Covering a sentence in form and meaning with fewer retrieved sentences

Yuan Liu
Graduate School of IPS
Waseda University
Kitakyushu, Japan
lyalltowell@ruri.waseda.jp

Yves Lepage
Graduate School of IPS
Waseda University
Kitakyushu, Japan
yves.lepage@waseda.jp

Abstract

Retrieving similar sentences from a given collection of sentences is essential in a range of applications. In this work, we propose a novel method to retrieve several sentences that cover an input sentence in form and meaning with minimal redundancy, so as to enhance the overall coverage quality of the output sentences. We focus on the hierarchical granularity levels of sentence pieces, matching from common or similar n-grams to finer-grained words o subwords, using techniques from similar sentence retrieval and monolingual phrase alignment. Our method shows promising source and target coverage evaluation results when applied to parallel corpora. This shows the potential of our approach if integrated into an example-based machine translation system.

1 Introduction

Recent research has indicated that informative sentences retrieved from translation memories (TM) boost the performance of neural machine translation (NMT) systems (Xu et al., 2020; Bulté and Tezcan, 2019). In particular, to better integrate TM in NMT, Tezcan et al. (2021) implemented a fuzzy match combination method to maximise the coverage of words in source sentence for data augmentation.

In their work, Tezcan et al. (2021) argued that the idea of TM-NMT integration closely relates to the principle of example-based machine translation (EBMT) (Nagao, 1984). In EBMT, sentences which are similar to a given input sentence are retrieved from a bilingual corpus. The machine builds a translation of the input sentence from pieces of the translations of these similar sentences. For the machine to translate properly, the retrieved sentences should cover the input sentence.

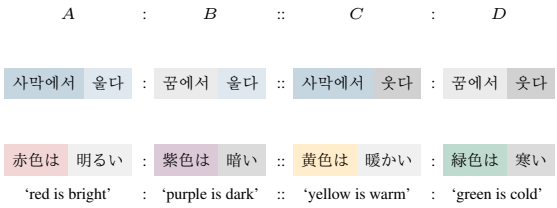
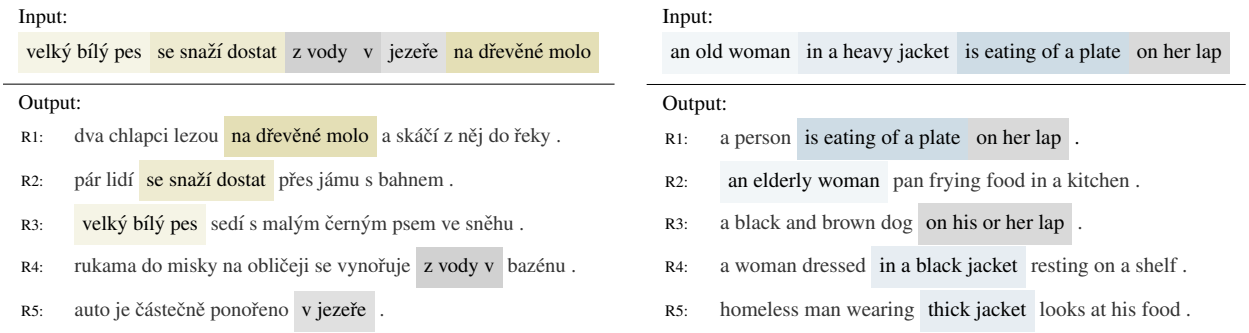


Figure 1: General pattern for analogy on the 1st line, formal analogy on the 2nd line, and semantic analogy on the 3rd line (with its translation into English). Analogies require formal or semantic coverage of a given sentence.

One approach to EBMT is translation by analogy (Lepage and Denoual, 2005). It relies on the preservation of proportional analogies between sentences across languages. A necessary condition for formal analogies and semantic analogies to hold is that sentence A has to be covered by sentences B and C, either in form or in meaning (Stroppa and Yvon, 2005; Langlais et al., 2009; Lepage, 2019), as illustrated in Figure 1. Further examples of sentences formally and semantically covering a sentence are shown in Figure 2.

As mentioned above, example sentences similar to the input sentence are retrieved for generating, tuning or boosting the translation of the input sentence, in a way that the similarity features of the retrieved sentences are decisive for the translation or



(a) Formally cover a given sentence (cs).

(b) Semantically cover a given sentence (en).

Figure 2: Examples of sentences formally and semantically covering a sentence (from Multi30k corpus).

for the explanation of the translation. It can be argued that the extent of the coverage of these similarity features contributes to the overall quality of the retrieved sentences and the performance of machine translation systems. In this paper, we discuss such coverage in form and meaning. However, we leave the evaluation in actual machine translation for future reports.

This work has a two-fold goal. Firstly, to retrieve similar sentences that formally and semantically cover a sentence as much as possible, in order to ultimately be able to translate a sentence by using the translation of the corresponding parts in the retrieved sentences that cover the input sentence. Secondly, to reduce the number of retrieved sentences as much as possible, in order to cover longer compact pieces of the input sentence so as to ultimately ensure a more reliable translation. Indeed, to cover a sentence in form or in meaning, one can simply match each token within a sentence, or identify the most similar tokens in the source corpus. By doing so, a large number of sentences would be retrieved, in proportion to the length of the input sentence. To avoid that, in addition to maximising the coverage, we aim at reducing the number of redundant sentences and intend to retrieve the least possible number of sentences.

2 Similarity scores

This section introduces several common scores used in similar sentence retrieval, including formal methods described in Sections 2.1 and 2.2, and the distributed method in Section 2.3.

2.1 Fuzzy matching score

The fuzzy matching score between two sentences is based on the edit distance between sentences in terms of tokens. The token-based edit distance is the number of operations, i.e., insertion, deletion and substitution between two sequences of tokens within sentences. The fuzzy match score is defined as:

$$\text{FM}(s_i, s_j) = 1 - \frac{\text{EditDistance}(s_i, s_j)}{\max(|s_i|, |s_j|)} \quad (1)$$

where $\text{EditDistance}(s_i, s_j)$ computes the edit distance between sentences, and $|s|$ denotes the length of sentence s. We compute the fuzzy matching score using an implementation[1] of the computation of the Levenshtein distance by Hyyrö (2001).

2.2 N-gram matching score

The N-gram matching score measures the length of the longest sequence of words that can be found in the source corpus, i.e., the longest common n-gram. Sentences containing this longest common n-gram are returned as output. More formally, the n-gram matching score is defined as:

$$\text{NM}(s_i, s_j) = \max\{|z| \ / \ z \in \mathcal{S}(s_i) \cap \mathcal{S}(s_j)\} \quad (2)$$

where $\mathcal{S}(s)$ denotes the set of all n-grams in s and $|s|$ the length of a string.

2.3 Contextual similarity search

Sentences can be retrieved by measuring the cosine similarity of sentence embeddings. The contextual

[1] https://github.com/roy-ht/editdistance

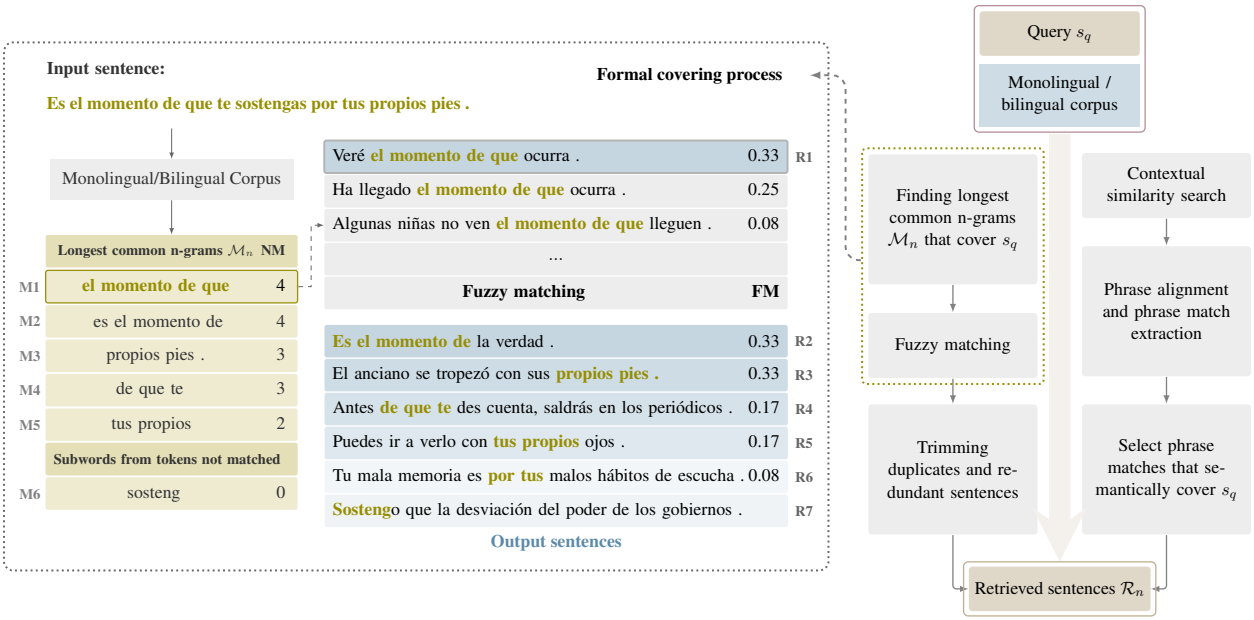


Figure 3: Overview of system architecture and the formal covering process. The sentences shown in this figure are extracted from the News Commentary Corpus (es-en).

similarity score is defined as:

$$\text{EM}(s_i, s_j) = \cos(\vec{s_i}, \vec{s_j}) = \frac{\vec{s_i} \cdot \vec{s_j}}{\|\vec{s_i}\| \times \|\vec{s_j}\|} \quad (3)$$

where $\|\vec{s}\|$ denotes the norm of vector s. In our work, we average the word embeddings derived from FastText (Bojanowski et al., 2017) embedding models and use pre-trained Sentence-BERT (Reimers and Gurevych, 2019) models to represent sentences.

3 Methodology

The architecture of our system is illustrated on the right of Figure 3. We consider both formal and semantic aspects of coverage and focus on large sentence pieces, i.e., formally common n-grams and semantically similar chunks, so as to cater for our aim of reducing redundancy.

3.1 Coverage in form

By coverage in form we mean that the words themselves or the sequences of words in the given sentence are found in the set of sentences that are retrieved. The overview of the formal coverage process is shown in Figure 3. The process is performed in the 3 main components described afterwards.

Algorithm 1: Matching longest common n-grams that cover a given sentence

Input: A query s_q and a source corpus $\mathcal{S}$
Output: A set of longest common n-gram matches $\mathcal{M}$ that covers s_q

$\mathcal{M} \leftarrow \emptyset$;
$l_q \leftarrow$ length of query s_q;
$l_{ngr} \leftarrow 0$, where l_{ngr} is the length of current matched n-gram;
for $start \leftarrow$ ***from*** 0 ***to*** l_q **do**
  if $l_{ngr} \neq 0$ **then**
    decrement l_{ngr};
  end
  $end \leftarrow l_q$;
  while $end \geq start + 1 + l_{ngr}$ **do**
    $m \leftarrow slice(s_q, start, end)$;
    if m *occurs in corpus* $\mathcal{S}$ **then**
      $\mathcal{M}.add(m)$;
      $l_{ngr} \leftarrow end - start$;
      break;
    end
    decrement end;
  end
end
return $\mathcal{M}$

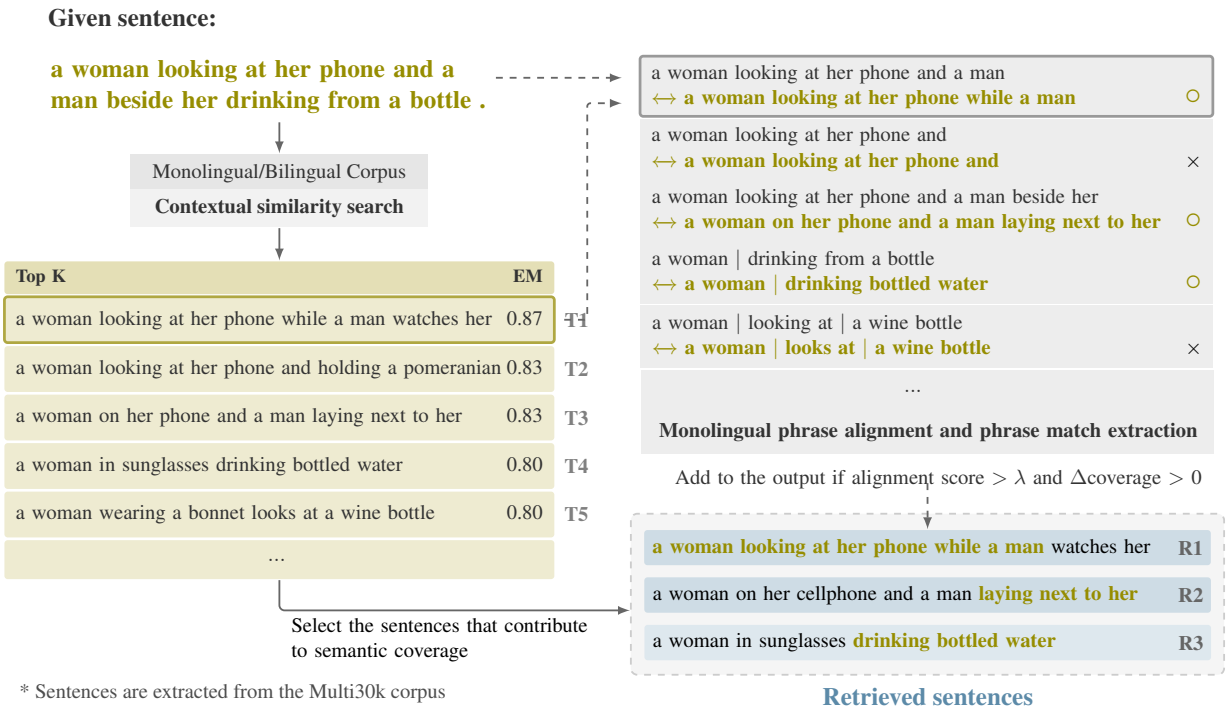


Figure 4: Overview of the process for semantic coverage.

Matching n-grams We start by matching the longest common n-grams that cover a given sentence in an iterative way by performing n-gram matching for each n-gram of the given sentence. Algorithm 1 implements this matching process. It ensures maximal formal coverage of the given sentence and a minimum number of common n-grams. Note that partial overlappings are allowed for the matched n-grams. For the n-gram without any match in the corpus, we derive subword tokens from them for subsequent processes.

Fuzzy matching selection After matching the covering n-grams, we retrieve all the sentences that contain each n-gram and subword from the source corpus. They consist in a certain number of groups of candidates to be trimmed. These groups are sorted in descending order of n-gram score. To reduce the number of candidates, we use the fuzzy matching score (see Section 2.1) to rank each group of candidate sentences and select the sentence with the highest score.

Trimming redundancies As sentence matching is separated from n-gram matching, the retrieved sentences tend to over-cover the query, despite the fact that the matched n-grams properly cover the given sentence. Sentences which are higher-ranked may contain common n-grams that match the sentences ranked lower. We trim these redundant sentences in a last phase.

3.2 Coverage in meaning

The process for semantic coverage is illustrated in Figure 4 and detailed as follows.

Retrieving similar sentences We search for the sentences most similar to the input sentence using the distributed method detailed in Section 2.3. Here we use the pre-trained sentence-BERT model to represent sentences with vectors. Efficient semantic search of a sentence vector space is facilitated by the Faiss library[2] (Johnson et al., 2019).

Similar phrase match extraction From the retrieved similar sentences, we attempt to extract phrase matches using the monolingual phrase alignment approach proposed in (Yoshinaka et al., 2020), particularly the phrase extraction module. This

[2] https://github.com/facebookresearch/faiss

method delivers word alignments based on a matrix of cosine similarity between pre-trained word embeddings, from which candidate phrase matches are extracted using the phrase alignment heuristic in (Koehn et al., 2007).

Screening candidate phrase matches So as to reduce the number of covering retrieved sentences, we adopt an intuitive procedure. Phrase match candidates are sorted by the rank of the sentences containing these phrases. From these candidates, we select those with an alignment score larger than a threshold and which contributes to the increase of coverage. This selection process is performed on the phrase match candidates in descending order of contextual similarity score, so that the coverage accumulation starts from the most similar sentence. This mechanism reduces the number of retrieved sentences as the most similar sentence tends to cover a larger piece of information in the given sentence.

4 Experimental setup

4.1 Datasets

We use 3 different corpora in our experiments: parallel corpus Multi30k (Elliott et al., 2016), the News Commentary Corpus (Tiedemann, 2012) and sentences from the Tatoeba corpus[3]. The language pairs used are Czech $\leftrightarrow$ English, German $\leftrightarrow$ English, French $\leftrightarrow$ English. The Multi30k corpus contains multilingual image descriptions for multilingual and multimodal research. The News Commentary Corpus is a collection of translation examples for training machine translation systems. The Tatoeba sentences and their translations are from a collaborative online database. Some statistics for each corpus are given in Table 2. We extracted 1,000 sentence pairs from each corpus as input sentence pairs. In particular, we used the English sentences as the query sentences.

The languages we tested our proposal on are all written with the Latin script. However, because formal retrieval uses suffix arrays, it can be applied with any kind of script. It will match at the character level, not below, something which might be wanted, for example, for Korean. It is indifferent to the direction of writing, and thus applicable without modification to scripts like the Arabic script, from right to left. As for semantic retrieval, a segmentation might be required for some scripts in advance so as to decompose sentences into words, so as to obtain their vector representations in the FastText models. This might be the case for languages like Thai, Chinese, Korean or Japanese.

[3] https://tatoeba.org/

4.2 Baselines and proposed systems

We compare our proposal with four common approaches in similar sentence retrieval and one exact matching method concerning coverage:

(a) matching sentences by the Jaccard similarity between sets of tokens in sentences, i.e., the cardinality of the intersection divided by the cardinality of the union of two sets;
(b) fuzzy matching, as described in Section 2.1;
(c) n-gram matching, as described in Section 2.2;
(d) matching sentences by contextual similarity, as described in Section 2.3;
(e) simply matching each token of the input sentence.

Implementation details are shown in Table 1.

For (a), (b), (c) and (d), sentences are usually retrieved when the match score is greater than a threshold. To ensure comparability between approaches under the context of maximising coverage and minimising redundancy, we limit the number of retrieved sentences instead of setting a constraint with a threshold. The difference between Cov_{tok} and Cov_{phr} is that for Cov_{tok}, the phrase alignment process is excluded in the phrase match extraction and aligned word pairs are treated as candidates.

4.3 Bilingual setting

We perform experiments on the bilingual corpora detailed in Section 4.1. These corpora contain pairs of sentences, i.e., each source sentence is aligned with a target sentence in another language. When we retrieve a group of source sentences to cover a given source sentence, a corresponding group of target sentences is indirectly retrieved. We assess how much they cover the target sentence aligned with the given sentence. We thus evaluate both the source and target coverage because we aim at applying our method in the framework of example-based machine translation.

	Match Unit	Embedding Model	Match Limit	Coverage Feature
JaccardSim_{10}	token	-	10	×
NgramMatch_{10}	n-gram	-	10	×
FuzzyMatch_{10}	token	-	10	×
$\text{ContextSim}_{\text{ft10}}$	sentence	Averaged FastText	10	×
$\text{ContextSim}_{\text{bt10}}$	sentence	Sentence-BERT	10	×
NaïveCov	token	-	-	○
Cov_{tok}	n-gram/token	Sentence-BERT	-	○
Cov_{phr}	n-gram/phrase	Sentence-BERT	-	○

Table 1: Implementation details of baselines and proposed methods. Match unit is the units compared in matching processes. Match limit is the fixed number of retrieved sentences for each query.

	Language	Avg. Length (en)	Vocab. Size (en)	Sentences
Multi30k	cs-de-en -fr	13	9,781	30,014
Tatoeba	de-en	7	25,585	229,205
	fr-en	7	24,052	220,608
News Commentary	cs-en	21	51,372	239,932
	de-en	21	58,417	327,817
	fr-en	22	57,347	316,398

Table 2: Statistics of the used datasets

5 Evaluation

As the central notion of coverage is recall, we evaluate the formal recall and semantic recall at different levels of granularity.

5.1 Formal coverage

We evaluate the recall of the words and subwords in a sentence. Sentence tokenization is facilitated by the SentencePiece [4] toolkit (Kudo, 2018).

We use BLEU (Papineni et al., 2002) as a rough evaluation of the recall of the n-grams in a sentence to be covered. This is justified by the fact that the BLEU score is the geometric mean of the probability of n-grams in the hypothesis to be present in the references (multiplied by some brevity penalty). In our work, the hypothesis is the input sentence and the references are the retrieved sentences.

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

5.2 Semantic coverage

We use the F1 and R values of BERTScore (Zhang et al., 2019) to evaluate the sentences retrieved. Strictly speaking, these values do not represent semantic coverage as BERTScore only scores the most similar sentence. We consider concatenating the sequences of token embeddings of the retrieved sentences, to extend greedy matching of tokens from one reference to multiple references. We evaluate the R value of this concatenated BERTScore, which arguably implies semantic coverage.

5.3 Normalisation

As one of our goals is to reduce the number of output sentences as much as possible, we simply normalise the coverage evaluation metrics by scaling each metric with an inverse ratio to the number of retrieved sentences. We define the normalised score as:

$$\text{normalised score} = \frac{\text{score}}{|\mathcal{S}_r|} \times 10 \tag{4}$$

where $|\mathcal{S}_r|$ denotes the number of retrieved sentences. The normalised result represents the extent of the coverage achieved by a certain amount of retrieved sentences. It makes a balance between the extent of coverage and the reduction of redundancy.

6 Results

We evaluate our results according to the three objectives that we want to achieve for the ultimate goal of use in an example-based machine translation system,

The first one is to ensure a high coverage when used in a bilingual context, i.e., we check whether

Method	Avg. Match	Source Coverage						Target Coverage					
		Formal Coverage			Semantic Coverage			Formal Coverage			Semantic Coverage		
		R_{word}	R_{sub}	BLEU	F1	R	$R_{cat.}$	R_{word}	R_{sub}	BLEU	F1	R	$R_{cat.}$
$JaccardSim_{10}$	10	84.02	83.13	38.23	60.23	57.36	70.94	72.39	76.43	27.88	54.38	51.76	63.40
$NgramMatch_{10}$	10	66.84	69.21	29.32	51.80	49.86	61.07	63.01	68.93	21.50	46.02	45.29	55.50
$FuzzyMatch_{10}$	10	79.17	78.81	38.40	59.59	56.72	69.83	69.67	73.91	27.24	53.67	51.46	62.64
$ContextSim_{ft10}$	10	67.99	68.73	23.43	54.97	48.80	60.00	60.04	64.24	18.38	48.58	43.97	53.20
$ContextSim_{bt10}$	10	74.43	77.26	24.87	57.16	54.81	67.20	68.66	75.29	23.46	51.80	50.29	61.25
$JaccardSim_{12}$	12	85.24	84.42	39.63	**60.57**	57.83	72.03	74.19	78.38	29.11	**54.88**	52.39	64.75
$NgramMatch_{12}$	12	68.40	70.76	30.80	52.23	50.26	62.13	64.61	70.74	22.27	46.53	45.76	56.69
$FuzzyMatch_{12}$	12	80.45	80.17	39.53	59.85	56.97	70.81	71.28	75.72	28.34	54.14	51.96	63.87
$ContextSim_{ft12}$	12	69.75	70.61	24.57	55.68	49.61	61.47	62.06	66.47	19.38	49.40	44.84	54.77
$ContextSim_{bt12}$	12	87.11	86.58	37.62	57.16	58.41	72.12	70.64	77.19	24.65	52.38	50.87	62.57
NaïveCov	12	**98.40**	98.21	40.61	59.21	58.11	77.76	75.01	81.36	26.85	52.62	51.57	65.80
Cov_{tok}	12	98.33	**98.58**	**54.39**	59.95	**59.10**	**80.72**	**83.78**	**89.26**	**34.01**	54.49	**53.69**	**71.32**
Cov_{phr}	10	97.98	98.13	53.79	59.96	59.04	80.27	82.23	87.83	32.84	54.13	53.30	70.31

Table 3: Results of source coverage evaluation (en) and target coverage evaluation (ce, de, fr). Recall and F1-scores are given in percentage.

Method	Source Coverage						Target Coverage					
	Formal Coverage			Semantic Coverage			Formal Coverage			Semantic Coverage		
	R_{word}	R_{sub}	BLEU	F1	R	$R_{cat.}$	R_{word}	R_{sub}	BLEU	F1	R	$R_{cat.}$
$JaccardSim_{10}$	84.02	83.13	38.23	**60.23**	57.36	70.94	72.39	76.43	27.88	**54.38**	51.76	63.40
$NgramMatch_{10}$	66.84	69.21	29.32	51.80	49.86	61.07	63.01	68.93	21.50	46.02	45.29	55.50
$FuzzyMatch_{10}$	79.17	78.81	38.40	59.59	56.72	69.83	69.67	73.91	27.24	53.67	51.46	62.64
$ContextSim_{ft10}$	67.99	68.73	23.43	54.97	48.80	60.00	60.04	64.24	18.38	48.58	43.97	53.20
$ContextSim_{bt10}$	74.43	77.26	24.87	57.16	54.81	67.20	68.66	75.29	23.46	51.80	50.29	61.25
$JaccardSim_{12}$	71.03	70.35	33.03	50.48	48.19	60.03	61.83	65.32	24.26	45.73	43.66	53.96
$NgramMatch_{12}$	57.00	58.97	25.67	43.52	41.88	51.78	53.84	58.95	18.56	38.77	38.13	47.24
$FuzzyMatch_{12}$	67.04	66.81	32.94	49.88	47.47	59.01	59.40	63.10	23.62	45.12	43.30	53.22
$ContextSim_{ft12}$	58.12	58.84	20.47	46.40	41.34	51.22	51.72	55.39	16.15	41.17	37.37	45.64
$ContextSim_{bt12}$	72.59	72.15	31.35	47.63	48.67	60.10	58.87	64.33	20.54	43.65	42.39	52.14
NaïveCov	82.00	81.84	33.84	49.34	48.43	64.80	62.51	67.80	22.38	43.85	42.98	54.83
Cov_{tok}	81.94	82.15	45.32	49.96	49.25	67.27	69.82	74.38	28.34	45.41	44.74	59.43
Cov_{phr}	**97.98**	**98.13**	**53.79**	59.96	**59.04**	**80.27**	**82.23**	**87.83**	**32.84**	54.13	**53.30**	**70.31**

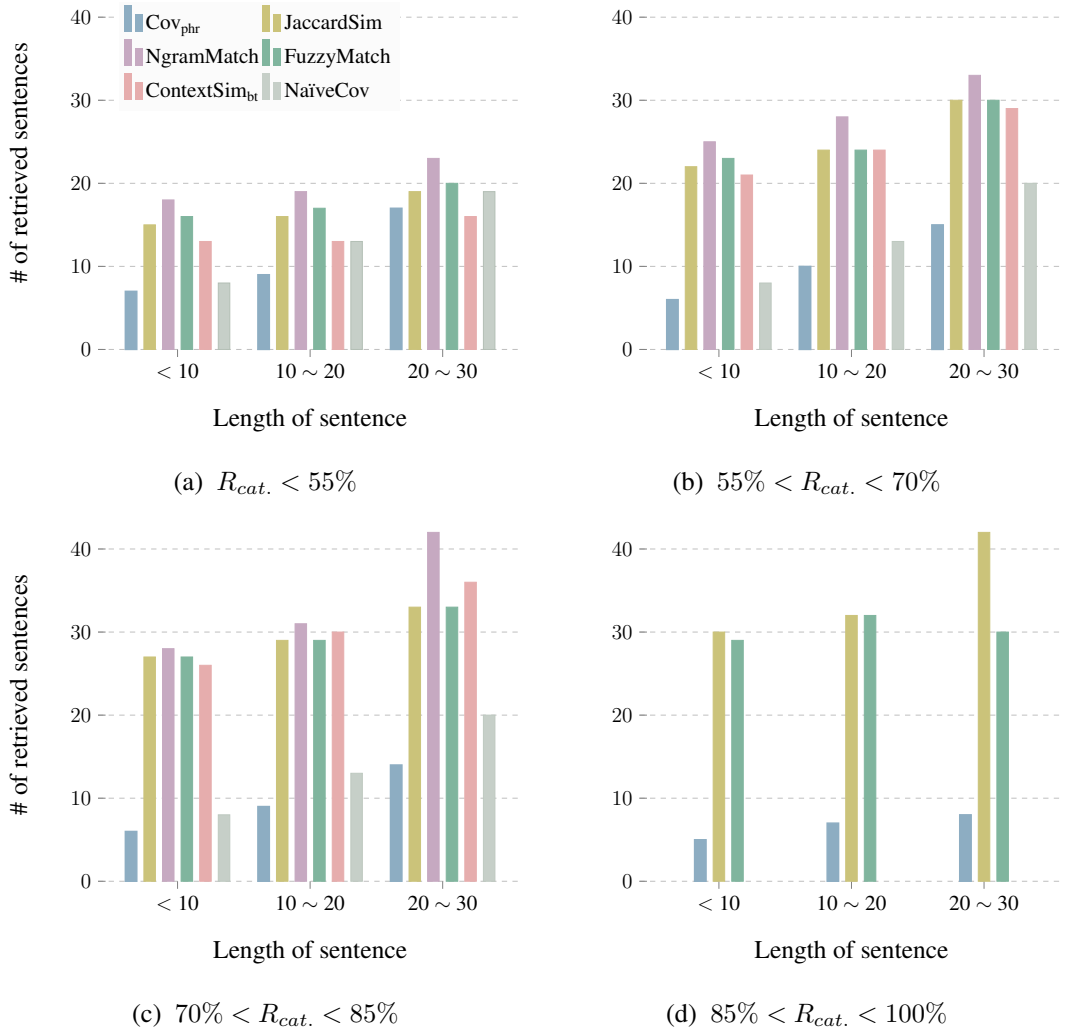
Table 4: Normalised results of source coverage evaluation (en) and target coverage evaluation (cs, de, fr).

we obtain a high coverage in the source and target languages.

Our second objective is to obtain a high coverage relatively to the length of the input sentence, i.e., we check whether we obtain a high normalised coverage.

Our third objective is to reduce the number of retrieved sentences, i.e., we check whether we achieve coverage with as little redundancy as possible.

High source and target coverage Table 3 gives the evaluation results. The difference between source and target evaluation was described in Section 4.2. The methods focusing on coverage reach high R_{word} and R_{sub} scores at around $98\,\%$ in the source coverage evaluation. In both the source and target results, Cov_{tok} performs best in R_{word}, R_{sub}, BLEU and $R_{cat.}$. In particular, for BLEU, our proposed methods Cov_{tok} and Cov_{phr}, outperform other methods by more than $30\,\%$. This indicates a high



(a) $R_{cat.} < 55\%$ (b) $55\% < R_{cat.} < 70\%$

(c) $70\% < R_{cat.} < 85\%$ (d) $85\% < R_{cat.} < 100\%$

Figure 5: Comparison of the number of sentences retrieved by $\mathrm{Cov_{phr}}$ and baselines for fixed ranges of semantic coverage. In (d), only $\mathrm{Cov_{phr}}$, JaccardSim and FuzzyMatch provide enough sample data for the semantic coverage of 85% to 100%. The number of retrieved sentences shown in the figure is the average number of the sentences retrieved for each given sentence of certain range of length.

level of n-gram coverage. Compared to n-gram matching, fuzzy-matching and contextual similarity on which our proposed methods are based, our mechanism of retrieving sentences shows considerable improvement in formal and semantic coverage.

Compared to source coverage results, target coverage results show a certain decrease due to the indirect retrieval. But our proposed methods tend again to perform better in the target coverage evaluation. This opens up the possibility to integrate our approach into an example-based machine translation system.

High normalised coverage Table 4 shows the normalised results. $\mathrm{Cov_{phr}}$ scores best in almost all the evaluation metrics, surpassing $\mathrm{JaccardSim_{10}}$ by over 10 % in R_{word}, R_{sub}, BLEU and $\mathrm{R}_{cat.}$. These values render an account of both formal and semantic coverage, as mentioned in Sections 5.1 and 5.2. $\mathrm{Cov_{phr}}$ does not reach the highest score in F1 due to a trade-off between recall and precision in the retrieved sentences, i.e., individual retrieved sentences with a high recall tend to include more irrelevant information.

Low redundancy Figure 5 shows the comparison of the number of sentences retrieved by Cov_{phr} and the baselines for some fixed ranges of semantic coverage. Cov_{phr} basically retrieves fewer sentences for the given sentence of any length and for different ranges of semantic coverage. The number of retrieved sentences increases with the increase of the length of the given sentence and with the decrease of semantic coverage. The reason is that a given sentence which is difficult to cover, usually results in a larger number of retrieved sentences and a smaller coverage by the retrieved sentences.

As shown by the tables and figures of results, a small number of sentences retrieved by Cov_{phr} reach higher scores in both coverage evaluation and normalised evaluation. This indicates that the sentences retrieved by Cov_{phr} are of higher formal and semantic coverage, and that they exhibit lower redundancy.

7 Conclusion

We proposed a novel approach to retrieve a group of sentences that cover a sentence in both aspects of form and meaning, using techniques from similar sentence retrieval and monolingual phrase alignment. The evaluation results show that our proposal achieves the two-fold goal of maximising formal and semantic coverage while delivering fewer retrieved sentences.

In future work, we want to integrate our retrieval system into an example-based machine translation system, like the one described in (Taillandier et al., 2020) where experiments were conducted in a setting where retrieval was left out. Another system in which we intend to integrate our retrieval system is an academic writing aid system, where we want to provide a module for similar sentence recommendation. The task is to help researchers who are non-native in English in writing scientific papers.

Acknowledgments

This research was supported in part by two grants-in-aid from the Japanese Society for the Promotion of Science (JSPS): n° 18K11447 entitled "Self-explainable and fast-to-train example-based machine translation using neural networks" and n° 18K11446 entitled "Natural language processing for academic writing in English" .

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bram Bulté and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *57th Annual Meeting of the Association-for-Computational-Linguistics (ACL)*, pages 1800–1809.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74.

Heikki Hyyrö. 2001. Explaining and extending the bit-parallel approximate string matching algorithm of myers. Technical report, Dept. of Computer and Information Sciences, University of Tampere, Tampere, Finland.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Philippe Langlais, François Yvon, and Pierre Zweigenbaum. 2009. Improvements in analogical learning: application to translating multi-terms of the medical domain. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 487–495.

Yves Lepage. 2019. Semantico-formal resolution of analogies between sentences. In *Proceedings of the 9th Language and Technology Conference (LTC 2019)*, pages 57–61.

Yves Lepage and Etienne Denoual. 2005. ALEPH: an EBMT system based on the preservation of proportional analogies between sentences across languages. In *International Workshop on Spoken Language Translation (IWSLT) 2005*.

Makoto Nagao. 1984. A framework of a mechanical translation between japanese and english by analogy principle. *Artificial and human intelligence*, pages 351–354.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.

Nicolas Stroppa and François Yvon. 2005. Analogical learning and formal proportions: Definitions and methodological issues. *ENST Paris report*.

Valentin Taillandier, Liyan Wang, and Yves Lepage. 2020. Réseaux de neurones pour la résolution d'analogies entre phrases en traduction automatique par l'exemple. In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 31e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition)*, volume 2 : Traitement Automatique des Langues Naturelles, pages 108–121. AFCP et ATALA.

Arda Tezcan, Bram Bulté, and Bram Vanroy. 2021. Towards a better integration of fuzzy matches in neural machine translation through data augmentation. *Informatics*, 8(1):7.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Jitao Xu, Josep M Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590.

Masato Yoshinaka, Tomoyuki Kajiwara, and Yuki Arase. 2020. Sapphire: Simple aligner for phrasal paraphrase with hierarchical representation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6861–6867.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, pages 177–180.