Incorporating Target Fuzzy Matches into Neural Fuzzy Repair

Tommi Nieminen and **Jörg Tiedemann** and **Sami Virpioja**
University of Helsinki, Dept. of Digital Humanities
firstname.lastname@helsinki.fi

Abstract

Neural fuzzy repair (NFR) is a simple implementation of retrieval-augmented translation (RAT), based on data augmentation. In NFR, a translation database is searched for translation examples where the source sentence is similar to the sentence being translated, and the target side of the example is concatenated with the source sentences. We experiment with introducing retrieval that is based on target similarity to NFR during training. The results of our experiments confirm that including target similarity matches during training supplements source similarity matches and leads to better translations at translation time.

1 Introduction

Retrieval-augmented translation (RAT) is a family of machine translation (MT) approaches where an MT system has access to translation examples when generating a translation for a source sentence. The translation examples are usually retrieved from a translation database based on similarity with the current translation context, which can be either the source sentence alone or a combination of the source sentence and the translation that has been generated so far. The similarity between the translation context and the translation examples from the database can be measured using lexical methods, such as edit distance and longest matching N-gram, or based on the distance between the vector representations of the example and the translation context. The intuition behind RAT is that the MT system can, given an unseen source sentence, use the retrieved matches as additional information when constructing a translation. This supports the translation task, as the MT system no longer has to rely solely on the information embodied in the neural network, and different RAT methods have been shown conclusively to improve MT quality (see for example Bulte and Tezcan (2019), Khandelwal et al. (2021)).

This article focuses on a variant of RAT based on augmenting data with lexical matches, first discussed in Bulte and Tezcan (2019), called Neural Fuzzy Repair (NFR). Our work further develops NFR by incorporating translation examples that have been retrieved based on target instead of source similarity. We also test how annotating source sentences with the similarity levels of the translation examples affects quality.

2 Related work

Many RAT approaches draw inspiration from retrieval methods that have been used in professional translation from the 1960s onward. The three main traditional forms of retrieval in professional translation (Hutchins, 1998) are terminology lookup from a terminology database, full segment fuzzy match retrieval from a translation memory (usually based on edit distance), and concordance search from a translation memory (retrieving translation pairs based on the occurrence of a particular substring on the source side). In recent decades, various subsegmental retrieval methods have also been introduced (Flanagan, 2014).

In MT research prior to the adoption of neural machine translation (NMT), the concept of retrieving translation examples based on source similarity and the construction of new translations from the retrieved examples was first proposed in the 1980s in the form of example-based MT (Nagao, 1984). In statistical MT, retrieving parts of existing translations from translation tables in order to generate new translations was a core component of MT systems, and there were also attempts to integrate translation memory retrieval more directly in a manner resembling RAT (Koehn and Senellart, 2010).

Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025), pages 408–418
March 3-4, 2025

Within NMT, various RAT methods have been proposed. They can be roughly divided into three categories, depending on whether they are based on purpose-built neural network architectures, data augmentation, or changes in the decoder component of the MT system.

Gu et al. (2017) introduces the first NMT architecture designed for RAT: translation examples are retrieved from a translation database based on sentence similarity, and the attention component of the MT system is extended to cover the retrieved examples. Bapna and Firat (2019) uses a similar architecture-based approach, but uses N-gram- and vector-based retrieval to increase the amount of matches. Hoang et al. (2022) attempts to control the source-match interactions by encoding each retrieved match separately with the source sentence.

RAT based on data augmentation was introduced in Bulte and Tezcan (2019), where source sentences are concatenated with target translations from translation examples that are retrieved from the translation database with lexical matching. Xu et al. (2020) extends the lexical matching to separate relevant and irrelevant target tokens by using word alignment data, and also utilizes matches based on vector similarity. Concatenation-based data augmentation methods are also used to constrain MT output to contain terms from a terminology database (Dinu et al., 2019), which can be considered a form of RAT.

Decoder-based RAT has the advantage of being usable with any NMT model, since the model parameters and architecture are not changed. One early implementation utilized phrase tables from SMT systems (Dahlmann et al., 2017). Currently, the most prominent form of decoder-based RAT is kNN-MT (Khandelwal et al., 2021), which generates a datastore consisting of pairs of translation contexts and output tokens. When generating the next token of a translation, the decoder searches for similar translation contexts based on vector similarity, and utilizes the output tokens corresponding to the most similar translation contexts in generating the next token.

Neural RAT has also been implemented with large language models (LLM) (Moslem et al., 2023) using in-context learning (ICL), where the LLM is prompted with the retrieved examples. Bouthors et al. (2024) compare LLM-based RAT with NFR, and NFR seems to have a clear quality advantage, although more advanced LLMs may have better results.

3 NFR with lexical matches

In NFR, the source language sentences in the training data are concatenated with target language sentences. The concatenated target language sentences originate from translation examples, where the source sentence is similar to the source sentence in the training data by some similarity measure. The concatenated target language sentences are separated from each other and the source sentence with a special symbol, and maximum amount of examples per sentence is usually limited to 3 (see Table 1 for examples).

NFR has been implemented using both lexical and vector-based retrieval methods (Tezcan and Bulté, 2022). It is easier to conceptualize with lexical retrieval methods, since there is a clear mechanism for utilizing the retrieved matches: find parts of the retrieved translation that match the parts of the new source sentence, and copy them to the new translation. Note that this copy behaviour has to be selective in two ways:

1. **Match selection**: The MT system may be provided with irrelevant or contradictory examples (if the system is designed to support multiple translation examples), so the system must be able to discard examples or to select the most appropriate one amongst multiple valid examples.

2. **Sub-sentential selection** Given relevant translation examples, the MT system has to identify the parts of the examples that can be exploited for constructing new translations and then adapt them correctly.

With vector-based retrieval methods, the mechanism for utilizing the matches is more murky, as there is often no lexical similarity with the retrieved translations and any acceptable translation for the new source sentence. Xu et al. (2020) found that using vector-based matches improves translation quality (although not by as much as lexical matches), and they hypothesize that vector-based matches improve quality by providing context during translation.

One issue, which is not explored in the existing research literature, is how a RAT system actually learns to utilize the retrieved matches.

Fuzzies	Augmented source sentence
1	Tuensaajia on kaksi . *FUZZYBREAK* There are two situations .
2	Turvallisuutta koskevat lisä vaatimukset *FUZZYBREAK* Käyttövarmuutta koskevat vaatimukset *FUZZYBREAK* Security requirements
3	Toimivaltaisen viranomaisen tehtävät ja velvollisuudet *FUZZYBREAK* Välimiesten tehtävät ja velvoitteet *FUZZYBREAK* HRE:n tehtävät ja velvollisuudet *FUZZYBREAK* Duties and obligations of children

Table 1: Source sentences (the English sentence after the last FUZZYBREAK delimiter symbol) augmented with 1 to 3 target sentences from similar translation examples (Finnish sentences separated by the delimiter symbols). Highlighted text indicates matching source and target portions.

For instance, for the MT model to learn the sub-sentential selection behaviour associated with lexical matches, it would seem necessary for the training data to contain examples consisting of a source sentence, one or more translations from retrieved translation examples, and a target sentence containing parts of those retrieved translations. However, in the existing RAT literature, the matches are retrieved based on source similarity, with no concern for whether any part of the target sides of the matches are actually present in the translations of the training data. The only article in which the target side similarity of the retrieved examples is discussed is Xu et al. (2020), where in one experiment source-side matches are re-ranked according to target-side similarity. Otherwise, there seems to be an implicit assumption that source-side similarity implies target-side similarity. However, most naturally occurring sentences have billions of possible translations (Dreyer and Marcu, 2012). Even though most of those possible translations are slight variations of other translations, for most sentences there is a large amount of valid translations that are meaningfully different both lexically and syntactically, as is demonstrated by the literature on increasing output diversity in machine translation (see for instance Roberts et al. (2020)).

This diversity in naturally occurring translations makes it unlikely that most translation pairs retrieved from naturally occurring data are optimal training examples for the copy behaviour that a RAT system should exhibit. However, since RAT systems trained with such data have been conclusively shown to improve translation quality and to copy tokens from the target sides of the retrieved matches to the new translations more often than normal MT systems (Xu et al., 2020), there must be enough good examples of copy behaviour in the training data. However, it is likely, that a large part of the lexical matches that are retrieved with source similarity do not exemplify the subsentential copy mode, but rather contextualize the translation in the same way as vector-based matches.

The objective of this work is to verify whether having training data that contains more suitable training examples of the expected selective copy behaviour improves the performance of NFR models. To obtain such training data, we retrieve lexical matches based on target similarity during the training phase. One issue with using target similarity at training time is that a model that is trained only with target similarity data cannot learn the first type of selective copy behaviour explained above, match selection. There will be no examples in the training data of irrelevant or contradictory matches, since all matches will be similar to their respective target sentences. Since at inference time, only matches based on source similarity will be available, the model will almost certainly copy irrelevant tokens from irrelevant matches to the output. On the other hand, the data is more conducive to learning the second type of copy behaviour, sub-sentential selection, since all the training examples are relevant for that purpose.

In our experiments, we attenuate the problem of copying irrelevant tokens by adding source-similarity matches to the target-similarity training data, and by ensembling source- and target-similarity models. We also include similarity class annotations in most models (a numerical suffix from 5 to 9 attached to the example marker), indicating the degree of similarity that each translation example has with the source or target sentence,

with the aim of training the model to process examples from different classes differently (for instance to copy less from low-similarity examples).

4 Data

Models are trained using the English to Finnish data from the Tatoeba-Challenge data set (release v2023-09-26) (Tiedemann, 2020). This data set consists of most of the data included in the OPUS corpus collection[1] at the date of the release. The data in OPUS includes many crawled data sets. Due to quality issues in crawled data (Kreutzer et al., 2022), the data is filtered with Bicleaner AI v2.0 (Zaragoza-Bernabeu et al., 2022): 5 million best sentence pairs according to Bicleaner AI are included in the training set (referred to as *Train-5M* from here on). During the initial experiments, we noticed that even after Bicleaner AI cleaning, much of the crawled data was of very low quality (containing for instance machine translations and lists of SEO terms). The crawled data also contains many repetitive text templates, which occur hundreds of times with small changes, such as *You can fly from [X] to [Y] indirect via [Z]* or *[WORD] pronunciation in [LANGUAGE]*. We suspected that the presence of these repetitive similar sentences in the training data (often with substandard translations) would affect the RAT training adversely. Because of these concerns, we decided to create another training set, which consists of 5 million best scoring non-crawled sentence pairs in the Tatoeba-Challenge data set (referred to as *NC-Train* from here on).

RAT can be used for **domain adaptation** by using a domain-specific translation database for retrieval. In order to test the domain adaptation performance of our RAT models, we exclude a portion of the Tatoeba-Challenge data set as domain test data. As there are no domain annotations included in the data, we treat each individual corpus in the dataset as a separate pseudo-domain and extract at most 1,000 sentence pairs from each of them as domain test sets. The corpora in the dataset mostly map to actual domains, e.g. the EMEA corpus contains data that is mostly from the pharmaceutical/medical domain. The crawled corpora are an exception, as they contain data from many domains, and they are therefore excluded from the domain test data. The domain test data is excluded from the training sets.

Each 5 million sentence pair training set is used as a database from which the translation examples are retrieved during the training phase for its respective training set. The training set database is also used as a translation database during testing. We also use a larger *All-Filtered* database consisting of all of the Tatoeba-Challenge data with a BiCleaner-AI score of at least 0.7 for testing. The *All-Filtered* database is used to determine whether the RAT system is capable of utilizing matches that it has not seen during training. For the domain-specific test sets, we also use domain-specific translation databases, which consist of all the domain-specific data in the *All-Filtered* database. For the *NC-Train*, the crawled data is excluded from the *All-Filtered* database.

4.1 Retrieving translation examples

Retrieving similar sentences from a large database for the millions of sentences in the training set is computationally costly, so expensive similarity metrics such as edit distance cannot be directly used. The training database needs to be filtered with a fast method that approximates more sophisticated methods, so that the more accurate similarity metrics can be applied to a smaller set of translation examples. Multiple retrieval methods have been proposed for RAT, but according to Bouthors et al. (2024), the choice of retrieval strategy does not have a noticeable effect on NFR performance. Because of this, we use the open-source *fuzzy-match* library[2] and do not explore other retrieval strategies. *fuzzy-match* uses suffix arrays for the initial filtering, and then calculates the edit distance over the resulting filtered set of translation examples. The search is performed on sentences tokenized to words. Note that this means that the morphological complexity of the language will affect the number of matches that are found: fewer matches will be found for morphologically complex languages in otherwise identical scenarios, as tokens tend to contain more morphemes and are therefore more varied.

To retrieve similar sentences for the sentence pairs in the training set, we first search the training database (*Train* and *NC-Train* for source similarity, *Train-TS* and *NC-Train-TS* for target similarity) for a maximum of 100 matches with a *fuzzy-match* edit distance score of at least 0.5 (with 1 being identical and 0 completely different). Per-

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

[2] https://github.com/SYSTRAN/fuzzy-match

Data set	DB	0.9-0.99	0.8-0.89	0.7-0.79	0.6-0.69	0.5-0.59	Total
Train	Train	1,085,811	1,659,270	1,461,893	2,248,088	3,214,326	9,669,388
Train	Train-TS	680,150	1,957,426	1,290,774	1,914,313	2,308,767	8,151,430
NC-train	NC-train	855,918	2,098,593	1,650,462	2,465,417	3,118,555	10,188,945
NC-train	NC-train-TS	680,150	1,957,426	1,290,774	1,914,313	2,308,767	8,151,430

Table 2: Amounts of translation examples retrieved for each data set and translation database. The examples are divided into five classes of with different similarity ranges, which are indicated on the header row.

Data set	DB	0.9-0.99	0.8-0.89	0.7-0.79	0.6-0.69	0.5-0.59	Total
Train	Train	250,475	375,277	335,332	523,819	933,652	2,418,555
Train	Train-TS	213,722	358,967	278,138	424,350	712,278	1,987,455
NC-train	NC-train	202,711	441,879	383,819	577,702	871,335	2,477,446
NC-train	NC-train-TS	167,540	439,522	335,202	494,354	670,596	2,107,214

Table 3: Amounts and classes of translation examples that were actually used to augment the data sets, with 1 matches max per sentence (the counts are somewhat larger with training sets that allow multiple matches).

fect matches are excluded from the results. Subsets of the matches are then selected randomly to augment the training data with translation examples. A maximum of three matches out of the possible hundred are actually used in our experiments, but retrieving the extra matches makes it possible to vary the examples based on their mutual similarity and to control the distribution of examples of different similarity scores in the training data. We use the contrastive retrieval functionality of *fuzzy-match* with a value of 0.7 to increase diversity in the retrieved examples. See Table 3 for details on the retrieved examples.

5 Models

We trained several models in the English to Finnish translation direction with both the *Train* and *NC-train* datasets. All the models are standard *transformer-base* models and were trained with the Marian NMT toolkit (Junczys-Dowmunt et al., 2018) v1.11.13 using default settings. We use SentencePiece (Kudo and Richardson, 2018) to create a vocabulary of 50,000 symbols, which includes marker symbols for indicating different similarity classes. A shared vocabulary is used for both source and target to facilitate the copying of tokens from the examples to the translation. All the models were trained to convergence.

The validation sets were selected from the development set included in the Tatoeba-Challenge data set by picking the longest sentences for which retrieved examples were available (the development set skews towards short sentences, which are problematic from the point of view of example retrieval). The validation sets were augmented using the same schemes that were used with the training data. This differs from test time, where only source similarity augmentation is used, but initial experiments indicated that using a different augmentation scheme for validation than the one used in the training data leads to unstable validation scores.

5.1 Augmentation schemes

The following augmentation schemes were used:

Baseline: A standard transformer model trained with non-augmented data.

Src-Sim: This is the standard augmentation scheme from Bulte and Tezcan (2019). Examples are retrieved based on source similarity. This can be considered the NFR baseline.

Trg-Sim: Examples are retrieved based on target similarity.

Combo: Sentence pairs from *Src-Sim* and *Trg-Sim* sets are combined. We test both combining all the sentence pairs from both sets (doubling the training set size to 10 M, referred to as *2X-Combo*), and picking odd sentence pairs from one set and even sentence pairs from the other (original training set size, referred to as *Combo*).

Mix-Sim: This scheme is only used when multiple translation examples are allowed. *Mix-Sim* differs from *Combo* in that translation examples

from both *Src-Sim* and *Trg-Sim* sets can be used simultaneously to augment the same source sentence. There can be a maximum of one *Trg-Sim* example per a sentence, the rest of the examples are picked from the *Source-Sim* set.

Manual inspection in early testing confirmed that models trained with the *Trg-Sim* scheme were prone to copying irrelevant tokens from the translation examples, especially with short sentences. The motivation for the *Combo-Sim* and *Mixed-Sim* schemes is to attenuate this problem of over-copying by mixing in source similarity examples into the training set. Another approach that we used to attenuating this problem was to ensemble *Src-Sim* and *Trg-Sim* models, as Hoang et al. (2024) indicates that ensembling models with diverse strengths leads to larger quality improvements than ensembling similar models. As a comparison, we also ensemble different checkpoints of some models.

We train models that allow a minimum of 1 and a maximum of 1-3 examples. In the augmentation phase, the examples are picked randomly from the full list of retrieved examples and concatenated with the source sentence. For all augmentation schemes, we generate training files both with and without fuzzy classes. The fuzzy class of an example is indicated in the data by using class-specific delimiter markers. Table 3 shows the ranges of *fuzzy_match* scores for each of the five classes used.

6 Evaluation

The test sets which are commonly used for MT evaluation are a bad fit for RAT evaluation, as they generally have very few fuzzy matches available even in large translation databases. For instance, for the *flores-devtest*, matches were found for only 72 out of 1,012 sentences in the *All-Filtered* set. More matches are found for the WMT news test sets, but the news domain is otherwise not well suited for RAT, as it is more varied and less repetitive than other domains.

Because of these concerns, we compiled our own test set. We extracted a maximum of 1,000 sentence pairs from each of the corpora that compose the Tatoeba-Challenge data set. We compiled separate test sets for the *Train* (75,249 sentence pairs) and *NC-Train* (65,549 sentence pairs) models. These test sets are mainly designed for domain translation performance evaluation, so we designate them as the *Domeval* and *Domeval-NC*. As the data has not been annotated with domain information, we use the sub-corpora as pseudo-domains.

For each sub-corpus, we build a *fuzzy_match* index using all the sentence pairs from that sub-corpus included in the respective *All-Filtered* set. We generate augmented versions of the source sentences of the *Domeval* sets using the subcorpus indexes, as well as the *Train* and *All-Filtered* indexes, and then translate the augmented *Domeval* source sentences using a model.

Domeval, 72,549 sents, with crawled data				
	Train DB		**All-Filtered**	
Scheme	**BLEU**	**chrF**	**BLEU**	**chrF**
Baseline	31.14	62.43	31.14	62.43
Src-Sim 1	32.32	62.66	41.08	66.95
-classes	32.67	63.06	41.14	67.17
Src-Sim 2	32.16	62.61	40.99	66.92
Src-Sim 3	31.92	62.45	40.32	66.35
-classes	32.01	62.56	39.69	65.88
Trg-Sim-1	31.87	62.52	40.23	66.62
-classes	32.08	62.54	40.58	66.67
Trg-Sim-2	31.49	62.22	39.60	66.07
Combo	32.38	62.76	41.21	67.14
2X-Combo	**32.82**	**63.16**	**41.57**	**67.47**
Mix-Sim-2	32.11	62.79	41.10	67.18
Mix-Sim-3	31.94	62.60	40.55	66.67
-classes	31.97	62.61	40.26	66.38

Domeval-NC, 65,549 sents, no crawled data				
	NC-Train DB		**NC-All-Filtered**	
Scheme	**BLEU**	**chrF**	**BLEU**	**chrF**
Baseline	30.86	62.35	30.86	62.35
Src-Sim 1	**32.26**	**62.99**	**37.61**	**65.62**
Trg-Sim-1	31.70	62.58	36.84	65.09
Combo	32.20	62.90	37.61	65.57

Table 4: Scores for all augmentation schemes. The scores are calculated over the whole *Domeval*, including sentences for which there are no examples. The results in the two tables are not directly comparable, but the relative performance of the models is similar. *-classes* indicates that a model has been trained without similarity class annotations.

We also evaluate the performance of the models on full *Domeval* set with the *Train* and *All-Filtered* databases to measure general translation performance. SacreBLEU (Post, 2018) is used to generate BLEU and chrF metric scores. Neural eval-

Domeval, 72,549 sents, with crawled data				
	Train		**All-Filtered**	
Ensemble	**BLEU**	**chrF**	**BLEU**	**chrF**
Baseline	31.14	62.43	31.14	62.43
Src-Sim-1 + Trg-Sim-1	**32.99** (32.32)	**63.27** (62.66)	**41.69** (41.08)	67.48 (66.95)
2X-Combo + 2X-Combo	32.93 (32.82)	63.21 (63.16)	41.66 (41.57)	**67.52** (67.47)
Src-Sim-1 + Src-Sim-1	32.93 (32.32)	63.18 (62.66)	41.54 (41.08)	67.36 (66.95)

Domeval-NC, 65,549 sents, no crawled data				
	NC-Train DB		**NC-All-Filtered**	
Ensemble	**BLEU**	**chrF**	**BLEU**	**chrF**
Baseline	30.86	62.35	30.86	62.35
Src-Sim-1 + Trg-Sim-1	**33.11** (32.26)	**63.62** (62.99)	**38.41** (37.61)	**66.20** (65.62)

Table 5: Ensemble scores. *Src-Sim-1+Src-Sim-1* and *2XCombo+2X-Combo1* are ensembles of different checkpoints of the same model. Values in parentheses indicate the metric scores for the model in the ensemble that had better scores individually. Note that the differences between the different ensembles in the upper table are not statistically significant.

uation metrics, such as COMET, have been found to be superior to lexical metrics, such as BLEU and chrF, in recent meta-evaluations (Freitag et al., 2022). However, in the context of evaluating RAT systems, it is desirable for metrics to reward copying parts of the translation examples to the translation. With lexical metrics, this happens to some degree (depending on the lexical similarity of the translation examples and reference translations). With neural metrics, the translations do not need to be lexically similar with the reference translations, which is usually their advantage, but it becomes a potential problem in the context of RAT evaluation. Lexical metrics have also been found to be adequate in contexts where they are used to evaluate similar MT systems (Kocmi et al., 2024), and all the models we compare share their training data, subword segmentation, and model architecture. Because of these factors, we decided to use only lexical evaluation metrics.

During test time, only examples retrieved based on source similarity are used, also with the models that were trained with target similarity, since target-side data would not be available in actual translation scenarios.

7 Discussion of the results

All results are in accordance with earlier evaluations of NFR in Bulte and Tezcan (2019) and Xu et al. (2020): NFR improves translation quality very significantly (up to 10 BLEU points) compared to a NMT baseline.

The domain translation results for the five domains with most retrieved translation examples (see Table 6) are more ambivalent, although it should be noted that two of the five domains are highly atypical. The *Open-Subtitles* corpus consists of subtitles of TV shows and films, which are typically very short in order to fit the screen and often non-literal, due to e.g. jokes and references to visual content. Consequently the metric scores are very low for the domain. The *bible-uedin* corpus receives very high scores, which is probably due to repetition in the corpus, which means that very similar translation examples are available for many sentences. The scores are higher for *Train*, indicating that the crawled data contains bible translations.

Evaluation of both full test sets and specific domains suggests that annotating similarity classes of examples in the source sentences degrades translation quality slightly compared to treating all examples in the same way. It should be noted, though, that for the *EMEA*, *DGT*, and *Mozilla-I0n* domains similarity class annotation does seem to improve translation quality. These are also domains that are well-suited for RAT, as they are repetitive and noncreative.

The *Trg-Sim* scheme underperforms all other schemes on its own, probably due to excessive copying from the retrieved matches. However, models combining source and target similarity matches perform better than pure *Src-Sim* models. In domain-specific evaluation, the best results are

Train: domain translation, domains with most matches, only matches from domain DB										
	Open-Subtitles (822)		EMEA (654)		DGT (589)		bible-uedin (523)		Mozilla-I10n (482)	
	BLEU	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**
Src-Sim-1	28.15	53.60	58.89	77.31	69.78	82.60	93.38	96.01	69.23	79.04
-classes	28.35	53.72	58.17	77.99	69.87	82.62	93.35	96.07	68.19	78.37
Src-Sim-2	27.41	53.50	58.30	77.06	66.25	80.47	**93.96**	96.54	65.24	76.17
Src-Sim-3	28.77	54.03	57.42	76.62	61.58	76.38	93.84	96.42	62.48	74.68
-classes	**28.99**	54.20	55.82	75.23	59.66	75.14	93.52	96.13	59.65	70.78
Trg-Sim-1	17.43	43.85	57.25	78.11	70.97	83.63	79.24	88.52	69.96	80.11
-classes	20.35	44.55	56.62	78.01	70.21	82.76	91.52	95.24	69.39	79.52
Trg-Sim-2	14.63	39.42	56.57	75.68	65.83	80.15	79.73	88.80	64.51	76.05
Combo	24.71	51.77	59.15	78.98	71.04	83.28	93.11	96.06	70.50	80.30
2X-Combo	26.79	52.73	58.73	79.17	70.85	83.36	93.52	96.50	70.77	79.99
Mix-Sim-2	22.56	49.30	59.00	78.22	67.65	81.37	92.79	95.92	67.91	78.33
Mix-Sim-3	24.49	50.82	56.65	74.95	64.69	79.03	93.10	96.18	63.64	75.11
-classes	25.69	51.06	56.57	76.23	62.54	77.53	93.12	95.92	62.84	74.63
Src-Sim-1 + Trg-Sim-1	22.00	48.13	**59.24**	77.92	**71.75**	**83.99**	90.92	94.90	**71.21**	**80.43**
Src-Sim-1 + Src-Sim-1	28.49	**54.21**	58.74	77.39	70.57	83.06	93.56	96.09	69.36	78.67
2X-Combo+ 2X-Combo	26.84	52.70	58.80	**79.30**	71.18	82.90	93.55	**96.58**	70.74	80.24

NC-Train: domain translation with domain database, domains with most matches										
	Open-Subtitles (822)		EMEA (654)		DGT (589)		bible-uedin (523)		Mozilla-I10n (482)	
	BLEU	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**
Src-Sim-1	**29.94**	**54.48**	60.00	**79.91**	72.47	84.86	**89.36**	**94.20**	69.81	79.52
Trg-Sim-1	18.25	44.51	57.91	78.67	72.05	84.28	75.28	86.59	68.95	78.61
Combo	27.66	53.24	**60.08**	79.76	**73.23**	**84.99**	87.21	93.03	69.64	79.52
Src-Sim-1 + Trg-Sim-1	22.84	49.23	59.61	79.61	73.02	**84.99**	85.36	92.06	**71.38**	**80.41**

Table 6: Domain translation BLEU and chrF metrics scores for all models and ensembles. The number in the parentheses under the domain name indicates how many sentences out of 1,000 had at least one translation example.

Train: domains with short sentences, only matches from domain DB								
	Ubuntu (313)		KDE (418)		GNOME (420)		WikiTitles (373)	
	BLEU	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**	**BLEU**	**chrF**
Mix-Sim-3	**66.30**	**78.34**	**68.12**	**80.23**	**67.21**	**78.95**	**56.71**	**74.16**
Src-Sim-1	62.58	74.68	62.73	76.27	63.64	76.35	47.09	69.17
Trg-Sim-1	60.61	74.40	62.92	77.07	62.89	76.54	45.58	69.84

Table 7: Scores for domains with short sentences (max 5 words per line). Not all models are shown here, but Mix-Sim models perform best, notably against Src-Sim-1, which we use as NFR baseline.

obtained with the *Combo* models and the ensemble of *Src-Sim* and *Trg-Sim* models.

While the *Mix-Sim* scheme does not appear to work generally, it performs better than alternatives with a specific subgroup of domains, i.e. those with very short sentences (see Table 7). In general, models that allow multiple examples are better with short sentences. One reason for this is probably that more examples are available for shorter sentences. However, it might also be due to the long source sentences becoming too long when augmented with multiple translation examples, thus degrading performance.

8 Conclusion and future work

Our experiments demonstrate that both adding target similarity matches to the training data, and ensembling *Trg-Sim* models with *Src-Sim* models improve the quality of translation output compared to normal NFR. In the future, we plan to extend the *2X-ComboSim* approach by replicating source sentences with different source and target similarity matches in the training data at a larger scale.

We also plan to experiment further on ensembling NFR models, including ensembles of models trained with different numbers of translation examples. Ensembling may also offer an alternative way of handling multiple translation examples: a 1-example model can be provided with multiple translation examples as separate inputs, the outputs of which can then be ensembled to produce a translation that is influenced by all the examples. Ensembling could also be used to combine terminology models (Dinu et al., 2019) and NFR models, by preparing separate inputs annotated with terminology and translation examples respectively, and ensembling the outputs.

References

Ankur Bapna and Orhan Firat. 2019. https://doi.org/10.18653/v1/N19-1191 Non-parametric adaptation for neural machine translation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Maxime Bouthors, Josep Crego, and François Yvon. 2024. https://doi.org/10.18653/v1/2024.findings-naacl.190 Retrieving examples from memory for retrieval augmented neural machine translation: A systematic comparison. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 3022–3039, Mexico City, Mexico. Association for Computational Linguistics.

Bram Bulte and Arda Tezcan. 2019. https://doi.org/10.18653/v1/P19-1175 Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.

Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. 2017. https://doi.org/10.18653/v1/D17-1148 Neural machine translation leveraging phrase-based models in a hybrid search. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1411–1420, Copenhagen, Denmark. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. https://doi.org/10.18653/v1/P19-1294 Training neural machine translation to apply terminology constraints. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Markus Dreyer and Daniel Marcu. 2012. https://aclanthology.org/N12-1017 HyTER: Meaning-equivalent semantics for translation evaluation. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 162–171, Montréal, Canada. Association for Computational Linguistics.

Kevin Flanagan. 2014. https://aclanthology.org/2014.tc-1.1 Filling in the gaps: what we need from TM subsegment recall. In Proceedings of Translating and the Computer 36, London, UK. AsLing.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. https://aclanthology.org/2022.wmt-1.2/ Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In Proceedings of the Seventh Conference on Machine Translation (WMT), pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2017. https://api.semanticscholar.org/CorpusID:3750771 Search engine guided non-parametric neural machine translation. ArXiv, abs/1705.07267.

Cuong Hoang, Devendra Singh Sachan, Prashant Mathur, Brian Thompson, and Marcello Federico. 2022. https://api.semanticscholar.org/CorpusID:252815975 Improving retrieval augmented neural machine translation by controlling source and fuzzy-match interactions. ArXiv, abs/2210.05047.

Hieu Hoang, Huda Khayrallah, and Marcin Junczys-Dowmunt. 2024. https://doi.org/10.18653/v1/2024.findings-naacl.35 On-the-fly fusion of large language models and machine translation. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 520–532, Mexico City, Mexico. Association for Computational Linguistics.

John Hutchins. 1998. https://api.semanticscholar.org/CorpusID:10644577 The origins of the translator's workstation. Machine Translation, 13:287–307.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. http://www.aclweb.org/anthology/P18-4020 Marian: Fast neural machine translation in C++. In Proceedings of ACL 2018, System Demonstrations, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. https://openreview.net/forum?id=7wCBOfJ8hJM Nearest neighbor machine translation. In 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net.

Tom Kocmi, Vilém Zouhar, Christian Federmann, and Matt Post. 2024. https://doi.org/10.18653/v1/2024.acl-long.110 Navigating the metrics maze: Reconciling score magnitudes and accuracies. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1999–2014, Bangkok, Thailand. Association for Computational Linguistics.

Philipp Koehn and Jean Senellart. 2010. https://aclanthology.org/2010.jec-1.4 Convergence of translation memory and statistical machine translation. In Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry, pages 21–32, Denver, Colorado, USA. Association for Machine Translation in the Americas.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. https://doi.org/10.1162/tacl_a_00447 Quality at a glance: An audit of web-crawled multilingual datasets. Transactions of the Association for Computational Linguistics, 10:50–72.

Taku Kudo and John Richardson. 2018. https://doi.org/10.18653/v1/D18-2012 SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. https://aclanthology.org/2023.eamt-1.22 Adaptive machine translation with large language models. In Proceedings of the 24th Annual Conference of the European Association for Machine Translation, pages 227–237, Tampere, Finland. European Association for Machine Translation.

Makoto Nagao. 1984. https://api.semanticscholar.org/CorpusID:18366233 A framework of a mechanical translation between japanese and english by analogy principle.

Matt Post. 2018. https://www.aclweb.org/anthology/W18-6319 A call for clarity in reporting BLEU scores. In Proceedings of the Third Conference on Machine Translation: Research Papers, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Nicholas Roberts, Davis Liang, Graham Neubig, and Zachary Lipton. 2020. https://www.amazon.science/publications/decoding-and-diversity-in-machine-translation Decoding and diversity in machine translation. In NeurIPS 2020 Workshop on Resistance AI.

Arda Tezcan and Bram Bulté. 2022. https://api.semanticscholar.org/CorpusID:245815894 Evaluating the impact of integrating similar translations into neural machine translation. Inf., 13:19.

Jörg Tiedemann. 2020. https://www.aclweb.org/anthology/2020.wmt-1.139 The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In Proceedings of the Fifth Conference on Machine Translation, pages 1174–1182, Online. Association for Computational Linguistics.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. https://doi.org/10.18653/v1/2020.acl-main.144 Boosting neural machine translation with similar translations. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1580–1590, Online. Association for Computational Linguistics.

Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz Rojas. 2022. ”https://aclanthology.org/2022.lrec-1.87” ”bicleaner AI: Bicleaner goes neural”. In ”Proceedings of the Thirteenth Language Resources and Evaluation Conference”, pages ”824–831”, ”Marseille, France”. ”European Language Resources Association”.