Negative Lexically Constrained Decoding for Paraphrase Generation

Tomoyuki Kajiwara
Institute for Datability Science
Osaka University, Osaka, Japan
kajiwara@ids.osaka-u.ac.jp

Abstract

Paraphrase generation can be regarded as monolingual translation. Unlike bilingual machine translation, paraphrase generation rewrites only a limited portion of an input sentence. Hence, previous methods based on machine translation often perform conservatively to fail to make necessary rewrites. To solve this problem, we propose a neural model for paraphrase generation that first identifies words in the source sentence that should be paraphrased. Then, these words are paraphrased by the negative lexically constrained decoding that avoids outputting these words as they are. Experiments on text simplification and formality transfer show that our model improves the quality of paraphrasing by making necessary rewrites to an input sentence.

1 Introduction

Paraphrase generation is a generic term for tasks that generate sentences semantically equivalent to input sentences. These techniques make it possible to control information other than the meaning of the text. Typical paraphrase generation tasks include subtasks such as text simplification to control complexity, formality transfer to control formality, grammatical error correction to control fluency, and sentence compression to control sentence length. These paraphrase generation applications not only support communication and language learning but also contribute to the performance improvement of other natural language processing applications (Evans, 2011; Štajner and Popović, 2016).

Paraphrase generation can be considered as a monolingual machine translation problem. Sentential paraphrases with different complexities (Coster and Kauchak, 2011; Xu et al., 2015) and formalities (Rao and Tetreault, 2018) were created manually, and parallel corpora specialized for each subtask were constructed. As in the field of machine translation, phrase-based (Coster and Kauchak, 2011; Xu et al., 2012) and syntax-based (Zhu et al., 2010; Xu et al., 2016) methods were proposed early. In recent years, the encode-decoder model based on the attention mechanism (Nisioi et al., 2017; Zhang and Lapata, 2017; Jhamtani et al., 2017; Niu et al., 2018) has been studied, inspired by the success of neural machine translation (Bahdanau et al., 2015).

In machine translation, all words appearing in an input sentence must be rewritten in the target language. However, paraphrase generation does not require rewriting of all words. When some criteria are provided, words not satisfying the criteria in the input sentence are identified and rewritten. For example, the criterion for text simplification is the textual complexity, and rewrites complex words to simpler synonymous words. Owing to the characteristics of the task where only a limited portion of an input sentence needs to be rewritten, previous methods based on machine translation often perform conservatively and fail to produce necessary rewrites (Zhang and Lapata, 2017; Niu et al., 2018). To solve the problem of conservative paraphrasing that copies many parts of the input sentence, we propose a neural model for paraphrase generation that first identifies words in the source sentence requiring paraphrasing. Subsequently, these words are paraphrased by the negative lexically constrained decoding that avoids outputting them as they are.

We evaluate the performance of the proposed method with two major paraphrase generation tasks. Experiments on text simplification (Xu et al., 2015) and formality transfer (Rao and Tetreault, 2018) show that our model improves the quality of paraphrasing by performing necessary rewrites to an input sentence.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6047–6052
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

2 Proposed Method

To improve the conservative rewriting of the neural paraphrase generation, we first identify the words to be paraphrased for a given input sentence (Section 2.1). Next, we paraphrase the input sentence using the pretrained paraphrase generation model. Here, we select sentences not including those words by adding negative lexically constrained decoding to the beam search (Section 2.2). Because our method only changes the beam search, it can be applied to various paraphrase generation models and model retraining is not necessary.

2.1 Identification of Word to be Paraphrased

We extract words strongly related to the source style included in the input sentence s_i as vocabulary V_i to be paraphrased. Point-wise mutual information is used to estimate the relatedness between each word $w \in s_i$ and style $z \in \{x, y\}$ (Pavlick and Nenkova, 2015). Here, x and y are the source style (*e.g.* informal) and the target style (*e.g.* formal), respectively.

$$\mathrm{PMI}(w, z) = \log \frac{p(w, z)}{p(w)p(z)} = \log \frac{p(w|z)}{p(w)} \quad (1)$$

We define the vocabulary V_i to be paraphrased using the threshold θ as follows.

$$V_i = \{w \mid w \in s_i \wedge \mathrm{PMI}(w, x) \geq \theta\} \quad (2)$$

After extracting the vocabulary V_i to be paraphrased for each input sentence s_i, we generate paraphrase sentences using it as a hard constraints. Note that PMI score is calculated using a training parallel corpus for paraphrase generation.

2.2 Negative Lexically Constrained Decoding

Lexically constrained decoding (Anderson et al., 2017; Hokamp and Liu, 2017; Post and Vilar, 2018) adds constraints to the beam search to force the output text to include certain words. The effectiveness of these methods are demonstrated in image captioning using given image tags (Anderson et al., 2017) and in the post-editing of machine translation (Hokamp and Liu, 2017).

In paraphrase generation, there is no situation that words to be included in the output sentence are given. Therefore, positive lexical constraints used in the image captioning and post-editing of machine translation cannot be applied to this task as they are. Meanwhile, negative lexical constraints that are forced to not include certain words in output sentence are promising for paraphrase generation. This is because, for example, text simplification is a task of generating sentential paraphrase without using complex words that appear in the source sentence.

	Train	Dev	Test
Newsela	94,208	1,129	1,077
GYAFC-E&M	52,595	2,877	1,416
GYAFC-F&R	51,967	2,788	1,332

Table 1: Number of sentence pairs for each dataset.

In this study, we add negative lexical constraints to beam search using dynamic beam allocation (Post and Vilar, 2018), which is the fastest lexically constrained decoding algorithm. In negative lexical constraints, we exclude hypotheses including the given words during beam search. Consequently, the words identified in Section 2.1 will not appear in our generated sentences.

3 Experiment

We evaluate the performance of the proposed method on two major paraphrase generation tasks. We conduct experiments on text simplification and formality transfer using datasets shown in Table 1. For text simplification, we identify complex words in the input sentence and generate simple paraphrase sentence without using these complex words. Similarly, for formality transfer, we identify informal words in the input sentence and generate formal paraphrase sentence without using these informal words.

3.1 Setup

For text simplification, we used the Newsela dataset (Xu et al., 2015) split and tokenized with the same settings as the previous study (Zhang and Lapata, 2017). For formality transfer, we used the GYAFC dataset (Rao and Tetreault, 2018) normalized and tokenized using Moses toolkit.[1] For each task, we used byte-pair encoding[2] (Sennrich et al., 2016) to limit the number of token types to $16,000$. In the GYAFC dataset, it is reported that a correlation exists between manual evaluation

[1] https://github.com/moses-smt/mosesdecoder

[2] https://github.com/rsennrich/subword-nmt

	Newsela					GYAFC-E&M				GYAFC-F&R			
	Add	Keep	Del	BLEU	SARI	Add	Keep	Del	BLEU	Add	Keep	Del	BLEU
RNN-Base	1.8	60.8	22.3	24.1	17.4	31.9	90.0	57.5	71.2	32.9	90.5	61.1	74.7
RNN-PMI	**2.8**	**61.1**	**36.5**	**24.7**	**22.8**	**33.5**	90.0	**59.9**	**71.7**	**34.3**	**90.9**	**63.1**	**75.9**
RNN-Oracle	10.4	82.9	89.9	36.4	40.0	34.8	92.7	72.4	75.2	35.7	93.2	74.6	79.3
SAN-Base	1.8	60.9	23.8	24.0	17.8	34.4	90.0	59.9	71.8	34.5	91.1	63.2	76.7
SAN-PMI	**2.5**	**61.3**	**38.0**	**24.6**	**23.3**	**35.2**	90.0	**61.2**	**72.1**	**35.3**	91.1	**64.0**	**77.0**
SAN-Oracle	10.1	82.0	89.4	35.9	39.9	36.6	92.4	71.4	75.1	36.6	92.9	73.7	79.8

Table 2: Performance of our paraphrase generation models on text simplification (complex → simple) in Newsela dataset and formality transfer (informal → formal) in GYAFC dataset. For both RNN and SAN models, our method consistently improves BLEU and SARI scores across styles or domains. In addition, a consistent improvement on Add and Del means that our method promotes active rewriting.

and automatic evaluation using BLEU only when paraphrasing from an informal style to formal style (Rao and Tetreault, 2018). Therefore, we will only experiment with this setting.

For lexical constraints, we identified words with a PMI score above the threshold θ. We selected a threshold $\theta \in \{0.0, 0.1, 0.2, ..., 0.7\}$ that maximizes the BLEU score between the output sentence and the reference sentence in the development dataset. We calculated PMI scores using each training dataset shown in Table 1.

As a paraphrase generation model, we constructed the recurrent neural network (RNN) and self-attention network (SAN) models using the Sockeye toolkit (Hieber et al., 2017).[3] Our RNN model uses a single LSTM with a layer size of 512 for both the encoder and decoder, and MLP attention with a layer size of 512. Our SAN model uses a six-layer transformer with a model size of 512 and a single attention head. We used word embeddings in 512 dimensions tying the source, target, and the output layer's weight matrix. We added dropout to the embeddings and hidden layers with probability 0.2. In addition, we used layer-normalization and label-smoothing for regularization. We trained using the Adam optimizer (Kingma and Ba, 2014) with a batch size of 4,096 tokens and checkpoint the model every 1,000 updates. The training stopped after five checkpoints without improvement in validation perplexity.

BLEU (Papineni et al., 2002) is primarily used for our evaluation metrics; SARI (Xu et al., 2016) is also used for text simplification. For a more detailed comparison of the models, we evaluated the F1 score of the words that are added (Add), kept (Keep), and deleted (Del) by the models.[4]

Our proposed method is compared with previous methods trained only on the dataset shown in Table 1. For detailed analysis, we chose the methods whose model outputs are published. Among these, Dress-LS (Zhang and Lapata, 2017) and BiFT-Ens (Niu et al., 2018) with the highest BLEU score in each task are compared with our model. Following BiFT-Ens, we also used a bidirectional domain-mixed ensemble model for formality transfer task.

We also experimented with Oracle settings that can properly identify words to be paraphrased. In this setting, we used all words that did not appear in the reference sentence among the words included in the input sentence as lexical constraints.

3.2 Results

The experimental results are shown in Table 2. These results in both RNN and SAN architectures and three datasets showed that our PMI-based method consistently improves the Base method that does not use constraints in both BLEU and SARI metrics. As a result of a detailed analysis of the model outputs, our PMI method always improves the Base method in terms of Add and Del in both model architectures. These results mean that our proposed method promotes active rewriting as expected. In addition, since Oracle method shows higher performance, it is worthwhile to further improve PMI-based identification. In this study, we identified words to be paraphrased using the training corpus for paraphrase generation. In future work, we plan to identify these words using not only a parallel corpus but also larger data.

[3] https://github.com/awslabs/sockeye

[4] Because the test dataset of GYAFC is multi-reference, the F1 scores of each reference sentence does not reach 100.

	Newsela					GYAFC-E&M				GYAFC-F&R			
	Add	Keep	Del	BLEU	SARI	Add	Keep	Del	BLEU	Add	Keep	Del	BLEU
Source	0.0	60.3	0.0	21.4	2.8	0.0	85.4	0.0	49.1	0.0	85.8	0.0	51.0
Reference	100	100	100	100	70.3	57.2	82.9	61.2	100	56.5	82.7	60.6	100
Dress-LS	2.4	60.7	**44.9**	24.3	**26.6**								
BiFT-Ens						32.1	90.0	58.2	71.4	32.6	90.6	60.9	74.5
Ours (RNN)	**2.8**	**61.1**	36.5	**24.7**	22.8	**33.5**	90.0	**59.9**	**71.7**	**34.3**	**90.9**	**63.1**	**75.9**
Ours (SAN)	**2.5**	**61.3**	38.0	**24.6**	23.3	**35.2**	90.0	**61.2**	**72.1**	**35.3**	**91.1**	**64.0**	**77.0**

Table 3: Comparison with previous models on text simplification in Newsela dataset and formality transfer in GYAFC dataset. Our models achieved the best BLEU scores across styles and domains.

GYAFC-E&M: Informal → Formal	
Source	**mama** so ugly, she scares buzzards off of a meat wagon.
Reference	Your mother is so unattractive she scared buzzards off of a meat wagon.
SAN-BASE	**mama** is so ugly, she scares buzzards off of a meat wagon.
SAN-PMI	The mother is so unattractive that she scares buzzards off of a meat wagon.
GYAFC-F&R: Informal → Formal	
Source	Well, if the one boy **picks** on you, why like him?
Reference	Well, if that one boy bullies you, why the attraction to him?
SAN-BASE	If the one boy **picks** on you, why like him?
SAN-PMI	Well, if the one boy teases you, why like him?

Table 4: Examples of formality transfer. Bolded words are words that are identified as the source style (informal). We succeeded in paraphrasing as follows: mama → mother, picks on → teases.

Table 3 shows a comparison between our models and comparative models. Whereas Dress-LS has a higher SARI score because it directly optimizes SARI using reinforcement learning, our models achieved the best BLEU scores across styles and domains.

Table 4 shows examples of generated paraphrases in formality transfer task. We succeeded in identifying informal expressions of `mama` and `picks`, and successfully paraphrased them. Our proposed method avoids these informal words during beam search, and outputs their synonymous formal expressions, *i.e.*, `mother` and `teases`.

Figure 1 shows the sensitivity of the quality of generated paraphrases to PMI threshold θ on the development dataset. Too low thresholds cause a large amount of constraints, which adversely affect paraphrase quality. However, with a high threshold, the proposed method can achieve high performance stably. Finally, we used a threshold of $\theta = 0.5$ to maximize the BLEU score on the development dataset for formality transfer tasks. Similarly, in the text simplification task, we used a threshold of $\theta = 0.2$.

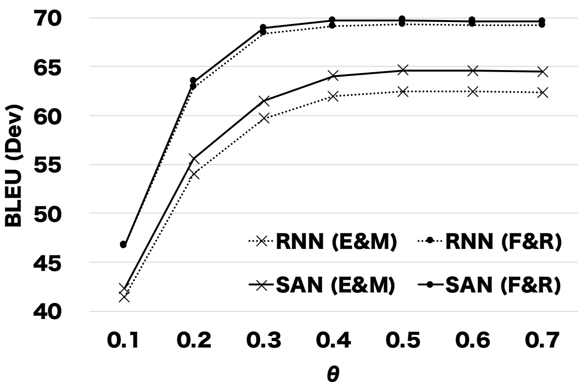

Figure 1: Thresholds of PMI and quality of generated paraphrases on the development dataset.

4 Related Work

4.1 Style-Sensitive Paraphrase Acquisition

Pavlick and Nenkova (2015) worked on a style-sensitive paraphrase acquisition. They used a large-scale raw corpus in each style to calculate PMI scores for each word or phrase and assigned style scores to paraphrase pairs in the paraphrase database (Ganitkevitch et al., 2013; Pavlick et al., 2015). Pavlick and Callison-Burch (2016) further improved style-sensitive paraphrase acquisition based on supervised learning with additional features such as frequency and word embeddings. In this study, as in these previous studies, we have identified words that are strongly related to a particular style. Furthermore, we used these words to control the neural paraphrase generation model and improved the performance of sentential paraphrase generation.

4.2 Lexically Constrained Paraphrasing

Hu et al. (2019b) automatically constructed a large-scale paraphrase corpus[5] via lexically constrained machine translation. In a Czech–English bilingual corpus, sentence pairs of a Czech-to-English machine translation and an English reference can be regarded as automatically generated sentential paraphrase pairs (Wieting and Gimpel, 2018). They used words in reference sentences as positive or negative constraints and succeeded in generating diverse paraphrases via machine translation. In addition, recent work (Hu et al., 2019a) has used lexically constrained paraphrase generation for data augmentation and improve performance in some NLP applications. Unlike these previous studies, we focused on the paraphrase generation as an application. Furthermore, we have shown that negative lexical constraints consistently improve the performance of paraphrase generation applications such as text simplification and formality transfer.

5 Conclusion

To improve the conservative rewriting of the paraphrase generation model, we proposed the identification of words to be paraphrased and the addition of negative lexical constraints on beam search. Experimental results on English text simplification and formality transfer indicated that the proposed method consistently improved the quality of paraphrase generation for both RNN and SAN models across styles or domains. Our proposed method deleted complex or informal words appearing in source sentences and promoted the addition of simple or formal words to paraphrased sentences.

Acknowledgments

We are grateful to Atsushi Fujita, Yuki Arase and Chenhui Chu for helpful discussions. We also thank anonymous reviewers for their constructive comments. This work was supported by JST, ACT-I Grant Number JPMJPR18UB, Japan.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided Open Vocabulary Image Captioning with Constrained Beam Search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations*, pages 1–15.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A New Text Simplification Task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669.

Richard J. Evans. 2011. Comparing Methods for the Syntactic Simplification of Sentences in Information Extraction. *Literary and Linguistic Computing*, 26(4):371–388.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A Toolkit for Neural Machine Translation. *arXiv:1712.05690*.

Chris Hokamp and Qun Liu. 2017. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1535–1546.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019a. Improved Lexically Constrained Decoding for Translation and Monolingual Rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–850.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019b. ParaBank: Monolingual Bitext Generation and Sentential Paraphrasing via Lexically-constrained Neural Machine Translation. *arXiv:1901.03644*.

[5] http://decomp.io/projects/parabank/

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing Modern Language Using Copy-Enriched Sequence to Sequence Models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring Neural Text Simplification Models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 85–91.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. Multi-Task Neural Models for Translating Between Styles Within and Across Languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A Paraphrase Database for Simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 143–148.

Ellie Pavlick and Ani Nenkova. 2015. Inducing Lexical Style Properties for Paraphrase and Genre Differentiation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 218–224.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better Paraphrase Ranking, Fine-grained Entailment Relations, Word Embeddings, and Style Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 425–430.

Matt Post and David Vilar. 2018. Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1314–1324.

Sudha Rao and Joel Tetreault. 2018. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 129–140.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Sanja Štajner and Maja Popović. 2016. Can Text Simplification Help Machine Translation? *Baltic Journal of Modern Computing*, 4(2):230–242.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 451–462.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Wei Xu, Alan Ritter, William B. Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for Style. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 2899–2914.

Xingxing Zhang and Mirella Lapata. 2017. Sentence Simplification with Deep Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1353–1361.