Paraphrase Conditioning Model: Final Report

G088 (s1740055, s1762811, s1706479)

Abstract

In this project, we studied the effectiveness of length ratio conditioning on paraphrase generation. We constructed two datasets. One used a pretrained NMT model to construct a new paraphrasing dataset from WMT 2019 Europarl German-English dataset, and the other was extracted from MSCOCO and QUORA question pairs dataset. We proposed tagging and fine-tuning methods on length ratio for conditioning paraphrase generation. Finally we evaluate length-ratio conditioning performance of models from semantic similarity and length change. The major finding is that tagging had little to no effect on conditioning tasks while having a better paraphrase performance. In comparison, however, fine-tuning is more effective in conditioning but showed poor performance in paraphrasing.

1. Introduction

Paraphrasing is an important part of NLP that automatically generates semantic equivalences by replacing various units of language, such as words, phrases and sentences (Madnani & Dorr, 2010). For example, "I prefer dogs to cats" is semantically equivalent with "I like dogs more than I like cats". These two sentences can be regarded as a pair of paraphrased sentences, and there are other ways to paraphrase these two sentences. The capability of a model which can express different sentences with the same meaning under different scenarios improves language variety and model robustness. Paraphrasing has been shown beneficial to a wider range of NLP problems, includes question answering (Buck et al., 2017) (Dong et al., 2017), summarization, semantic parsing (Su & Yan, 2017) and machine translation (Cho et al., 2014). Additionally, paraphrasing also demonstrates the diversity of human languages.

The modern paraphrasing method are modifications from machine translation (Mallinson et al., 2017) and have limited control on the quality and styles of paraphrases. However, in real world applications, people may demand paraphrases with various styles.

Conditioning has been widely used in other fields such as computer vision (Aliakbarian et al., 2020) and robotics (Santos et al., 2020). However, no published work on introducing conditioning to paraphrase tasks has been found, as far as we know. Thus, we believe our idea is innovative. To be specific, our goal is to propose an end-to-end neural paraphrasing model that generates paraphrases conditioned on input sentence and prior defined styles.

Our project is built on existing work on paraphrase generation. We used a pretrained FAIRSEQ models (Ott et al., 2019) to generate our paraphrasing dataset. Additionally, due to restricted control over quality of paraphrases sentences on a state-of-the-art model (Mallinson et al., 2017) and conditioning-relevant work in other fields, we propose an innovative method to introduce conditioning to paraphrase generation.

To summarise, our main contribution includes:

- Use a pretrained Neural Machine Translation model to construct a new paraphrasing dataset from WMT 2019 Europarl German-English dataset.
- Propose tagging and fine-tuning methods on length ratio for conditioning paraphrase generation on two dataset we generated.
- Evaluate the paraphrasing and conditioning performance on models conditioned by length ratio.

2. Dataset and task

2.1. Task definition

Our task can be defined as a method to build a paraphrase conditioning model for desired features. A traditional paraphrase model maximises probabilities of target sentences given source sentences. Mathematically, we want to find

$$\hat{t} = \max_{t} P(t|s) \tag{1}$$

where s, t are the source sentence and the target sentence respectively. Now, we want to introduce a conditioning term into the probability maximisation process. Mathematically, we now want

$$\hat{t} = \max_{t} P(t|s, c) \tag{2}$$

where c is a condition term. For example, if we want a longer paraphrase, then the condition term can be $\frac{Len(t)}{Len(s)} > 1$.

2.2. Datasets

There are many existing datasets for paraphrase tasks. Inspired by previous works (Prakash et al., 2016), we decided to extract two datasets with distinct traits. The first dataset is made from WMT 2019 Europarl German-English dataset (Bojar et al., 2019). We used a pre-trained machine translation model on the first 400k sentences in the dataset to construct syntactic paraphrase sentence pairs. Each pair contains an original English sentence and another English sentence translated from its corresponding German sentence. In particular, we concatenated the forward and reverse direction of the paraphrase pairs. That is to say, for each source to target sentence pair in the dataset, we also added target to source sentence pair in the dataset. The reason is because we suppose the model's performance on paraphrasing may be affected by only using manually-written sentences or only using translated sentences. By doing so, the size of our dataset is doubled which is a side advantage for our model training.

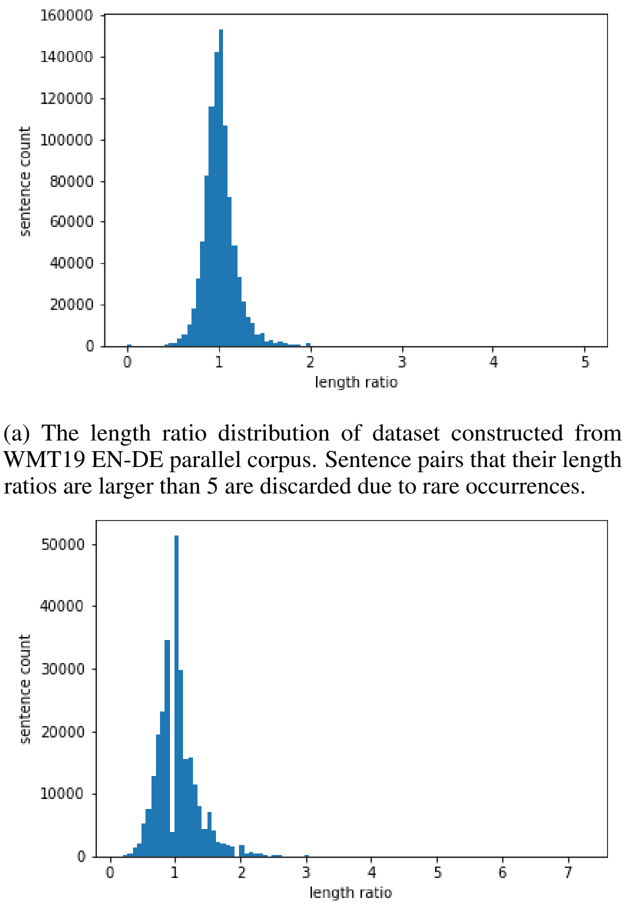
The second dataset we used is extracted from two datasets: MSCOCO (Lin et al., 2014) and QUORA[1], both of which are popular datasets for paraphrase generation task. MSCOCO is a dataset containing approximately 120,000 images and 600,000 captions. In particular, each image has 5 different captions annotated by different people. The annotators are asked to write a caption that can summarise each image. This requirement for annotators ensures that the captions are describing the same scene or object, so they can be regarded as paraphrased sentences. Inspired by the methods used in previous work (Prakash et al., 2016), We extract source-target pairs for paraphrase tasks by randomly selecting four caption sentences for each image as two paraphrase pairs for each image. QUORA is another dataset with about 400,000 probable question pairs with duplicated annotations. If the annotation for a pair is 1, it means that this question pair contains two duplicated questions. In order to get meaningful paraphrase pairs, we filtered out all the duplicated questions pairs and get a new sub-dataset for our model to train and test on. In our work, we combined these two datasets, and the combined dataset comprises 272550 paraphrase pairs to conduct experiments.

From Figure 1, we can see that the distribution for both paraphrase datasets are Laplace-like distributions. The fact that both the existing and generated datasets have Laplace-like distributions poses a challenge to our task because the length ratio distribution is right-skewed. Additionally, the length ratio distributions for both datasets are very sharp and narrow with a great number of samples aggregate around mean value. Thus it introduces ambiguity to categorise paraphrase sentences, which makes our task much more challenging.

2.3. Data preprocessing

The dataset generated from WMT corpora contains "bad" sentence pairs since the modern machine learning systems are robust in cycle consistency. The naive approach of pivoting of a single language and back-translation generates plenty of identical sentences compared with the source sentences. Therefore, we removed the sentence pairs with extreme values in BLEU scores or word edit distances, where extreme values are 0 and 1.

Then, we applied Byte Pair Encoding (Sennrich et al., 2016) model (35k operations for each corpus) and truecaser model from Moses toolkit (Koehn et al., 2007) on the dataset and split the dataset into training (755936 sentences), validation (95830 sentences) and test (95831 sentences) subsets. The processing of the second dataset which is the combination of MSCOCO and QUORA follows the same procedure, where the training, validation and test subsets have 218037 sentences, 27255 sentences and 27255 sentences respectively.



(a) The length ratio distribution of dataset constructed from WMT19 EN-DE parallel corpus. Sentence pairs that their length ratios are larger than 5 are discarded due to rare occurrences.

(b) MSCOCO and QUORA Length Ratio Distribution

Figure 1: Figures of length ratio distribution of different datasets

2.4. Evaluation methods

There are basically two levels of evaluation in our task. Firstly, it is needed to evaluate the paraphrasing quality of the model. This is a prior requirement for this task as we do not want sentences produced after conditioning do not seem like paraphrases any more. One can do this by using semantic textual similarity comparison. Semantic textual similarity (STS) rates the degree of semantic equivalence between two text snippets, which was also used by ParaNet (Mallinson et al., 2017). In this work, we used the BERT (Base)-based SentenceTransformer (Reimers & Gurevych, 2019) model, which was pretrained on the SNLI (Bowman et al., 2015) and fine-tuned on the STS

[1] https://www.kaggle.com/c/quora-question-pairs/data

benchmark (Cer et al., 2017), to compute the sentence embeddings between the source and target sentences. We then used the cosine similarity score between the source and target embeddings as the similarity score. The mean similarity score of the sentence pairs on the test set can be seen as a paraphrasing quality measurement.

The other level of evaluation is to see whether the model accomplishes the conditioning task itself. To evaluate this, we plotted out the feature distribution of paraphrases generated by the model. If the feature distribution shifts to the correct direction compared to the baseline, the model can be declared to have the right behaviour. When the distributions of the generated text are similar across different models, means of the features of the generated text can be directly compared for a simpler purpose. Another metric is to see the ratio of sentences matching the requirement in the test. For example, if the requirement is to generate longer paraphrases than source sentences, then a good metric can be the ratio of paraphrases in the test that that are longer than the source sentences.

3. Methodology

3.1. Corpus construction from NMT

Due to the limited availability of large-scale paraphrase corpus, we firstly generated a corpus of English paraphrases via pivoting from NMT. We generally followed the method proposed by Mallinson et al. (Mallinson et al., 2017). We first translated German sentences into English from a parallel corpus and use the translated English sentences as targets and the raw sentences in the parallel corpus as the sources. The method they proposed used multi-pivoting with a range of languages, while we used only German. This is because we have limited computational resources and it may take much time to finish the translation if multi-pivoting is used.

As for the NMT model, to save computation resources and time, we decided to use pretrained models. The one we selected was a pretrained FAIRSEQ model (Ott et al., 2019), which was trained on WMT2019 (Bojar et al., 2019) English-German datasets. The reason why we chose this is that this model is relatively new with Transformer architecture (Vaswani et al., 2017b), which shows a better performance and efficiency than the other available convolutional models. Furthermore, it was trained on a very large dataset with a wide range of domain information, which is therefore more robust and better for general-purpose research.

3.2. Training an end-to-end paraphrase system

The next step was to train an end-to-end paraphrase system. Since paraphrasing is similar to machine translation with the exception that the source and target sentences are in the same language but in different styles, we can simply use an NMT framework to train a paraphrase system. The framework we used was also FAIRSEQ (Ott et al., 2019). This framework is python-based, so it is easy to investigate the intermediate variables in the model or change parameter when performing experiments.

One natural question is to ask why to train a new end-to-end model instead of employing an existing method from the machine translation models in Section 3.1. The reason is the NMT method s several separate models to generate a paraphrase (at least 2 with one pivot). For example, we need a DE-EN model and an EN-DE model to generate a paraphrase in German or English. However, to conducting the conditional paraphrasing, we will need to perform fine-tuning or tagging, described below, which is only feasible within one single model.

3.3. Conditions for paraphrase controlling

To control the styles of paraphrasing, we focused on one feature only – length ration of sentence pairs. Length ratio is mathematically defined as

$$f(src, tgt) = \frac{Len(src)}{Len(tgt)} \tag{3}$$

We chose this feature as firstly, shorter or longer paraphrases are indeed different styles of paraphrases, which would need to model to perform either summarising or expansion. Also, it is straightforward to see and compare from raw sentences. This is important as in reality, people tend to ask for more concrete features, such as longer paraphrases than abstract ones, such as syntactically different features. Therefore, it makes this system more useful.

We also considered other features, which can be seen from Appendix A.

3.4. Tagging for conditioning

One way to perform the conditioning is to use tagged sequences. Tagging a sequence by introducing a new token for the target language has proved successful in multilingual machine translation (Johnson et al., 2017). We employed the same idea to tag the source sentence with a new token of the feature score of the intended pair. That is to say, the format of input of the model will be:

`<intended feature score> original source sentence`.

Since the feature now appears as the first token of the sentence, the model should learn the dependency relationship between this feature and other tokens and as a result, the model will generate paraphrases based on feature scores.

The implementation method is simple, we added feature scores of each pairs of sentences to the beginning of the source sentence and retrained a new model on the tagged data. One remaining issue is the tag token. Since a token can only be discrete values instead of continuous ones, one will need to discretise these values. We divided the training data into several parts, each part with approximately equal size to ensure the model would not be biased to any part of the data. Then, we assigned a new tag to the each sentence pair. For example, if we divided paraphrases as shorter, same and longer groups and each with similar number of samples, we then assign 0 to the "shorter" group, 1 to the "same" group and 2 to the "longer" group.

There are two advantages of this method. Firstly, there is no need to train multiple models for different feature scores. Instead, one can use one model to generate different styles of paraphrases. Secondly, it allows finer granularity of feature scores. One may use an arbitrary feature score for the tag, not just "short" or "long", although we did not use arbitrary tags in our experiments.

The risk of this approach is that unlike machine translation where different tags would lead to very different outputs, impact of variance of tags on the outputs is much smaller in paraphrasing systems. Therefore, the system may not learn the dependency very well. Also, sentences with different tags do not vary too much, which may make the semantic of the tag more ambiguous for the model.

3.5. Fine-tuning for conditioning

This is our backup approach to accomplish the goal. To exploit the recency bias in training neural networks, we simply used part of our training data to further train the model. For example, one could pick pairs of sentences with longer targets than sources and continue to train the baseline model with this part of data. Due to recency bias, the fine-tuned model will tend to produce longer paraphrases. The limit of this method is obvious: one will need different models for different features.

4. Experiments

4.1. Model architecture and training configuration

We conducted experiments to make a sequence to sequence paraphrasing model be responsive to a input signal and maintain the quality of generated paraphrases at the same time, instead of improving the performance of paraphrasing models. In this project, we trained a vanilla transformer as the baseline paraphrasing model and compared the performance of the models of two different approaches with baseline: training with tagged data or fine-tuning the pre-trained baseline model.

For all experiments, we used a vanilla transformer architecture from the sequence modelling toolkit - FAIRSEQ (Ott et al., 2019), which consists 6 encoder layers, 6 decoder layers and 8 attention heads in each layer. We shared the word embeddings since the source language and target language are the same and this can reduce the number of parameters of the model. The inverse root square scheduler was used with initial learning rate of 1e-4. The warm-up phase was 4000 updates with initial learning rate being 1e-7. The optimiser algorithm was Adam (Kingma & Ba, 2017) and dropout rate is 0.3.

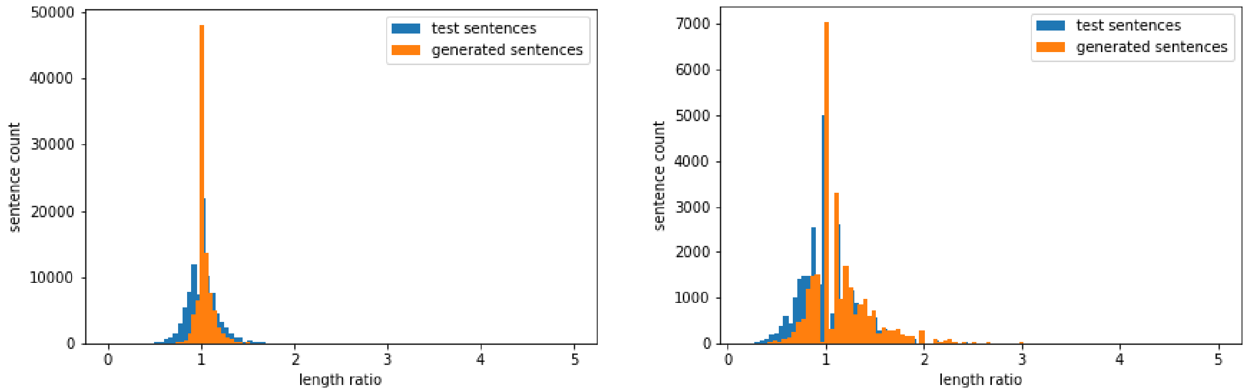
4.2. Baseline models

At this section, we primarily focused on the distribution change between test set and generated sentences on both datasets. We initially trained the baseline models with a vanilla Transformer architecture without tagging on two datasets. To make the results of two datasets comparable, we omitted any sentences pair whose length ratio is larger than 5, since they rarely appear but have strong effect on the calculation of mean length ratios.

In Table 1, the average length ratios of generated paraphrases from both baseline models show the similar trend that the average length ratio of generated paraphrases is slightly larger than that of original paraphrases in the test set. The same observation is also made on the distribution of length ratios from two models (Figure 2). Most of generated sentences have similar length with corresponding source sentences. The distributions of generated paraphrases from both baselines shifted towards the left and also squeezed to the mode. Therefore, the vanilla Transformer may prefer to generate sentences with shorter or similar length.

The phenomenon that the Transformer has a tendency to generate sentences with identical or shorter length of the source sentences, is because one-to-one or many-to-one mapping between tokens of source and target sentences is easy to learn for a sequence to sequence model. Oppositely, one-to-many mapping is much harder. Another possible reason is current training techniques can implicitly provide information about sentence length. A well-known example is teacher-forcing. Teacher-forcing makes training much faster by feeding the target word into decoder and paralleling the process of each token. However, it forces the decoder to stop whenever the true <eos> token comes up. Moreover, most of the sentence pairs are distributed close to 1, as shown in the Figure 1. Hence, it is reasonable that the model learns the length differences of most of the sentences but cannot generalise to edge cases in the datasets.

We then calculated the semantic textual similarity (STS) scores of baselines on the two datasets. On the raw datasets, we simply encoded the source and target sentences, calculated the cosine similarity scores on each pair and averaged them. On the trained baseline model, we still used the raw source sentences but with the model generated target sentences as the other side to calculate the cosine similarity. Results are also shown in Table 2. As can be seen from the table, in both datasets, the STS similarity scores of the trained baseline model are higher than those of the raw datasets. Since we assumed that the raw datasets are genuine paraphrases, the baseline model can be declared to be able to retain the semantics of the raw sentences. However, as stated above, the baseline model has a stronger tendency to copy the source sentence and source and target sentences are too similar in the WMT19 dataset, so the STS similarity score is very high for the WMT19 dataset. This, however, cannot be seen as very good paraphrases as people do not want semantically fully copied sentences



(a) WMT19 EN-DE

(b) MSCOCO+QUORA

Figure 2: The distribution comparison of two baseline models trained on two datasets. Sentences pairs that their length ratios are larger than 5 are omitted.

Data set	Test	Baseline	tag longer	tag neutral	tag shorter	Fine-tune
WMT19 EN-DE	1.016	1.038	-	-	-	-
MSCOCO+QUORA	1.045	1.161	1.121	1.124	1.130	0.839

Table 1: The average length ratios on test set by different methods. Test mean is the raw test data. Baseline mean is generated sentence from baseline model with source sentences. Tagged mean is from model trained with tagged data. Fine-tune mean is from baseline fine-tuned by appropriate data. The sentences pairs with length ratio greater than 5 are discarded due to rare occurrence.

Data set	STS similarity mean
WMT19 EN-DE (raw)	0.885
WMT19 EN-DE (seq2seq baseline)	0.967
MSCOCO+QUORA (raw)	0.731
MSCOCO+QUORA (seq2seq baseline)	0.787

Table 2: Table of the STS similarity scores of the raw data and the baseline seq2seq model on the test set of the two datasets.

test set	longer	neutral	shorter
raw	9075	8531	9649
mixed tag	5264	10643	11348
tag shorter	5137	10644	11474
tag neutral	5214	10781	11260
tag longer	5385	10704	11166

Table 3: The count of sentence length changes on different test sets. Raw is tag distribution on original test set. Mixed tag is the tag distribution of generated paraphrases from raw tagged test set. Tag `shorter`, `neutral` and `longer` are test set tagged only by that tag respectively.

from a paraphrasing systems. The problem is relieved in MSCOCO+QUORA dataset as sentences in this dataset are more variable.

4.3. Results

In this section, we applied two approaches to check if the Transformer has the ability to learn the input signal about desired sentence length. One approach is tagging the source sentences with discretised length-ratio tag. Another approach is fine-tuning the baseline model with a set of paraphrases of target length ratio. We especially concentrated on testing the performance of the model with `longer` tags or model which is fine-tuned by longer sentences, since it is a harder task for model and the results will be more obvious.

For the tagging experiments, we assign its length ratio to every paraphrase pair in both datasets. Since length ratio is continuous number and tagging data with continuous number causes infinite size of dictionary, the length ratio has to be discretised into three categories for simplicity: `shorter`, `neutral` and `longer`. The distributions of length ratio of both datasets (Figure 1b) are close to Laplace distribution with the mean of 1. So, we take roughly the 33.3 percentile (length ratio=0.89) and 66.6 percentile (length ratio=1.11) of length ratios as the boundaries of categories. Only source sentences are tagged as both encoder and decoder depend on the tag token.

$$P(tgt, src|tag) = P(tgt|src, tag)P(src|tag)$$

Data set	STS similarity mean
MSCOCO+QUORA (tagged)	0.783
MSCOCO+QUORA (fine-tune)	0.564

Table 4: Table of the STS similarity scores of the raw data and the baseline seq2seq model on the test set of the two datasets.

Since the first dataset (constructed from WMT19) is not as good as the second one (the combination of human annotated sentences) from syntactic diversity and semantic similarity perspectives, we only use the second dataset for the following experiments.

Even though the distribution of the second dataset does not show a significant shift to the left (Figure 3), the tagging method still has marginal improvement on the average length ratio. By comparing the average length ratio of the test data with three different tags (Table 1), we observed that the average length ratio increases from 1.121 to 1.130 as the tag moves from `longer` to `neutral` then to `shorter`. Table 3 also conforms to the above observation. The counts of sentence changes move to desired direction. For instance, generated sentences from test set with tag `longer` has more long sentences than sentences from test set with mixed tags. It is reasonable since the calculation of length ratio is the length of source sentence over the length of target sentence. The tagging approach also maintains the same semantic similarity score as the baseline. (Table 2 & 4).

However, the average length ratios of generated paraphrases with all three tags are higher than the raw test paraphrases and lower than the baseline results. So, the nature of the Transformer model that it prefers to generate slightly shorter or identical sentences in length does not affected by the tag. We believe the reason that tagging source sentences would generate longer sentence is that tag adds one extra token to the source sentences and the variance of length ratio of the dataset is small. From the model perspective, the every sentence in the dataset becomes longer by adding a tag token.

We fine-tuned the baseline with paraphrases which target sentences are longer than source sentences (length ratio is smaller than 0.8). The fine-tune dataset contains 47502 sentences and 5981 sentences in the training and validation sets respectively. As shown in the Figure 4, there is a significant left shift on the distribution of length ratio. The result in the Table 1 also conforms this observation, which the average length ratio is smaller than the raw test value by almost 0.2. However, the paraphrase quality is poor. The mean STS similarity of fine-tuned model is the worst along with other models by 0.2 scores.

4.4. Discussion

It can be seen from the conditioning performance that the tagging approach did not work as well as the fine-tuning approach, while the tagging approach produces better paraphrases. Given that in the tagging approach, the only thing that affected the model was the tag tokens added to the source sentence, we hypothesis that the model did not actually learn the dependencies on the tag tokens well. To further investigate how well the model dealt with dependencies on the tag token, we plotted out the cross attention (Encoder-Decoder attention) matrices on the decoder side of the model. In specific, we sampled three sentences from the test set of the MSCOCO+QUORA dataset and ran the model to get the attention matrices at all decoder layers. Then, we averaged these attention matrices to get an overall view of the attention scores. Visualisation of the cross attention matrices can be seen in Figure 5.

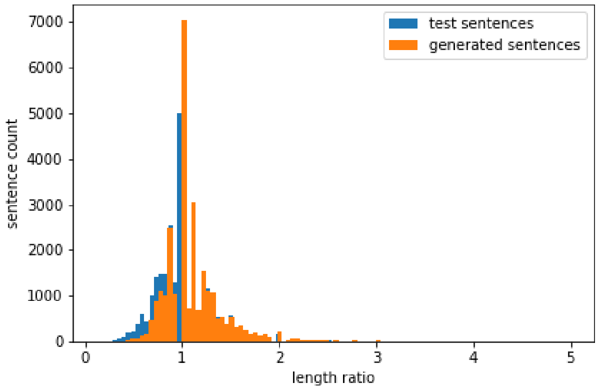


Figure 3: The distribution comparison between raw test data and generated data of MSCOCO+QUORA dataset with tag `longer`. Sentences pairs that their length ratios are larger than 5 are omitted.

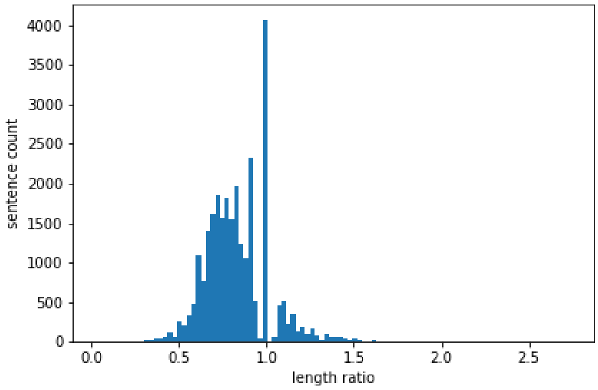


Figure 4: The distribution comparison between raw test data and generated data after fine-tuning with longer paraphrases. Sentences pairs that their length ratios are larger than 5 are omitted.

As can be seen from the figure, the model attends the tag mostly with the first words in the generated text, instead of uniformly distributed with words. This might be because the tag is added to the start of the sentence, so that the words surrounding it are affected the most. Therefore, to further improve the model, one can try to add the tag both at the

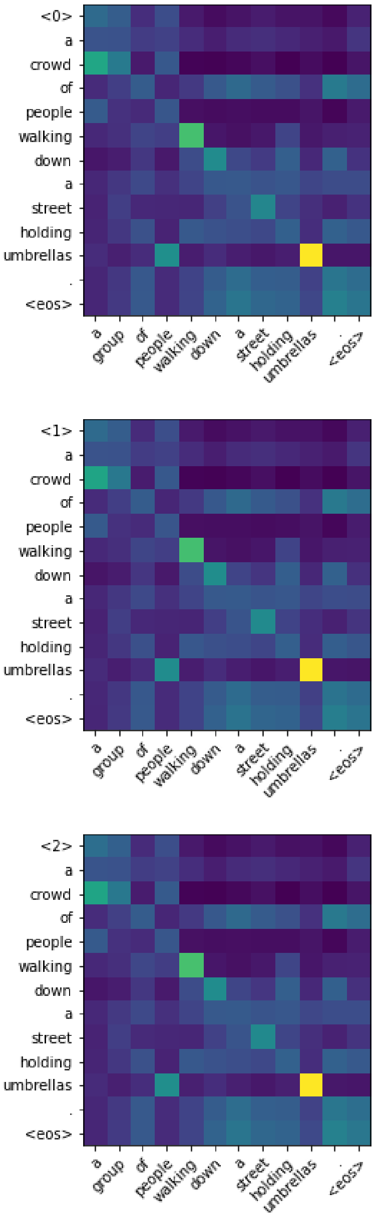


Figure 5: Cross attention matrices of three samples in the MSCOCO+QUORA dataset. Note the deep colours imply smaller attention values and light colours imply lager values.

start and end of the source sentences. Also, the distributions of attentions on the tag are more uniform than other words. One may also try to discourage such uniform distributions to get a more sparse and useful attention scores.

Regarding fine-tuning, although this approach achieved better conditioning performance, it did show worse paraphrasing performance. The reason we hypothesis is that only a small subset of the training data was involved in fine-tuning. Therefore, recency bias reinforces not only the style of the model, but also the bias of the model on this small subset. The model is thus more likely to lose the ability to fit the general domain so as to produce a bad results.

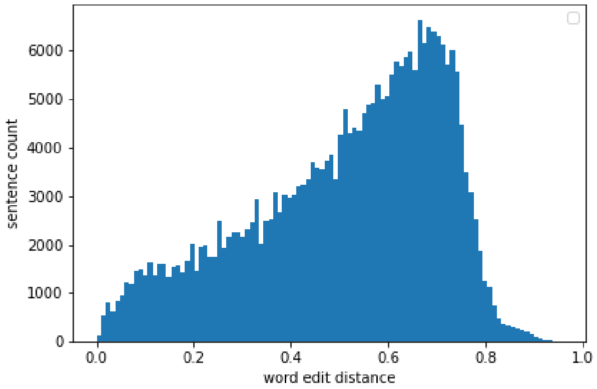


Figure 6: The distribution of word edit distribution of MSCOCO+QUORA, which is normalised by the length of the longest sentence.

One other interesting finding is that the STS similarity became larger when a model was fitted to the raw data on both datasets. This means that after training a seq2seq model, paraphrasing performance can be lifted in some way, although some more experiments may be needed to support this argument. We hypothesis that this is because the seq2seq model will tend to copy the information from the source while still match the target style. Therefore, it may produce a better paraphrasing performance. However, there is a risk when the training source and target sentences are too similar for the model to catch the style change. In that case, the model will have a strong tendency to become a language model, i.e. the model will tend to generate the exact same sentences as source sentences, just as the our model on the corpus constructed from WMT19 data.

4.5. Future extension

The first extension is to follow the findings of the investigation of the tagging approach. Specifically, one can try to add the tags at both the start and end of source sentences. Also, one can try to discourage uniform distributions on the attentions of the tags by introducing a penalty term in the loss function.

Secondly, due to the time limit, we only explored the length ratio. However, it may not be the best feature for conditioning, especially for the tagging approach. As can be see from the Figure 6, word edit distance is more uniformly distributed than length ratio. This might make the amounts of data for each tag more even. Also, the difference between sentences with different tags is more salient.

Finally, one other extension is that one can try to modify an NMT model to perform conditioning, given that paraphrasing from NMT is the state-of-the-art method. Our experiments also showed that NMT can produce much better paraphrasing than trained an end-to-end model. Therefore, introducing a bridging variable between the two NMT models and conditioning on that variable may be a beneficial trial. One can also try reinforcement learning (Hu et al.,

2020) for conditioning. Reinforcement learning can treat different conditions as rewards and generating actions as policies. It may also achieve good results, though it may cause more inefficiency and be more unlikely to converge.

5. Related work

Early research on automatic generation of paraphrases focus on rule-based (McKeown, 1980) (Meteer & Shaked, 1988) and data-driven methods (Madnani & Dorr, 2010). In particular, there are many researches on data-driven method with various training data such as alignments of comparable corpora (e.g. news describing the same events (Dolan et al., 2004) (Barzilay & Lee, 2003)) or different machine translations on the same sentences (Barzilay & McKeown, 2001) (Pang et al., 2003).

Later, machine translation becomes the most popular methodology for paraphrase generation. In 2005, (Bannard & Callison-Burch, 2005b) find English and non-English words mapping from a Machine Translation phrase table. Then (Kok & Brockett, 2010) used random walks to get paraphrased sentences from many phrase tables. With the popularity of neural network surges, the performance of generating paraphrasing sentences from translation models has been vastly improved (Bahdanau et al., 2014) (Vaswani et al., 2017a). Neural Machine Translation (Wieting & Gimpel, 2017) (Iyyer et al., 2018) are one of the best machine translation models for paraphrase generation.

Recent studies on paraphrase generation mainly focus on machine translation models with pivoting (Mallinson et al., 2017) (Yu et al., 2018). Pivoting is often used to overcome the shortage of parallel data in machine translation and takes advantage of translation paths through an intermediate language. Since there is no translation path from English to English, a path from English to a foreign language to English can be used. The foreign language is our pivoting language in this case. The creation of Paraphrase Database (PPDB) (Ganitkevitch et al., 2013), a large-scale syntactic paraphrase dataset, uses a syntax-based Statistical Machine Translation (SMT) model with bilingual pivoting method (Bannard & Callison-Burch, 2005a). Two English strings e_1 and e_2 that translate to the same foreign string f can be assumed to have same meaning. The model then pivots over f to extract a pair of paraphrase $<e_1, e_2>$.

Paraphrasing in the context of NMT is also revisited (Mallinson et al., 2017). End-to-end NMT models only require parallel data for training and minimal linguistic information. Neural paraphrasing model learns continuous space representation for phrases and sentences that can be useful for downstream tasks such as text similarity measurement and entailment. However, one-to-one back-translation assumes that pivot language must fully capture the exact meaning of a English sentence. A information loss during translation is inevitable since there is rarely a clear one-to-one mapping between sentences in different languages. The neural paraphrasing model pivots through the sets of K-best sentences $\mathcal{F} = \{F_1, ...F_k\}$ of E_1 from multiple languages, respectively. The averaging probability over multiple sentences ensures that multiple aspects of the source sentence are captured and provides resilience against a single bad translation. For instance, two languages are used as pivots:

$$P(E_2|E_1) = \prod_{t'=1}^{T_{E_2}} P(y_{t'}|y_{<t'}, \mathcal{F}^{F^1}, \mathcal{F}^{F^2})$$

6. Conclusions

In conclusion, we have found the following outcomes. The dataset we generated using NMT model from WMT 2019 Europarl German-English dataset is not suitable for end-to-end paraphrase tasks, and we believe it is due to the fact that its distribution is too centralised. For the two methods we proposed for conditioning, we found that tagging had limited effect on conditioning, while it improved the performance of paraphrasing task. In contrast, fine-tuning method had a noteworthy effect on conditioning, and as a trade-off, the quality for paraphrasing sentences dropped. There are still a lot that can be explored on conditioning paraphrase generation. In the future, other than length ratio, features like average BLEU score and word edit distance can also be used for conditioning. Other ways of tagging such as changing the position of the tag and hinder uniform distribution on attentions can be experimented. Modifications on state-of-the-art NMT model for conditioning tasks are also awaiting to be explored.

References

Aliakbarian, Sadegh, Saleh, Fatemeh Sadat, Salzmann, Mathieu, Petersson, Lars, and Gould, Stephen. A stochastic conditioning scheme for diverse human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5223–5232, 2020.

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Bannard, Colin and Callison-Burch, Chris. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 597–604, Ann Arbor, Michigan, June 2005a. Association for Computational Linguistics. doi: 10.3115/1219840.1219914. URL https://www.aclweb.org/anthology/P05-1074.

Bannard, Colin and Callison-Burch, Chris. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 597–604, 2005b.

Barzilay, Regina and Lee, Lillian. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. *arXiv preprint cs/0304006*, 2003.

Barzilay, Regina and McKeown, Kathleen. Extracting paraphrases from a parallel corpus. In *Proceedings of the*

39th annual meeting of the Association for Computational Linguistics, pp. 50–57, 2001.

Bojar, Ondřej, Chatterjee, Rajen, Federmann, Christian, Fishel, Mark, Graham, Yvette, Haddow, Barry, Huck, Matthias, Yepes, Antonio Jimeno, Koehn, Philipp, Martins, André, Monz, Christof, Negri, Matteo, Névéol, Aurélie, Neves, Mariana, Post, Matt, Turchi, Marco, and Verspoor, Karin (eds.). *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, Florence, Italy, August 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W19-5200.

Bowman, Samuel R., Angeli, Gabor, Potts, Christopher, and Manning, Christopher D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

Buck, Christian, Bulian, Jannis, Ciaramita, Massimiliano, Gajewski, Wojciech, Gesmundo, Andrea, Houlsby, Neil, and Wang, Wei. Ask the right questions: Active question reformulation with reinforcement learning. *arXiv preprint arXiv:1705.07830*, 2017.

Cer, Daniel, Diab, Mona, Agirre, Eneko, Lopez-Gazpio, Iñigo, and Specia, Lucia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://www.aclweb.org/anthology/S17-2001.

Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Dolan, William, Quirk, Chris, Brockett, Chris, and Dolan, Bill. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. 2004.

Dong, Li, Mallinson, Jonathan, Reddy, Siva, and Lapata, Mirella. Learning to paraphrase for question answering. *arXiv preprint arXiv:1708.06022*, 2017.

Ganitkevitch, Juri, Van Durme, Benjamin, and Callison-Burch, Chris. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1092.

Hu, J., Niu, H., Carrasco, J., Lennox, B., and Arvin, F. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(12): 14413–14423, 2020. doi: 10.1109/TVT.2020.3034800.

Iyyer, Mohit, Wieting, John, Gimpel, Kevin, and Zettlemoyer, Luke. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

Johnson, Melvin, Schuster, Mike, Le, Quoc V., Krikun, Maxim, Wu, Yonghui, Chen, Zhifeng, Thorat, Nikhil, Viégas, Fernanda, Wattenberg, Martin, Corrado, Greg, Hughes, Macduff, and Dean, Jeffrey. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://doi.org/10.1162/tacl_a_00065.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization, 2017.

Koehn, Philipp, Hoang, Hieu, Birch, Alexandra, Callison-Burch, Chris, Federico, Marcello, Bertoldi, Nicola, Cowan, Brooke, Shen, Wade, Moran, Christine, Zens, Richard, Dyer, Chris, Bojar, Ondřej, Constantin, Alexandra, and Herbst, Evan. 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*, pp. 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P07-2045.

Kok, Stanley and Brockett, Chris. Hitting the right paraphrases in good time. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 145–153, 2010.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Madnani, Nitin and Dorr, Bonnie J. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387, 2010. doi: 10.1162/coli_a_00002. URL https://www.aclweb.org/anthology/J10-3003.

Mallinson, Jonathan, Sennrich, Rico, and Lapata, Mirella. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 881–893, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1083.

McKeown, Kathleen R. Paraphrasing using given and new information in a question-answer system. *Technical Reports (CIS)*, pp. 723, 1980.

Meteer, Marie and Shaked, Varda. Strategies for effective paraphrasing. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*, 1988.

Ott, Myle, Edunov, Sergey, Baevski, Alexei, Fan, Angela, Gross, Sam, Ng, Nathan, Grangier, David, and Auli, Michael. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

Pang, Bo, Knight, Kevin, and Marcu, Daniel. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 181–188, 2003.

Prakash, Aaditya, Hasan, Sadid A, Lee, Kathy, Datla, Vivek, Qadir, Ashequl, Liu, Joey, and Farri, Oladimeji. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*, 2016.

Reimers, Nils and Gurevych, Iryna. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Santos, João, Oliveira, Miguel, Arrais, Rafael, and Veiga, Germano. Autonomous scene exploration for robotics: A conditional random view-sampling and evaluation using a voxel-sorting mechanism for efficient ray casting. *Sensors*, 20(15):4331, 2020.

Sennrich, Rico, Haddow, Barry, and Birch, Alexandra. Neural machine translation of rare words with subword units, 2016.

Su, Yu and Yan, Xifeng. Cross-domain semantic parsing via paraphrasing. *arXiv preprint arXiv:1704.05974*, 2017.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Lukasz, and Polosukhin, Illia. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017a.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, undefinedukasz, and Polosukhin, Illia. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017b. Curran Associates Inc. ISBN 9781510860964.

Wieting, John and Gimpel, Kevin. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*, 2017.

Yu, Adams Wei, Dohan, David, Luong, Minh-Thang, Zhao, Rui, Chen, Kai, Norouzi, Mohammad, and Le, Quoc V. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

A. Other features we have considered to explore

We also considered the following features but did not implement them due to various reasons.

- Feature 1: The cosine similarity between the sentence embeddings of two sentences. Mathematically,

$$f_1 = \mathrm{CosSim}(Embed(s_1), Embed(s_2)) \tag{4}$$

  where s_1, s_2 are paraphrases. This is sensible in that sentences with more similar semantics usually have similar embeddings. However, sentence embeddings are less explainable and thus being less useful.

- Feature 2: Word edit distance between two paraphrases. Mathematically,

$$f_2 = \mathrm{EditDistance}(s_1, s_2) \tag{5}$$

  We hypothesise that paraphrases with some simple word/syntactic changes should have small edit distance while those with more complicated changes should differ much more. This is actually a very sensible feature. It also has a good distribution on our data and we will investigate into this feature in the future.

- Feature 3: Average BLEU score. Mathematically,

$$f_4 = (BLEU(s_1, s_2) + BLEU(s_2, s_1))/2 \tag{6}$$

  BLEU is a good measure for both the fluency difference and word similarity between the hypothesis and reference. Thus, one can use BLEU as a measure of complexity of paraphrasing. One issue is that BLEU is generally not a symmetric metric, i.e. $BLEU(a, b) \neq BLEU(b, a)$. Therefore, we average the BLEU scores of two directions to make it symmetric for paraphrases. This is also a sensible feature but again, we did not have enough time on this and will leave this in the future.