A Study of Re-generating Sentences Given Similar Sentences that Cover Them on the Level of Form and Meaning

Hsuan-Wei Lo, Yifei Zhou, Rashel Fam, Yves Lepage
Graduate School of Information, Production and Systems
Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
hsuanweilo@akane.waseda.jp, yifei.zhou@ruri.waseda.jp,
fam.rashel@fuji.waseda.jp, yves.lepage@waseda.jp

Abstract

In this paper, we define a sentence re-generation task: re-generate a sentence given a set of sentences that cover it. Due to the absence of a dataset to perform this task, we firstly build three language resources of a new type containing more than 4 million annotated sentences. They contain sentences annotated with similar sentences from the same corpus, that cover them on the level of form or meaning. We then perform the sentence re-generation task on the newly produced language resources using two approaches. The first one is a naïve approach where we rely on a language model to reorder the covering parts. The second one is a neural approach where we treat the sentence re-generation task as a translation task from a sequence of covering parts to the respective original sentence. The performance of the systems is evaluated on the level of form and meaning according to the type of covering used to re-generate the sentence. On the level of form, experimental results show that the neural approach outperforms the baseline in edit distance with up to 40% lower scores. However, in BLEU scores, the neural approach is similar or worse than the baseline. On the level of meaning, the neural approach always performs better than the baseline with average scores of 89% BERTScore.

1 Introduction

Translation memories (TM) are used by translators to retrieve close sentences to a given source sentence as hints for translations. It has been shown that similar sentences or informative sentences retrieved from TM can significantly boost the performance of neural machine translation (NMT) systems (Xu et al., 2020; Bulté and Tezcan, 2019a). The idea of a set of sentences covering a given input sentence can be introduced to combine TM with machine translation. (Liu and Lepage, 2021) proposed a twofold-objective approach to retrieve similar sentences based on an input sentence.

To assess the quality of sentence coverage, we define a sentence re-generation task:

> Re-generate an input sentence back given a set of sentences that cover it on two levels, form or meaning.

Yet, there is a lack of data containing for this specific task. The needed data should contain sentences similar in form and meaning along with the covering parts annotated for each input sentence from some corpus. Therefore, we build and release such language resources to meet the requirement of this new task. We then test a naïve baseline and a neural approach on the newly built resources to perform the sentence re-generation task. The two above aspects are the contributions of our work.

The paper is organized as follows. Section 2 explains the notion of coverage of a sentence by other sentences. Section 3 describes the new language resources created. Section 4 explains the two approaches to the sentence re-generation task and the experiment protocol. Section 5 presents the experimental results and provides an analysis. Section 6 gives the conclusion and possible future directions.

Input	*two young toddlers outside on the grass.*
Form	*two white bunnies are **outside on the grass**.*
	***two young** girls playing outside on the playground.*
	*two **toddlers** posing for the camera.*
Meaning	*there is a toddler playing **on a** playground.*
	*two kids are laughing **in the grass**.*
	*a baby is sitting **outside in grass**.*
	*a man **and a young toddler are playing outside in the grass**.*

Figure 1: An input sentence (Input) and the list of sentences that cover it in form (Form) and meaning (Meaning). All sentences are from the English part of Multi30K.

2 Coverage of a Sentence in Form and Meaning

To produce the newly built released language resources[1], we use a tool for the retrieval of sentences similar to an input sentence that has the claimed twofold objective of

- maximising the coverage of the input sentence in both form and meaning,
- while minimising the number of retrieved sentences.

The tool is implemented as a Python package (Liu and Lepage, 2021). Basically, the input is a sentence (the input sentence) and a corpus, and the output is two lists of sentences extracted from the corpus which are similar to the input sentence. The two output lists are:

- a list of sentences similar in form to the input sentence that cover it as much as possible, and
- a list of sentences similar in meaning to the input sentence that cover it as much as possible.

The lengths of these two lists are not necessarily equal.

2.1 Coverage of a Sentence in Form

By covering an input sentence in form, we mean that the input sentence is covered by sequences of words found in the sentences from the list of sentences output by the retrieval process. The tool ensures coverage by recursive sub-string matching. A sentence with the longest possible sub-string in common with the input sentence is first retrieved. Then, the remaining parts of the input sentence are explored recursively by the same retrieval procedure.

The sentences marked Form in Figure 1 illustrate this. The input sentence is the English phrase *two young toddlers outside on the grass.* The corpus is the English part of Multi30K. The longest sub-string in common with another sentence in the corpus is ***outside on the grass***, found in the sentence *two white bunnies are outside on the grass.* The process then further retrieves sentences that have sub-strings in common with the remaining parts of the input sentence. Sentences with the longest possible sub-strings in common are first retrieved. This explains why the next sentence contains a sub-string of two words in common with the input sentence (***two young***), while the third one has only a sub-string of one word in common (***toddlers***).

An almost total coverage of the input sentence is obtained by combining the sub-strings. Indeed the coverage is not total, as the full stop is not covered. All together, but in a different order, the three sub-strings ***outside on the grass***, ***two young***, and ***toddlers*** make back the entire input sentence *two young toddlers outside on the grass* (again without the full stop).

2.2 Coverage of a Sentence in Meaning

By covering a sentence in meaning, we mean finding portions of the input sentences that are semantically close to sentences from the corpus. Semantic closeness or similarity is achieved through vector cosine similarity of sub-word, word, or word sequence embeddings.

The four sentences marked Meaning in Figure 1

[1] `http://lepage-lab.ips.waseda.ac.jp` → Projects → Kakenhi Kiban C 18k11447 → Experimental data

are examples of covering sentences for the same input sentence as in the previous section. The substrings ***on a***, ***in the grass***, and ***outside in grass*** cover *on the grass* in the input sentence. Notice that semantic similarity does not necessarily mean synonymy.

3 Data

We use three corpora that contain diverse topics that may lead to different performance. For each sentence in the three corpora, we retrieve covering sentences from the same corpus. This is performed, of course, after removing the input sentence, so as to prevent the input sentence from being retrieved. The sentence coverage in the newly produced language resources is shown by annotating the parts in common, in form or in meaning, using an XML-like tags format.

3.1 Original Corpora Used to Produce the Released Language Resources

We use the following three corpora to produce three language resources:

- **Multi30K**[2] (Elliott et al., 2016; Elliott et al., 2017; Barrault et al., 2018) is a collection of image descriptions (captions). This corpus is available in four languages: Czech, English, French and German. In this work, we only use the English part of the corpus. This corpus is heavily used in multilingual image description and multimodal machine translation tasks.

- **Tatoeba**[3] is a collection of sentences in over 100 languages. The number of sentences per language ranges from 10 to over 100,000 sentences per language. In this work, we only consider the English part of the corpus.

- **ACL ARC**[4] (Bird et al., 2008) is a corpus that contains academic articles in English, in the field of computational linguistics. It is a collection of published papers in conferences associated with or organised by the Association for Computational Linguistics (ACL).

Table 1 shows the statistics of the above corpora. The ACL ARC corpus contains around 2.5 million sentences, while the number of sentences that we use from Tatoeba is a little bit more than 1.5 million. Multi30K has a relatively small number of sentences in comparison to ACL ARC and Tatoeba: 30,000, as the name says. The vocabulary of ACL ARC is 100 times larger than that of Multi30K. Tatoeba has the shortest average length of sentences in comparison with the other two corpora. ACL ARC also has around two times longer sentences in both, words and characters, compared with Multi30K.

Looking deeper into the use of words, we observe the following phenomena. In types as well as in tokens, words in ACL ARC are around two times longer than in Multi30K. ACL ARC has a high ratio of hapaxes with more than 60%, in comparison to Multi30K with just above 40% and Tatoeba with a little bit more than 45%. These phenomena are mostly caused by the characteristics of the sentences contained in the corpora themselves. Academic articles are more likely to contain longer and more specialized terms. In contrast, in Multi30K, which contains image captions, sentences are shorter and contain more frequent words, which are in trend, shorter than scientific terms.

3.2 Format of the Released Language Resources

The format of the released language resources follows standard practice in Natural Language Processing, where language pieces appear as raw data, separated by tabulations and with annotations by XML-like tags. Each resource consists of a unique file containing a sequence of sentences on each line. On each line, the input sentence comes first, then the list of sentences for coverage in form, and finally the list of sentences for coverage in meaning. Each sentence is separated from the next one by a tabulation.

The sub-strings or parts in common with the input sentence in the retrieved sentences are identified by XML-like tags. There is only one tag used: `coverage`. It takes one attribute `type` with two possible values: `formal` and `semantic`. For instance, the first sentence in the Form part of Figure 1 appears in the language resource produced from the English part of the Multi30K corpus as follows:

[2] `https://github.com/multi30k/dataset`

[3] `https://tatoeba.org`

[4] `https://catalog.ldc.upenn.edu/docs/LDC2009T29/lrec_08/`

	Multi30K	Tatoeba	ACL ARC
# of sentences	30,014	1,519,509	2,491,483
# of tokens	390,843	11,561,489	57,585,605
# of types	10,376	160,454	1,083,298
Avg. token length	3.87±2.40	4.26±2.19	5.34±3.14
Avg. type length	6.93±2.41	8.06±2.74	8.76±4.28
Type-Token-Ratio	0.03	0.01	0.02
Hapax ratio (%)	41.94	47.24	62.55
# of char./sent.	62.38±20.37	39.03±23.52	145.48±73.26
# of words/sent.	13.19±4.17	9.55±4.97	27.45±14.06

Table 1: Statistics on the original data (the average values are given with standard deviation after the ± sign).

	per input sentence on average	Multi30K	Tatoeba	ACL ARC
Form	# of retrieved sent.	5.43±2.21	3.29±2.32	5.79±3.77
	# of char./sent.	61.9±20.54	44.39±37.33	160.99±78.29
	# of words/sent.	13.51±4.18	10.56±7.54	29.96±14.67
	# of char. in coverage	17.97±13.33	16.19±15.65	35.50±46.24
	# of words in coverage	3.91±2.91	3.87±3.49	6.71±8.75
	Individual coverage (%)	32.08	44.71	27.42
	Cumulative coverage (%)	87.50	85.48	63.14
Meaning	# of retrieved sent.	2.05±1.21	1.52±0.82	2.70±1.42
	# of char./sent.	59.18±19.82	36.90±26.27	148.29±76.96
	# of words/sent.	12.84±4.09	9.18±5.38	27.60±14.44
	# of char. in coverage	33.57±23.49	20.14±16.87	95.19±92.52
	# of words in coverage	7.24±4.77	4.51±3.60	18.19±17.04
	Individual coverage (%)	86.58	82.43	82.93
	Cumulative coverage (%)	89.42	89.52	84.91

Table 2: Statistics on the data produced (the average values are given with standard deviation after the ± sign).

```
two white bunnies are
<coverage type=
"formal">outside on the
grass.</coverage>
```

Although the order of retrieved sentences in the released language resources is fixed as mentioned above, the order is theoretically free, because the values of the attribute `type` give the type of coverage.

3.3 Ratio of Coverage

The coverage of a sentence is the length of the sequence of words that is similar to the input sentence. We find that the average length of semantic coverage is higher than that for formal coverage. This is indicated in Table 2 by the rows # of char. in coverage and # of words in coverage. For the Tatoeba corpus, the average length of semantic coverage is 1.2 times that of formal coverage, while this ratio is 2 for Multi30K and roughly 3 for ACL ARC.

3.3.1 Ratio of Coverage in Form

We measure the ratio of coverage in form by counting the number of identical word sequences in the retrieved sentences covering the input sentence. There are two kinds of ratios of coverage per input sentence: individual coverage and cumulative coverage. The individual coverage computes the length of coverage of a retrieved sentence against the length of the input sentence. We report the average individual coverage over all retrieved sentences. In contrast to the average individual coverage, cumulative coverage is the ratio of the sum of the lengths of all substrings in the retrieved sentences over the length of the input sentence. Table 3 illustrates these ratios on several example sentences.

Table 2 shows that the average individual coverage in form is higher for Tatoeba than Multi30K and ACL ARC. This is not surprising as Tatoeba is known to be repetitive. In terms of cumulative coverage, an input sentence can be almost 85% covered by the retrieved sentences in Multi30K and Tatoeba, whereas it is only covered by 63% in ACL ARC.

3.3.2 Ratio of Coverage in Meaning

To compute the similarity between the input sentence and its retrieved similar sentences in meaning, we use the F1 value of BERTScore[5] (Zhang et al., 2020). The individual coverage in meaning is defined as the BERTScore of the semantic coverage per retrieved sentence with the input sentence. Furthermore, we use the concatenation of the sequences of coverage in all retrieved sentences to calculate the cumulative coverage.

Table 2 shows that the ratio of coverage is up to 80% in all three corpora, Multi30K, Tatoeba, and ACL ARC. Some example results for the calculation of coverage ratio in meaning are shown in Table 3.

4 Experiments

We carry out experiments on sentence re-generation with the newly produced language resources introduced in Section 3. For each resource, we divide the dataset into training and test sets, 90% and 10% respectively. Thus, we have a training set and a test set for each Multi30K, Tatoeba, and ACL ARC.

To illustrate the task, let us look at the example in Figure 1. The task consists in re-generating the sentence marked Input, from the only given of the three sentences marked Form or the four sentences marked Meaning. The covering parts are shown in boldface and are indicated by tags in the raw dataset.

We consider two approaches to address this task. The first one is a naïve approach which uses a language model to perform reordering of the covering parts. This is our baseline. The second one is the neural approach where we treat the sentence re-generation task as a translation task from the covering parts into the original sentence. The performance is evaluated separately according to the type of coverage used. Therefore, we have a distinct evaluation on each of the levels of form and meaning.

4.1 Baseline

As said above, the baseline, which is a naïve approach, just reorders the covering parts. It first extracts the covering parts in the retrieved sentences based on the tags (see Section 3.2). The covering parts are ordered in all possible permutations. From this set of all possible permutations, we select the output as being the permutation with the lowest perplexity according to a language model.

[5] https://github.com/Tiiiger/bert_score

Multi30K		Coverage (%)		
		indiv	avg	cum
Input	*two young toddlers outside on the grass.*			
Form	*two white bunnies are* ***outside on the grass****.*	50.00	29.17	87.50
	two young *girls playing outside on the playground.*	25.00		
	two ***toddlers*** *posing for the camera.*	12.50		
Meaning	*there is a toddler playing* ***on a*** *playground.*	82.97	90.12	90.32
	two kids are laughing ***in the grass****.*	92.51		
	a baby is sitting ***outside in grass****.*	90.89		
	a man ***and a young toddler are playing outside in the grass****.*	94.10		

Table 3: Example results for coverage ratio on the three corpora. The individual coverage (indiv) is for each individual sentence. The average coverage (avg) is the arithmetic mean over all individual coverage scores. The cumulative coverage (cum) is the coverage of all sub-strings in the retrieved sentences over the input sentence. Its maximal value is 100%.

Corpus	Approach	Accuracy (%)	Edit distance		# of chars per sent.	# of words per sent.	BLEU points
			in chars	in words			
Multi30K	Naïve	0.30	51.76	12.05	99.28	20.76	**43.60**
	Neural	**7.90**	**29.38**	**6.76**	55.06	12.01	39.61
Tatoeba	Naïve	0.44	33.83	8.37	58.47	12.80	45.05
	Neural	**24.64**	**19.70**	**4.65**	34.74	8.61	**45.57**
ACL ARC	Naïve	0.00	161.31	35.19	212.32	38.83	**23.53**
	Neural	**0.27**	**103.86**	**23.30**	103.47	20.34	8.76

Table 4: Evaluation on the level of form.

Corpus	Approach	# of chars per sent.	# of words per sent.	BERTScore (F1)
Multi30K	Naïve	69.41	14.82	0.86
	Neural	43.43	10.19	**0.91**
Tatoeba	Naïve	30.93	6.69	0.85
	Neural	28.14	7.95	**0.90**
ACL ARC	Naïve	287.11	54.98	0.84
	Neural	91.71	20.70	**0.86**

Table 5: Evaluation on the level of meaning.

4.1.1 Permutation of Covering Parts

We extract the covering parts from the retrieved sentences. These covering parts are to be found between the tags mentioned in Section 3.2. As an example, let us suppose that we have five sentences in the sentence coverage on the level of form for one input sentence. This gives five sub-strings between tags that we extract, which are the covering parts.

Carrying on with the situation of five retrieved sentences, we permute the five covering parts in all possible orders. This gives us 120 possible permutations, i.e., $n!$, where n is the number of covering parts, 5.

4.1.2 Selection by Language Model

We apply a language model to score all possible permutations. We use kenLM[6] (Heafield et al., 2013) for that, and use modified Kneser-Ney smoothing without pruning, for smoothing. In our experiments, we train the language model on the training set. We thus built 3 language models, one on each of the three different corpora: Multi30K, Tatoeba, and ACL ARC.

In our previous example of 120 combinations, applying the language model to each of the 120 combinations delivers a score for each of the combinations. We select the best combination, that is the one with the lowest score among the 120 combinations. Some examples of results are given in Table 6 for each of the three corpora.

4.2 Neural Approach

In a second, more modern and less naïve approach, we treat the sentence re-generation task as a translation task where:

- the source channel is the covering parts contained in the sentence coverage, and
- the target channel is the original sentence which we would like to re-generate.

Similar to the baseline, we only consider the covering parts as the input for the neural approach. Thus, we first extract the covering parts from the sentence coverage according to the tags (see Section 3.2). We then train the Transformer model on the training set.

4.2.1 Preprocessing Dataset

As mentioned above, we extract the covering parts from the retrieved sentences. The covering parts are concatenated as an input sequence to the neural approach. For example, in Figure 1, we use the covering parts on the level of form, "*outside on the grass*", "*two young*" and "*toddlers*" as an input sequence for the input sentence "*two young toddlers outside on the grass*". The source and target channels of the neural approach are shown as follows:

- source channel: "*outside on the grass two young toddlers*"
- target channel: "*two young toddlers outside on the grass.*"

All sentences are tokenized using SentencePiece[7] to break the sentences into sub-words. The subword model is known to improve the performance of the natural language generation systems (Kudo and Richardson, 2018). This tool is an unsupervised text tokenizer (encoding) for neural networks, especially in the text generation system.

4.2.2 Transformer: Open-NMT

We train a Transformer model provided in OpenNMT-py[8] (Klein et al., 2017) on each of the three language resources mentioned above to perform the sentence re-generation task. We keep the test set as it is (10%) and divide the original training set (90%) into 80% as a training set and 10% as a validation set to train our Transformer model. Next, we select the best-trained model to perform the sentence re-generation task in terms of the perplexity score given by the transformer model on the validation set.

5 Results and Analysis

We evaluate the performance of the baseline and the neural approach on the test sets on each of the three corpora. As mentioned in Section 4, the evaluation is performed on two levels, form and meaning, based on the type of coverage used. On the level of form, we use accuracy, edit distance, and BLEU scores as evaluation metrics. To measure the performance on the level of meaning, we use BERTScore. The overall results are given in Table 4 and Table 5.

[6] https://github.com/kpu/kenlm

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

[8] https://github.com/OpenNMT/OpenNMT-py

Corpus	Level	Approach	Sentence	BLEU	BERT-F1
Multi30K	-	-	*two young toddlers outside on the grass .*	-	
	Form	Naïve	*two young toddlers outside on the grass*	86.69	-
		Neural	*two young toddlers outside on the grass .*	100.00	-
	Meaning	Naïve	*on a and a young toddler are playing outside in the grass in the grass outside in grass*	-	0.91
		Neural	*a toddler is playing with a toy in the grass .*	-	0.93
Tatoeba	-	-	*I'd be unhappy, but I wouldn't kill myself.*	-	
	Form	Naïve	*but I would n't be unhappy, wouldn't kill I'd be unhappy, but I*	36.41	-
		Neural	*I'd be unhappy, but I wouldn't kill.*	79.56	-
	Meaning	Naïve	*but I don't intend to kill myself*	-	0.86
		Neural	*I don't intend to kill myself, but I don't want to kill myself.*	-	0.92
ACL ARC	-	-	*Single word may have different meanings under different situations.*	-	
	Form	Naïve	*under different situations Single word*	18.39	-
		Neural	*Sometimes different situations under different situations.*	28.32	-
	Meaning	Naïve	*have different meanings . g . word followed by comma) can also be addressed through truecasing .*	-	0.85
		Neural	*In other words, words may have different meanings.*	-	0.92

Table 6: Examples of sentences re-generated by the naïve baseline and neural approaches on the three corpora using coverage on the level of form and meaning. A grey background indicates the original sentence from the corpus, a white background is for the re-generated sentence.

5.1 Evaluation on the Level of Form

The first metric used to evaluate the performance of the system on the level of form is accuracy. Accuracy is defined as the proportion of exact match between the reference and the re-generated sentence (just a full stop missing counts as zero). The naïve baseline's performance is close to zero in terms of accuracy on all of the corpora: 0.30% on Multi30K, 0.44% on Tatoeba, and 0.00% on ACL ARC. There is no correct sentence re-generated by the baseline on the ACL ARC. However, the neural approach has higher accuracy than the baseline on the three corpora: 7.90% on Multi30K, 26.64% on Tatoeba, and 0.27% on ACL ARC. The difference is pretty high, particularly on Tatoeba.

A finer view is given by the use of the Levenshtein edit distance (Levenshtein, 1966)[9] to perform the formal evaluation. The Levenshtein edit distance involves three different operations: insertion, deletion, and substitution. Each operation counts as one. Table 4 gives the results of the application of the Levenshtein edit distance at two levels of granularity, that of characters and that of words. The edit distance of the baseline is very close to the average length of the original sentence. This means that almost all of the words in the re-generated sentence need to be modified. We also observe that the neural approach achieves around 40% lower edit distance than the baseline. Looking at some examples of the re-generated sentences in Table 6, the Transformer model has removed repeated words or grammatical errors such as punctuation. This makes re-generated sentences closer to the reference sentences in terms of edit distance.

We thus measure the extent to which groups of words might be in the correct order. To this end, we use BLEU (Papineni et al., 2002), in its SacreBLEU[10] (Post, 2018) implementation. BLEU evaluates the similarity between one or several original references, and a candidate sentence. Higher BLEU scores indicate higher similarity, with 100 being the maximum. Table 4 shows that the base-

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

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

line is able to obtain around 44 BLEU scores for Multi30k and Tatoeba, twice as much as for ACL ARC (23.53). BLEU scores of above 40 reflect the fact that some sequences of words are shared between the re-generated sentence and the reference sentence, i.e., not all words are scrambled in a completely different order. We also notice that the neural approach gets similar (Tatoeba) or lower (Multi30K and ACL ARC) BLEU scores than the baseline. This indicates that the Transformer model missed half of the portion of correct words in the correct position. For ACL ARC, the neural approach seems to change most of the content of the sentence into a completely different sentence which leads to a low BLEU scores (8.76).

5.2 Evaluation on the Level of Meaning

To evaluate the performance on the level of meaning, we use BERTScore (similar to what we did in Section 3.3.2). It computes the cosine similarity between pair-wise tokens in form of the re-generated sentence and the original sentence using a pre-trained BERT embedding model. BERTScore provides precision, recall, and F1, measured by the weighted maximum similarity scores. Here, we only consider the F1 score which is the harmonic mean of recall and precision. A value of 1 for the F1 score means that the meaning of the re-generated sentences and reference sentences are the same. Table 5 shows that F1 scores on Multi30K, Tatoeba, and ACL ARC are in the range of the mean of 0.87 to 0.92. This shows that the re-generated sentences are 85% semantically close to the original input sentences, according to BERTScore. Overall, the neural approach performs better than the baseline with an average score of 0.89. This does not necessarily show that our re-generated sentences are close to the reference sentence, as shown in Table 6.

5.3 Discussion

Our experimental results show a configuration where scores in accuracy close to zero and large Levenshtein edit distances are seemingly in contradiction with the reasonably high scores in BLEU and the excellent scores in BERTScore. Such an experimental configuration asks the question of what the adequate metrics are to reflect the fact that, although almost all the expected words are present, although some sequences of words are correct and match exactly the input sentence, as a whole, the re-generated candidate sentences produced by our naïve method are far away from the input sentences.

6 Conclusion

We defined a sentence re-generation task: re-generate an input sentence back given sentences that cover it on two levels, form and meaning. For this task, we built a new type of language resource produced from three different corpora: Multi30K, Tatoeba, and ACL ARC. Altogether, this represents over 4 million sentences annotated with similar sentences that cover them, and in which the covering parts are tagged. We released three resources.

We carried out experiments on this new type of resource using two approaches: a naïve approach of sequence reordering using a language model, and a neural approach that treats the sentence re-generation task as a translation task from covering parts to the original sentence. The experiments were performed on both the level of form and meaning. The performance of the systems was evaluated according to the type of coverage used. On the level of form, experimental results showed that the neural approach performed up to 40% better in terms of accuracy and edit distance. However, it performed similarly or lower in terms of BLEU. On the level of meaning, the neural approach achieved a higher BERTScore than the baseline by a margin of 4%.

For future work, as for the released language resources, their quality can be improved. A higher individual coverage percentage (mentioned in Section 3) is needed, particularly for the ACL ARC corpus, so that original sentences are enough covered. In addition, the calculation of coverage percentage on meaning needs to be revised since the average of 83% of individual coverage percentage was not reflected well in our evaluation on the level of meaning.

We also consider releasing similar resources for other languages than English. We believe that this new type of resource can be used for various types of NLP tasks such as language reference (Vossen et al., 2020), text generation (Nan et al., 2021), or the integration of translation memories with machine translation (Bulte and Tezcan, 2019b).

References

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology Reference Corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of LREC 2008*, Marrakech, Morocco, may. European Language Resources Association (ELRA).

Bram Bulté and Arda Tezcan. 2019a. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of ACL 2019*, pages 1800–1809.

Bram Bulte and Arda Tezcan. 2019b. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of ACL 2019*, pages 1800–1809, Florence, Italy, July. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 215–233, Copenhagen, Denmark, September. Association for Computational Linguistics.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of ACL 2013*, pages 690–696, Sofia, Bulgaria, August. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP 2018: System Demonstrations*, pages 66–71, Brussels, Belgium, November. Association for Computational Linguistics.

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

Yuan Liu and Yves Lepage. 2021. Covering a sentence in form and meaning with fewer retrieved sentences. In *Proceedings of PACLIC 35*, pages 1–10, November.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of NAACL 2021: Human Language Technologies*, pages 432–447, Online, June. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318, Philadelphia, July. Association for Computational Linguistics.

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

Piek Vossen, Filip Ilievski, Marten Postma, Antske Fokkens, Gosse Minnema, and Levi Remijnse. 2020. Large-scale cross-lingual language resources for referencing and framing. In *Proceedings of LREC 2020*, pages 3162–3171, Marseille, France, May. European Language Resources Association.

Jitao Xu, Josep M Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of ACL 2020*, pages 1580–1590.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proceedings of ICLR 2020*.