NMT	Neural Machine Translation
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MT	Machine Translation
PBSMT	Phrase-Based Statistical Machine Translation
SE	Fuzzy Matching using Sentence Embeddings
TER	Translation Edit Rate
TM	Translation Memory

Appendix A. Detailed Information on Data Set Sizes per Language Combination

Table A1. Number of sentences in train, validation and test sets, where M stands for millions, and average length of sentences in number of tokens, per language, calculated after tokenisation.

Language Combination	Train (M)	Validation	Test	Avg. Length
EN-HU	2.388	3000	3207	22–19.1
HU-EN	2.388	3000	3190	19.1–22
EN-NL	2.389	3000	3207	22–22.1
NL-EN	2.389	3000	3194	22.1–22
EN-FR	2.373	2998	3205	22.1–26
FR-EN	2.371	2998	3205	26–22.1
EN-PL	2.371	2998	3205	22.1–20.3
PL-EN	2.371	2998	3205	20.3–22.1

Appendix B. Hyper-Parameters and Training Details

Appendix B.1. NMT Architecture

We trained our models with OpenNMT [84] using the *Transformer* architecture [54], with the hyper-parameters listed in Table A2.

Table A2. Hyper-parameters for training NMT models.

Hyper-Parameter	**Value**
source/target embedding dimension	512
size of hidden layers	512
feed-forward layers	2048
number of heads	8
number of layers	6
batch size	64 sentences [1]
gradient accumulation	2
dropout	0.1
warm-up steps	8000
optimizer	Adam

[1] We do not set batch size using number of tokens as this approach leads to a considerable difference in the number of sentences in batches in the NFR and the baseline NMT settings.

For the systems that utilise source-side features, we use the source word embedding size of 506, with six cells for the features (total embedding size of 512). We use a total of 1 million steps in training with validation at every 5000 steps. All of the models are trained with early stopping: the training ends when the system has not improved for 10 validation rounds in terms of both accuracy and perplexity. We limit the source and target sentence length to 100 tokens for training the baseline NMT. The source sentence is limited to 300 tokens for training the NFR models as two additional sentences are used for data augmentation in all cases. All training runs are initialised using the same seed to avoid differences between systems due to the effect of randomness.

Appendix B.2. Sent2vec

To train our sent2vec models, we use the same hyper-parameters that are suggested in the description paper [55] for a sent2vec model trained on Wikipedia data containing both unigrams and bigrams. In our experiments, we distributed training of a sent2vec model over 40 threads. The hyper-parameters are provided in Table A3.

Table A3. Hyper-parameters for training sent2vec models.

Hyper-Parameter	**Value**
embedding dimension	700
minimum word count	8
minimum target word count	20
initial learning rate	0.2
epochs	9
subsampling hyper-parameter	5×10^{-6}
bigrams dropped per sentence	4
number of negatives sampled	10

Appendix B.3. FAISS

Because our goal is to find matches over all available sentences in the FAISS index, we create a Flat index with an inner product metric to do a brute-force search. By adding the L2-normalised vectors of the sentence representation to the index, and using an L2-normalised sentence vector as an input query, we are effectively using cosine similarity as match metric. More information can be found here: https://github.com/facebookresearch/faiss/wiki.

Appendix B.4. LMVR

To use LMVR, we first have to train a Morfessor model (https://morfessor.readthedocs.io/en/latest/cmdtools.html#morfessor-train). This baseline is then refined by LMVR. We use the same settings (see Table A4) as suggested in the examples here: https://github.com/d-ataman/lmvr/blob/master/examples/example-train-segment.sh.

Table A4. Hyper-parameters for training LMVR models.

Hyper-Parameter	**Value**
perplexity threshold	10
dampening	none
minimum shift remainder	1
length threshold	5
minimum perplexity length	1
maximum epochs	5
lexicon size	32,000

Appendix B.5. GIZA++

We use MGIZA [85], a multi-thread version of GIZA++, with the alignment heuristic `grow-diag-final-and` and Good–Turing smoothing.

Appendix C. Bin Sizes for Comparisons between NFR and Baseline Systems for Different Match Ranges and Sentence Lengths

Table A5 shows the number of sentences in the test set classified by input sentence length (i.e., number of tokens prior to sub-word segmentation) and fuzzy match score for EN-HU, for both SE_{tok}+*M* and SE_{bpe}+*M*, the two best-performing systems. The distribution of matches across match ranges is markedly different for the two systems, with the similarity score of token-based matches concentrated towards the higher end of the scale. There are hardly any best matches that score below 0.60. For SE_{bpe}+*M*, the matches are spread out more evenly across the different match ranges.

Table A5. Bin sizes per match range and sentence length (in number of words) for EN-HU.

	SE_{tok}+*M*			SE_{bpe}+*M*		
Sent. Length **Match Score**	**1–10**	**11–25**	**>25**	**1–10**	**11–25**	**>25**
0.50–0.59	5	9	3	81	181	164
0.60–0.69	41	197	124	104	237	310
0.70–0.79	101	253	420	122	194	318
0.80–0.89	147	217	350	136	184	277
0.90–0.99	346	381	606	125	219	412

A similar analysis for HU-EN is shown in Table A6. In addition, here the distribution of bin sizes differs for tokens and BPE, similarly to what was noted for EN-HU.

Table A6. Bin sizes per match range and sentence length (in number of words) for HU-EN.

	SE_{tok}+M			SE_{bpe}+M		
Sent. Length **Match Score**	**1–10**	**11–25**	**>25**	**1–10**	**11–25**	**>25**
0.50–0.59	3	5	3	63	230	204
0.60–0.69	26	197	217	114	269	294
0.70–0.79	90	375	400	152	202	219
0.80–0.89	152	237	250	163	202	194
0.90–0.99	417	428	376	207	297	320

References

1. Koehn, P. *Neural Machine Translation*; Cambridge University Press: Cambridge, UK, 2020; doi:10.1017/9781108608480.
2. Chung, J.; Cho, K.; Bengio, Y. A Character-level Decoder without Explicit Segmentation for Neural Machine Translation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1693–1703; doi:10.18653/v1/P16-1160.
3. Koponen, M. Is machine translation post-editing worth the effort? A survey of research into post-editing and effort. *J. Spec. Transl.* **2016**, *25*, 131–148.
4. Rossi, C.; Chevrot, J.P. Uses and perceptions of Machine Translation at the European Commission. *J. Spec. Transl.* **2019**, *31*, 177–200.
5. Stefaniak, K. Evaluating the usefulness of neural machine translation for the Polish translators in the European Commission. In Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, Lisboa, Portugal, 3–5 November 2020; European Association for Machine Translation: Lisboa, Portugal, 2020; pp. 263–269.
6. Macken, L.; Prou, D.; Tezcan, A. Quantifying the effect of machine translation in a high-quality human translation production process. *Informatics* **2020**, *7*, 19.
7. Läubli, S.; Amrhein, C.; Düggelin, P.; Gonzalez, B.; Zwahlen, A.; Volk, M. Post-editing Productivity with Neural Machine Translation: An Empirical Assessment of Speed and Quality in the Banking and Finance Domain. In Proceedings of the Machine Translation Summit XVII, Dublin, Ireland, 19–23 August 2019; European Association for Machine Translation: Dublin, Ireland, 2019; pp. 267–272.
8. Sanchez-Torron, M.; Koehn, P. Machine Translation Quality and Post-Editor Productivity. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA) Volume 1: MT Researchers' Track, Austin, TX, USA, 28 October–1 November 2016; Association for Machine Translation in the Americas (AMTA): Austin, TX, USA, 2016; pp. 16–26.
9. Christensen, T.P.; Schjoldager, A. Translation-memory (TM) research: What do we know and how do we know it? *HERMES J. Lang. Commun. Bus.* **2010**, 89–101, doi:10.7146/hjlcb.v23i44.97268.
10. Reinke, U. State of the art in translation memory technology. In *Language Technologies for a Multilingual Europe*; Rehm, G., Stein, D., Sasaki, F., Witt, A., Eds.; Language Science Press: Berlin, Germany, 2018; Chapter 5, pp. 55–84; doi:10.5281/zenodo.1291930.
11. Seal, T. ALPNET and TSS: The commercial realities of using a computeraided translation system. In *Translating and the Computer 13, Proceedings from the Aslib Conference*; Aslib: London, UK, 1992; pp. 120–125.
12. Federico, M.; Cattelan, A.; Trombetti, M. Measuring user productivity in machine translation enhanced Computer Assisted Translation. In Proceedings of the 2012 Conference of the Association for Machine Translation in the Americas, San Diego, CA, USA, 28 October–1 November 2012; AMTA: San Diego, CA, USA, 2012; pp. 44–56.
13. Simard, M.; Isabelle, P. Phrase-based machine translation in a computer-assisted translation environment. In Proceedings of MT Summit XII, Ottawa, ON, Canada, 26–30 August 2009; AMTA: Ottawa, ON, Canada, 2009; pp. 120–127.
14. Sánchez-Gijón, P.; Moorkens, J.; Way, A. Post-editing neural machine translation versus translation memory segments. *Mach. Transl.* **2019**, *33*, 31–59.
15. Baldwin, T. The hare and the tortoise: speed and accuracy in translation retrieval. *Mach. Transl.* **2009**, *23*, 195–240.
16. Bloodgood, M.; Strauss, B. Translation Memory Retrieval Methods. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 26–30 April 2014; Association for Computational Linguistics: Gothenburg, Sweden, 2014; pp. 202–210; doi:10.3115/v1/E14-1022.
17. Moorkens, J.; O'Brien, S. Assessing user interface needs of post-editors of machine translation. In *Human Issues in Translation Technology: The IATIS Yearbook*; Taylor & Francis: Abingdon, UK, 2016; pp. 109–130.
18. Langlais, P.; Simard, M. Merging example-based and statistical machine translation: an experiment. In Proceedings of the Conference of the Association for Machine Translation in the Americas, Tiburon, CA, USA, 6–12 October 2002; Springer: Tiburon, CA, USA, 2002; pp. 104–113.

19. Marcu, D. Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 9–11 July 2001; Association for Computational Linguistics: Toulouse, France, 2001; pp. 386–393; doi:10.3115/1073012.1073062.
20. Simard, M.; Langlais, P. Sub-sentential exploitation of translation memories. In Proceedings of the Machine Translation Summit VIII, Santiago de Compostela, Spain, 18–22 September 2001; EAMT: Santiago de Compostela, Spain, 2001; Volume 8, pp. 335–339.
21. Feng, Y.; Zhang, S.; Zhang, A.; Wang, D.; Abel, A. Memory-augmented Neural Machine Translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 1390–1399; doi:10.18653/v1/D17-1146.
22. Gu, J.; Wang, Y.; Cho, K.; Li, V.O.K. Search engine guided neural machine translation. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Association for the Advancement of Artificial Intelligence: New Orleans, LA, USA, 2018; pp. 5133–5140.
23. Zhang, J.; Utiyama, M.; Sumita, E.; Neubig, G.; Nakamura, S. Guiding Neural Machine Translation with Retrieved Translation Pieces. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 1325–1335; doi:10.18653/v1/N18-1120.
24. Bulte, B.; Tezcan, A. Neural Fuzzy Repair: Integrating Fuzzy Matches into Neural Machine Translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1800–1809; doi:10.18653/v1/P19-1175.
25. Xu, J.; Crego, J.; Senellart, J. Boosting Neural Machine Translation with Similar Translations. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1580–1590; doi:10.18653/v1/2020.acl-main.144.
26. Krollmann, F. Linguistic data banks and the technical translator. *Meta* **1971**, *16*, 117–124.
27. Chatzitheodorou, K. Improving translation memory fuzzy matching by paraphrasing. In Proceedings of the Workshop Natural Language Processing for Translation Memories, Hissar, Bulgaria, 11 September 2015; Association for Computational Linguistics: Hissar, Bulgaria, 2015; pp. 24–30.
28. Levenshtein, V. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
29. Vanallemeersch, T.; Vandeghinste, V. Assessing linguistically aware fuzzy matching in translation memories. In Proceedings of the 18th Annual Conference of the European Association for Machine Translation, Antalya, Turkey, 11–13 May 2015; EAMT: Antalya, Turkey, 2015; pp. 153–160.
30. Snover, M.; Dorr, B.; Schwartz, R.; Micciulla, L.; Makhoul, J. A study of translation edit rate with targeted human annotation. In Proceedings of the 2006 Conference of the Association for Machine Translation in the Americas, Cambridge, MA, USA, 8–12 August 2006; AMTA: Cambridge, MA, USA, 2006; pp. 223–231.
31. Vanallemeersch, T.; Vandeghinste, V. Improving fuzzy matching through syntactic knowledge. *Transl. Comput.* **2014**, *36*, 217–227.
32. Ranasinghe, T.; Orasan, C.; Mitkov, R. Intelligent Translation Memory Matching and Retrieval with Sentence Encoders. In Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, Lisboa, Portugal, 3–5 November 2020; European Association for Machine Translation: Lisboa, Portugal, 2020; pp. 175–184.
33. Steinberger, R.; Eisele, A.; Klocek, S.; Pilos, S.; Schlüter, P. DGT-TM: A freely available Translation Memory in 22 languages. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, 23–25 May 2012; European Language Resources Association (ELRA): Istanbul, Turkey, 2012; pp. 454–459.
34. Bulté, B.; Vanallemeersch, T.; Vandeghinste, V. M3TRA: integrating TM and MT for professional translators. In Proceedings of the 21st Annual Conference of the European Association for Machine Translation, Alicante, Spain, 28–30 May 2018; EAMT: Alicante, Spain, 2018; pp. 69–78.
35. Hewavitharana, S.; Vogel, S.; Waibel, A. Augmenting a statistical translation system with a translation memory. In Proceedings of the 10th Annual Conference of the European Association for Machine Translation, Budapest, Hungary, 30–31 May 2005; European Association for Machine Translation: Budapest, Hungary, 2005; pp. 126–132.
36. Kranias, L.; Samiotou, A. 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 (LREC'04), Lisbon, Portugal, 26–28 May 2004; European Language Resources Association (ELRA): Lisbon, Portugal, 2004; pp. 331–334.
37. Ortega, J.E.; Sánchez-Martınez, F.; Forcada, M.L. Fuzzy-match repair using black-box machine translation systems: What can be expected. In Proceedings of the AMTA, Austin, TX, USA, 30 October–3 November 2016; AMTA: Austin, TX, USA, 2016; Volume 1, pp. 27–39.
38. Ortega, J.; Sánchez-Martínez, F.; Turchi, M.; Negri, M. Improving Translations by Combining Fuzzy-Match Repair with Automatic Post-Editing. In Proceedings of the Machine Translation Summit XVII, Dublin, Ireland, 19–23 August 2019; European Association for Machine Translation: Dublin, Ireland, 2019; pp. 256–266.
39. Ortega, J.E.; Forcada, M.L.; Sanchez-Martinez, F. Fuzzy-match repair guided by quality estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, doi:10.1109/TPAMI.2020.3021361.
40. Carl, M.; Way, A. *Recent Advances in Example-Based MACHINE Translation*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2003; Volume 21.

41. Nagao, M. A Framework of a Mechanical Translation Between Japanese and English by Analogy Principle. In *Artificial and Human Intelligence*; Elithorn, A.; Barnerji, R., Eds.; North-Holland: Amsterdam, The Netherlands, 1984; pp. 173–180.
42. Dandapat, S.; Morrissey, S.; Way, A.; Forcada, M.L. Using example-based MT to support statistical MT when translating homogeneous data in a resource-poor setting. In Proceedings of the 15th Annual Meeting of the European Association for Machine Translation, Leuven, Belgium, 30–31 May 2011; European Association for Machine Translation: Leuven, Belgium, 2011; pp. 201–208.
43. Smith, J.; Clark, S. EBMT for SMT: A new EBMT-SMT hybrid. In Proceedings of the 3rd International Workshop on Example-Based Machine Translation, Dublin, Ireland, 12–13 November 2009; European Association for Machine Translation: Dublin, Ireland, 2009; pp. 3–10.
44. Castilho, S.; Moorkens, J.; Gaspari, F.; Calixto, I.; Tinsley, J.; Way, A. Is neural machine translation the new state of the art? *Prague Bull. Math. Linguist.* **2017**, *108*, 109–120.
45. Koehn, P.; Senellart, J. Convergence of Translation Memory and Statistical Machine Translation. In Proceedings of AMTA Workshop on MT Research and the Translation Industry, Denver, CO, USA, 31 October–4 November 2010; Association for Machine Translation in the Americas: Denver, CO, USA, 2010; pp. 21–31.
46. Biçici, E.; Dymetman, M. Dynamic translation memory: Using statistical machine translation to improve translation memory fuzzy matches. In Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, Haifa, Israel, 17–23 February 2008; Springer: Haifa, Israel, 2008; pp. 454–465.
47. Li, L.; Escartin, C.P.; Liu, Q. Combining Translation Memories and Syntax-Based SMT: Experiments with Real Industrial Data. In Proceedings of the 19th Annual Conference of the European Association for Machine Translation, Riga, Latvia, 30 May–1 June 2016; European Association for Machine Translation: Riga, Latvia, 2016; pp. 165–177.
48. Wang, K.; Zong, C.; Su, K.Y. Integrating Translation Memory into Phrase-Based Machine Translation during Decoding. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; Association for Computational Linguistics: Sofia, Bulgaria, 2013; pp. 11–21.
49. Cao, Q.; Xiong, D. Encoding Gated Translation Memory into Neural Machine Translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 3042–3047; doi:10.18653/v1/D18-1340.
50. Hokamp, C.; Liu, Q. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 1535–1546; doi:10.18653/v1/P17-1141.
51. Khandelwal, U.; Fan, A.; Jurafsky, D.; Zettlemoyer, L.; Lewis, M. Nearest neighbor machine translation. *arXiv* **2020**, arXiv:2010.00710.
52. Hokamp, C. Ensembling Factored Neural Machine Translation Models for Automatic Post-Editing and Quality Estimation. In Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 647–654; doi:10.18653/v1/W17-4775.
53. Dabre, R.; Cromieres, F.; Kurohashi, S. Enabling multi-source neural machine translation by concatenating source sentences in multiple languages. *arXiv* **2017**, arXiv:1702.06135.
54. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 30th Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Neural Information Processing Systems Foundation: Long Beach, CA, USA, 2017; pp. 5998–6008.
55. Pagliardini, M.; Gupta, P.; Jaggi, M. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, 1–6 June 2018; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 528–540; doi:10.18653/v1/N18-1049.
56. Dinu, G.; Mathur, P.; Federico, M.; Al-Onaizan, Y. Training Neural Machine Translation to Apply Terminology Constraints. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 3063–3068; doi:10.18653/v1/P19-1294.
57. Gu, J.; Wang, C.; Zhao, J. Levenshtein transformer. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Neural Information Processing Systems Foundation: Vanvouver, BC, Canada, 2019; pp. 11181–11191.
58. Susanto, R.H.; Chollampatt, S.; Tan, L. Lexically Constrained Neural Machine Translation with Levenshtein Transformer. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 3536–3543; doi:10.18653/v1/2020.acl-main.325.
59. Alkhouli, T.; Bretschner, G.; Peter, J.T.; Hethnawi, M.; Guta, A.; Ney, H. Alignment-Based Neural Machine Translation. In Proceedings of the First, Conference on Machine Translation, Berlin, Germany, 11–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 54–65; doi:10.18653/v1/W16-2206.
60. Li, Z.; Specia, L. Improving Neural Machine Translation Robustness via Data Augmentation: Beyond Back-Translation. In Proceedings of the 5th Workshop on Noisy User-generated Text, Hong Kong, China, 4 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 328–336; doi:10.18653/v1/D19-5543.

61. Hossain, N.; Ghazvininejad, M.; Zettlemoyer, L. Simple and Effective Retrieve-Edit-Rerank Text Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 2532–2538; doi:10.18653/v1/2020.acl-main.228.
62. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* **2019**, doi:10.1109/TBDATA.2019.2921572.
63. Artetxe, M.; Schwenk, H. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transact. Assoc. for Comput. Linguist.* **2019**, 7, 597-610.
64. Chaudhary, V.; Tang, Y.; Guzmán, F.; Schwenk, H.; Koehn, P. Low-Resource Corpus Filtering Using Multilingual Sentence Embeddings. In Proceedings of the Fourth Conference on Machine Translation, Florence, Italy, 1–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 261–266; doi:10.18653/v1/W19-5435.
65. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1715–1725; doi:10.18653/v1/P16-1162.
66. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv* **2016**, arXiv:1609.08144.
67. Ataman, D.; Negri, M.; Turchi, M.; Federico, M. Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *Prague Bull. Math. Linguist.* **2017**, *108*, 331–342.
68. Gage, P. A New Algorithm for Data Compression. *C Users J.* **1994**, *12*, 23–38.
69. Schuster, M.; Nakajima, K. Japanese and Korean voice search. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 5149–5152; doi:10.1109/ICASSP.2012.6289079.
70. Jurafsky, D.; Martin, J.H. *Speech and Language Processing*, 2nd ed.; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2009.
71. Och, F.J.; Ney, H. A Systematic Comparison of Various Statistical Alignment Models. *Comput. Linguist.* **2003**, *29*, 19–51.
72. Dyer, C.; Chahuneau, V.; Smith, N.A. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 9–14 June 2013; Association for Computational Linguistics: Atlanta, GA, USA, 2013; pp. 644–648.
73. Zenkel, T.; Wuebker, J.; DeNero, J. End-to-End Neural Word Alignment Outperforms GIZA++. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1605–1617.
74. Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, 23–30 June 2007; Association for Computational Linguistics: Prague, Czech Republic, 2007; pp. 177–180.
75. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; Association for Computational Linguistics: Philadelphia, PA, USA, 2002; pp. 311–318; doi:10.3115/1073083.1073135.
76. Banerjee, S.; Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; Association for Computational Linguistics: Ann Arbor, MI, USA, 2005; pp. 65–72.
77. Koehn, P. Statistical Significance Tests for Machine Translation Evaluation. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 388–395.
78. Zhang, W.; Feng, Y.; Meng, F.; You, D.; Liu, Q. Bridging the gap between training and inference for neural machine translation. In Proceedings of the Fourth Conference on Machine Translation, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 4334–4343.
79. Kudo, T.; Richardson, J. 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, Brussels, Belgium, 31 October–3 November 2018; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 66–71; doi:10.18653/v1/D18-2012.
80. Hodász, G.; Pohl, G. MetaMorpho TM: A linguistically enriched translation memory. In Proceedings of the International Workshop: Modern Approaches in Translation Technologies, Borovets, Bulgaria, 24 September 2005; Incoma Ltd: Shoumen, Bulgaria, 2005; pp. 26–30.
81. Reimers, N.; Gurevych, I. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 4512–4525.
82. Chatterjee, R.; Negri, M.; Turchi, M.; Blain, F.; Specia, L. Combining Quality Estimation and Automatic Post-editing to Enhance Machine Translation output. In Proceedings of the 13th Conference of the Association for Machine Translation in the Americas, Boston, MA, USA, 17–21 March 2018; Association for Machine Translation in the Americas: Boston, MA, USA, 2018; pp. 26–38.

83. Ding, S.; Xu, H.; Koehn, P. Saliency-driven Word Alignment Interpretation for Neural Machine Translation. In Proceedings of the Fourth Conference on Machine Translation, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1–12.
84. Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; Rush, A.M. OpenNMT: Open-source toolkit for neural machine translation. *arXiv* **2017**, arXiv:1701.02810.
85. Gao, Q.; Vogel, S. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*; Association for Computational Linguistics: Columbus, OH, USA, 2008; pp. 49–57.