Combining Paraphrase Pre-trained Model and Controllable Rules for Unsupervised Sentence Simplification

Anonymous ACL submission

Abstract

Although neural sequence-to-sequence models for sentence simplification achieve some progress, they still suffer from the data sparsity problem and are lack of controllability.

This paper proposes a two-stage approach for text simplification. First, considering text simplification is closely related to text summarization and paraphrase, we fine-tune the pre-trained model on the dataset of summarization and paraphrase. Further, in order to achieve interpretation and controllablity, we design controllable scorers to evaluate the simplified sentence from three aspects: adequacy, fluency and simplicity, which are applied to sort the generated sentences and output the best one. Experiments show that our approach improves the previous best performance of the unsupervised model by a considerable margin of 5.53 points, achieving a new state-of-the-art result. Our method even performs competitively with supervised models in both automatic metrics and human evaluation.

1 Introduction

Text simplification aims to reduce the linguistic complexity of a sentence while still retains the original information and meaning, which can be applied to multiple scenarios in real life. For instance, it can help non-native speakers learn language (Paetzold, 2016), assist people with aphasia to read (Carroll et al., 1998). Besides, as Alva-Manchego et al. (2020a) note, text simplification can serve as a preprocessing step to improve the performance of other language processing tasks such as parsing (Chandrasekar et al., 1996) and machine translation (Hasler et al., 2017).

With the development of deep learning, neural sequence-to-sequence (Seq2Seq) model becomes the mainstream paradigm for sentence simplification. However, such models encounter two serious problems for text simplification. First, a large number of complex-simple aligned sentence pairs are essential to train the model, but the existing training data is automatically collected from English Wikipedia and its simple version, and thus the quality of training data is unsatisfactory (Xu et al., 2015; Stajner et al., 2015) that prevent the supervised model from obtaining better performance. Second, the evaluation of text simplification is multi-dimensional, which can be mainly divided into three dimensions: adequacy, fluency and simplicity. But the end-to-end neural network models lack interpretability and controllability, which can not be controllable to pay more attention to a certain dimension for a specific application.

Due to the data spasity problem, more and more researchers try to employ unsupervised model for text simplification. Surya et al. (2019) propose a neural auto-encoding framework supported by adversarial loss and diversification loss. Narayan and Gardent (2016) build a pipeline unsupervised framework with lexical simplification, split and deletion. Kumar et al. (2020) propose an iterative edit-based simplification strategy including removal, extraction, reordering and substitution.

The purpose of text simplification is to make sentences simpler, and the simplified ways include paraphrasing, summarization and so on. So paraphrase and summarization are closely related to simplification, and fortunately these two tasks of paraphrase and summarization have large scale and high-quality training data. Arase and Tsujii (2019) show that further training of a pre-trained model on relevant tasks transfers well to similar tasks. In this end, we try to fine-tune the pre-trained model on the data of summarization and paraphrase to help sentence simplification.

In this paper, we propose a two-stage approach for text simplification. First, a pre-trained model is fine-tuned on the dataset of summarization and paraphrase, and we use this model to generate candidate simplified sentences by applying the stochastic decoding strategy. Second, in order to achieve

interpretation and controllablity, we design controllable scorers to re-rank the candidate sentences from three aspects: adequacy, fluency and simplicity, and finally output the sentence that best meets a specific requirement.

Experimental results show that our approach improves the previous state-of-the-art SARI score of the unsupervised model by a considerable margin of 5.53 points. Our method even performs competitively with supervised models in both automatic metrics and human evaluation. Quantitative analysis proves that our model can be controllable towards different simplification directions via adjusting the controllable scorers.

2 Proposed Methods

2.1 Paraphrase-Pegasus

The goal of text simplification is to make sentence simpler, which can be achieved via various ways including text paraphrase, text summarization. Due to the poor quality of the data for text simplification, we try to apply models of paraphrase and summarization to the task of simplification.

Pegasus (Zhang et al., 2020) is a transformer encoder-decoder model specially designed for the task of abstractive text summarization. Pegasus designs the gap-sentence generation pre-training task similar to the summarization task, and achieves state-of-the-art performance on many summarization datasets.

Paraphrase is a widely used method for sentence simplification(Zhao et al., 2018a). Sentences are expressed in a more straightforward way without losing information of the original sentence by effective paraphrase operations. Traditional paraphrase models can achieve effective rewriting of sentences, but the length of the sentence generated by the paraphrase model is always the same as the original sentence. So we fine-tune Pegasus with paraphrase dataset to let the model learn the knowledge of paraphrase while output the shorter sentence.

We adopt the Pegasus paraphrase model[1] published in the Huggingface hub, which fine-tunes the Pegasus model with Google PAWS paraphrasing dataset (Zhang et al., 2019).

We apply the stochastic decoding strategy to generate multiple candidate sentences(In our experiment, the num of candidate sentence is 10). Our model then ranks these candidate sentences using the subsequent controllable scorers and outputs the best one that meets the requirements.

2.2 Controllable Ranking Rules

Different needs of a simplified sentence can be roughly divided into three dimensions: adequacy (the simplified version should retain the semantics of the original sentence), fluency (the simplified sentence should be fluent and grammatical) and simplicity (the simplified version needs to be simpler than the original one). For each dimension, we design a corresponding indicator to quantitatively evaluate it.

Adequacy We calculate the semantic similarity of generated sentences and original sentences to measure the adequacy. SimCSE (Gao et al., 2021) proposes a simple contrastive learning framework that achieves state-of-the-art results on semantic textual similarity tasks. We use the SimCSE to get the similarity score of paired sentences, and adopt the min-max normalization on similarity scores. The adequacy score A is calculated as:

$$A = \frac{V_{sim} - Min_{sim}}{Max_{sim} - Min_{sim}} \quad (1)$$

where V_{sim} means the similarity score calculated by SimCSE. The Min_{sim} and Max_{sim} is the minimum and maximum score of similarity score of all candidate sentences. The following indicator calculations all use this normalization.

Fluency We use perplexity to measure the fluency of the generated sentences, which is defined as the exponentiated average negative log-likelihood of a sequence. The fluency score F is also normalized using min-max function. In our work, we use the GPT-2 model[2] to calculate perplexity.

Simplicity Evaluating the simplicity of a sentence is a difficult task. We break it down into three aspects: length simplicity, lexical simplicity, and syntactic simplicity. For length simplicity S_f, we use the traditional $fkgl$ indicator to measure it. The lexical complexity score S_l is computed by taking the log-ranks of each word in the frequency table(Martin et al., 2019). For syntactic simplicityS_s, we use the average depth of the constituency tree to measure it. In the experiment, we use the Benepar (Kitaev and Klein, 2018) to construct the constituency tree. The overall simplicity score S is the average of three values: $S = (S_f + S_l + S_s)/3$.

[1] https://huggingface.co/tuner007/pegasus_paraphrase

[2] https://huggingface.co/gpt2-medium

The final ranking score is the weighted average of the above three aspects:

$$score = \omega_1 \times F + \omega_2 \times A + \omega_3 \times S \quad (2)$$

In our experiment, ω_1, ω_2, ω_3 are set to 1, 1, 1 respectively.

3 Experimental Setup

3.1 Datasets

Our model is evaluated on two benchmark test datasets for text simplification: Turkcorpus (Xu et al., 2016) and Asset (Alva-Manchego et al., 2020b) with the same 359 source sentences. Both datasets are created by employing Amazon Mechanical Turk workers to simplify sentences.

3.2 Comparing Methods

Our model is totally unsupervised, and so we mainly compare our model with previous unsupervised methods but also list some supervised methods for reference.

Unsupervised models (1) **UNMT** (Artetxe et al., 2017) is an unsupervised NMT framework that can be directly applied to sentence simplification. (2) **UNTS** (Surya et al., 2019) is the first unsupervised neural network model specifically for sentence simplification. (3) **RM+EX+LS+RO** (Kumar et al., 2020) is an iterative, edit-based approach to unsupervised sentence simplification. According to the supported operations, the model has many variants: RM+EX, RM+EX+LS, RM+EX+RO.

Supervised models (1) **NTS-SARI** (Nisioi et al., 2017) is a vanilla RNN-based neural text simplification model. (2) **Dress** and **Dress-Ls** (Zhang and Lapata, 2017) adopt deep reinforcement learning method. (3)**DMASS-DCSS** (Zhao et al., 2018b) is a transformer based model with integration of the Simple PPDB. (4) **EditNTS** (Dong et al., 2019) is a neural sequence tagging model that learns explicit edit operations. (5) **ACCESS** (Martin et al., 2020) is a controllable model which can control certain attributes of generated sentences.

3.3 Evaluation Metrics

For automatic evaluation of our method and comparing methods, we use three metrics: SARI, BLEU, FKGL. **BLEU** (Papineni et al., 2002) is widely used for measuring sentence simplification before SARI is proposed. **SARI** (Xu et al., 2016) is the most commonly used and effective metric for sentence simplification. **FKGL** (Flesch, 1948) is a classical metric for measuring the readability of sentences. We calculate these three evaluation metrics by the EASSE (Alva-Manchego et al., 2019).

4 Results And Analysis

4.1 Automatic Evaluation Results

Table 1 shows the experimental results. Considering the SARI metric, our method achieves new state-of-the-art results on both datasets among unsupervised models. On Asset, our method improves the previous state-of-the-art unsupervised model by a considerable margin of 5.53 points, and even outperforms all supervised models. On Turkcorpus, we also obtain an obvious improvement of 1.14 points.

We also report three operations in SARI: $Keep$, Add and $Delete$. Our scores on Add and $Delete$ are generally higher than other models while $Keep$ is lower. Our model is more willing to change and simplify sentences while the previous models are conservative and tend to retain the original sentence.

In terms of BLEU, the score of our model is generally lower than other models. This is because the BLEU indicator reflects the token overlap between the generated sentence and the original sentence. The previous models tend to perform keep operation, so they get better BLEU scores.

4.2 Controllability of Ranking Scorer

An advantage of our method is that, by changing the weight of each scoring item, we can guide our model to generate sentences towards different simplification directions, and thus to meet specific requirements. Table 2 analyzes the effect of relative weights of scorers. Here besides BLUE, SARI and FKGL, we also utilize Compression Ratio (CR), which represents the compression ratio of generated sentences with respect to its source sentence.

As the weight of adequacy increases, the BLEU score increases from 67.41 to 71.81 and the SARI-Keep score increases from 52.27 to 55.01. It means the model becomes more conservative and tends to keep more the original content. As the weight of simplicity increases, the SARI-Delete increases from 61.97 to 74.00 and the Compression Ratio decreases from 0.79 to 0.74, which means the model tends to delete unimportant information in the sentence to make it simpler. As the weight of fluency increases, the SARI increases from 41.68 to 42.20,

	Model	Asset						Turkcorpus					
		BLEU	**SARI**	**SA**	**SK**	**SD**	**FKGL**	**BLEU**	**SARI**	**SA**	**SK**	**SD**	**FKGL**
Supervised	NTS-SARI	84.19	34.02	2.84	59.48	39.74	8.18	84.06	36.11	2.89	71.52	33.90	8.18
	Dress	84.24	37.07	2.52	56.54	52.15	7.53	78.16	36.84	2.50	65.65	42.36	7.53
	Dress-Ls	86.39	36.59	2.38	57.30	50.10	7.66	81.08	36.97	2.35	67.23	41.33	7.66
	DMASS-DCSS	71.44	38.68	4.36	60.29	51.37	7.73	73.29	39.92	4.94	70.15	44.67	7.73
	EditNTS	86.20	34.94	2.41	59.73	42.69	8.38	86.57	37.66	2.71	72.08	38.18	8.38
	ACCESS	75.99	40.13	6.54	**62.99**	50.85	7.29	76.36	**41.38**	6.58	72.79	44.78	7.29
Unsupervised	UNMT	68.41	32.78	1.42	56.45	40.47	8.97	72.55	34.84	1.43	68.48	34.60	8.97
	UNTS	76.14	35.19	0.83	58.75	45.98	7.60	76.44	36.29	0.83	69.44	38.61	7.60
	UNTS_10K	76.28	35.20	0.98	59.89	44.71	8.02	78.03	37.15	1.12	71.34	38.99	8.02
	RM+EX	**89.55**	32.61	0.61	59.91	37.30	7.43	**90.24**	35.88	0.84	**73.14**	33.65	7.43
	RM+EX+LS	75.55	36.56	1.18	58.16	50.34	**7.25**	74.84	37.48	1.59	68.20	42.65	**7.25**
	RM+EX+RO	84.75	33.05	0.78	59.40	38.98	7.53	86.31	36.07	0.99	72.36	34.86	7.53
	RM+EX+LS+RO	70.63	36.67	1.29	57.40	51.33	7.33	71.24	37.27	1.68	67.00	43.12	7.33
Ours	Paraphrase-Pegasus	73.75	41.74	6.76	55.31	63.14	7.60	67.16	**38.62**	6.53	59.42	49.90	7.60
	Paraphrase-Pegasus-C	71.81	**42.20**	**7.60**	55.01	**64.00**	7.30	64.25	38.55	**6.67**	58.63	**50.36**	7.30

Table 1: Experimental results on the Asset and Turkcorpus datasets comparing with previous methods, where Paraphrase-Pegasus-C denotes adding our controllable ranking rules on the pre-trained models. **SA** denotes SARI-Add, **SK** denotes SARI-Keep, **SD** denotes SARI-Delete.

Weight	**BLEU**	**SARI**	**SA**	**SK**	**SD**	**CR**
weight of Adequacy						
0	67.41	41.76	7.25	52.27	65.75	0.68
0.25	68.26	41.87	7.33	52.94	65.33	0.69
0.5	69.52	42.14	7.44	54.01	64.97	0.71
0.75	70.90	42.20	7.53	54.57	64.50	0.73
1	71.81	42.20	7.60	55.01	64.00	0.74
weight of simplicity						
0	72.45	41.68	7.20	55.87	61.97	0.79
0.25	72.23	41.78	7.22	55.84	62.29	0.79
0.5	72.61	42.07	7.54	55.73	62.94	0.77
0.75	72.28	42.14	7.65	55.34	63.41	0.76
1	71.81	42.20	7.60	55.01	64.00	0.74
weight of fluency						
0	71.72	41.68	6.54	54.97	63.52	0.73
0.25	72.17	41.89	6.80	55.18	63.69	0.74
0.5	72.26	41.94	7.11	55.00	63.71	0.74
0.75	71.87	42.12	7.32	55.04	64.01	0.74
1	71.81	42.20	7.60	55.01	64.00	0.74

Table 2: Results on the controllability of scorer. **SA** denotes SARI-Add, **SK** denotes SARI-Keep, **SD** denotes SARI-Delete, **CR** denotes Compression Ratio.

which proves the fluency of generated sentence has positive impact on the effect of simplification. These results prove that by changing the weight of each scoring item in the ranking rules, we can control the output sentences to pay more attention to a specific aspect.

4.3 Human Evaluation

We also conduct human evaluation to compare our system outputs with the best unsupervised model, using a five-point Likert scale. We randomly chose 30 sentences from the Asset test set, and three volunteers were given different model outputs in a randomized order. They were asked to evaluate generated sentences from three aspects: adequacy, simplicity, and fluency. The Spearman correlation coefficients between annotators are high. Table 3 shows that our model's score surpasses the best unsupervised model by a large margin on all metrics. Although Paraphrase-Pegaus-C is slightly worse than Paraphrase-Pegaus in Fluency and Simplicity, it greatly improves Adequacy which is the weakest aspect of Paraphrase-Pegasus.

Model	**Fluency**	**Adequacy**	**Simplicity**	**Avg**
RM+EX+LS+RO	3.87	3.40	2.16	3.14
Paraphrase-Pegasus	4.89	3.62	3.07	3.86
Paraphrase-Pegasus-C	4.81	3.92	3.00	3.91
Spearman	0.55	0.70	0.77	0.67

Table 3: Human Evaluation Results.

5 Conclusion

In this paper, we propose an unsupervised sentence simplification approach combining pre-trained models and controllable ranking rules. To alleviate the data sparsity problem for sentence simplification, we borrow the dataset of summarization and paraphrase to fine tune the pre-trained model, and experiments show our method achieves a new state-of-the-art performance among unsupervied methods, and even performs competitively with supervised models. In future work, we will employ linguistic knowledge for sentence simplification, and we will explore how to comprehensively and accurately evaluate the task of text simplification.

References

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020a. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Fernando Emilio Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020b. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *ACL*, pages 4668–4679. Association for Computational Linguistics.

Fernando Emilio Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. Easse: Easier automatic sentence simplification evaluation. In *EMNLP/IJCNLP (3)*, pages 49–54. Association for Computational Linguistics.

Yuki Arase and Jun'ichi Tsujii. 2019. Transfer fine-tuning: A BERT case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5393–5404, Hong Kong, China. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *CoRR*, abs/1710.11041.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Citeseer.

Raman Chandrasekar, Christine Doran, and Srinivas Bangalore. 1996. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. Editnts: An neural programmer-interpreter model for sentence simplification through explicit editing. In *ACL (1)*, pages 3393–3402. Association for Computational Linguistics.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Eva Hasler, Adrià de Gispert, Felix Stahlberg, Aurelien Waite, and Bill Byrne. 2017. Source sentence simplification for statistical machine translation. *Computer Speech & Language*, 45:221–235.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *ACL (1)*, pages 2676–2686. Association for Computational Linguistics.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *ACL*, pages 7918–7928. Association for Computational Linguistics.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *LREC*, pages 4689–4698. European Language Resources Association.

Louis Martin, Samuel Humeau, Pierre-Emmanuel Mazaré, Antoine Bordes, Éric Villemonte de la Clergerie, and Benoît Sagot. 2019. Reference-less quality estimation of text simplification systems. *CoRR*, abs/1901.10746.

Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *INLG*, pages 111–120. The Association for Computer Linguistics.

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 (volume 2: Short papers)*, pages 85–91.

Gustavo Henrique Paetzold. 2016. *Lexical simplification for non-native English speakers.* Ph.D. thesis, University of Sheffield, UK. British Library, EThOS.

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

Sanja Stajner, Hanna Béchara, and Horacio Saggion. 2015. A deeper exploration of the standard pb-smt approach to text simplification and its evaluation. In *ACL (2)*, pages 823–828. The Association for Computer Linguistics.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *ACL (1)*, pages 2058–2068. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Trans. Assoc. Comput. 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.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *EMNLP*, pages 584–594. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. In *NAACL-HLT (1)*, pages 1298–1308. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018a. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018b. Integrating transformer and paraphrase rules for sentence simplification. In *EMNLP*, pages 3164–3173. Association for Computational Linguistics.