Front. Comput. Sci., 2023, 17(1): 171303
https://doi.org/10.1007/s11704-022-1244-0

RESEARCH ARTICLE

Unsupervised statistical text simplification using pre-trained language modeling for initialization

Jipeng QIANG (✉)[1], Feng ZHANG[1], Yun LI (✉)[1], Yunhao YUAN[1], Yi ZHU[1], Xindong WU[2,3]

1 Department of Computer Science, Yangzhou University, Yangzhou 225127, China
2 Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei 23009, China
3 Mininglamp Academy of Sciences, Mininglamp, Beijing 100089, China

Abstract Unsupervised text simplification has attracted much attention due to the scarcity of high-quality parallel text simplification corpora. Recent an unsupervised statistical text simplification based on phrase-based machine translation system (UnsupPBMT) achieved good performance, which initializes the phrase tables using the similar words obtained by word embedding modeling. Since word embedding modeling only considers the relevance between words, the phrase table in UnsupPBMT contains a lot of dissimilar words. In this paper, we propose an unsupervised statistical text simplification using pre-trained language modeling BERT for initialization. Specifically, we use BERT as a general linguistic knowledge base for predicting similar words. Experimental results show that our method outperforms the state-of-the-art unsupervised text simplification methods on three benchmarks, even outperforms some supervised baselines.

Keywords text simplification, pre-trained language modeling, BERT, word embeddings

1 Introduction

The aim of text simplification (TS) is to reduce the complexity of the text while conveying the same meaning with the original text. Most recent methods including neural text simplification methods have addressed TS as a monolingual machine translation task that translating from complex sentences to simplified sentences [1–5]. In contrast to machine translation task, existing parallel simplification benchmarks WikiSmall [6] and WikiLarge [5] which align sentences from English Wikipedia and Simple English Wikipedia not only are relatively small-scale, but also contain a large proportion of noise sentence pairs. For example, the authors [7] pointed out that about 50 percent of the sentence pairs from the above benchmarks have the following two questions: inaccurate simplification (not aligned or only partially aligned) and inadequate simplification (not much simpler than complex sentence). Additionally, parallel TS corpus is difficult to obtain in all languages other than English. Unsupervised approaches [8,9] which forgo the use of parallel corpora are an appealing solution to overcoming the paucity of data.

In [10], we introduced an unsupervised statistical TS method based on phrase-based machine translation system (UnsupPBMT). Considering supervised TS systems need parallel corpus to initialize phrase tables, UnsupPBMT only utilizes word embedding modeling and word frequencies obtained from the English Wikipedia to initialize the phrase tables. Here, word embedding modeling is used to find similar words, and word frequency is used to reflect the complexity of each word. When initializes the phrase tables, UnsupPBMT extracts the most frequent 50,000 words, and aligns each of them to its 200 most similar words. Although UnsupPBMT achieves good performance, we found that it has the following two issues:

(1) UnsupPBMT aligns a large number of non-similar words to initialize the phrase tables, which will bring noise to the simplification system. Because the word embedding modeling is trained on word-word co-occurrence counts, it captures high-relevant words, not just high-similar words. We give two examples shown in Table 1. Many non-similar words shown in bold are generated by the word embedding modeling.

(2) Many simple high-similar words are hard to find using a word embedding modeling, which is not following the aim of TS task. Because the words with high frequency often occur different contexts, their similarity values are not usually very high when calculating the similarities between them and some specific words with low frequency. As shown in Table 1, for the word “enormous”, its simple high-similar words “big”, “large” and “vast” are not produced by word embedding modeling. For the word “merciless”, its simple high-similar words “cruel” and “severe” are also not produced by word embedding modeling.

As all known, pre-trained language modeling (LM) trained on unstructured text can implicitly store and retrieve knowledge [11–14]. For solving the above two issues, we propose a novel unsupervised statistical text simplification

Received May 10, 2021; accepted September 6, 2021

E-mail: jpqiang@yzu.edu.cn; liyun@yzu.edu.cn

Table 1 The high-similar words of "enormous" and "merciless" are obtained by word embedding modeling (abbr. EM) and BE (abbr. BERT), respectively. For each method, we generate the top 15 high-similar words. Non-similar words and words with inconsistent part-of-speech are shown in bold

Word	Method	High-similar words
Enormous	EM	Immense, huge, massive, gigantic, colossal, considerable, sizable, myriad, mammoth, **undoubtedly**, **staggering**, stupendous, **significant**, towering, **seemingly**
	BE	Big, large, immense, huge, giant, vast, ample, enlarged, oversized, gigantic, massive, excessive, overwhelming, tremendous, ferocious,
Merciless	EM	Ruthless, relentless, vicious, brutal, inhuman, heartless, **unyielding**, callous, **ruthlessly**, **mercilessly**, cold-blooded, murderous, savage, **punishing**, **torturous**
	BE	Ruthless, cruel, vicious, mean, cold, severe, relentless, lethal, murderous, fierce, horrible, harsh, violent,fatal, savage

wherein a pre-trained LM is used as a general knowledge base for initialization in the paper. Specifically, we generate many high-similar words for each word using a pre-trained LM. The LM collects high-similar words in the unlabeled corpus to capture distinctive information in different contexts. As shown in Table 1, the top high-similar words by our method not only have high semantic similarity, but also cover all possible high-similar words.

The contributions of our paper are as follows:

(1) We propose an unsupervised statistical text simplification method based on pre-trained language modeling. We propose a good strategy to find synonyms or high-similar words using pre-trained language modeling BERT, compared with synonym dictionary and word embedding modeling.

(2) On three benchmark datasets, our method outperforms significantly all unsupervised text simplification methods and has comparable performance to strong supervised methods.

The following sections are organized as follows: Section 2 details the procedure of unsupervised statistical text simplification method UnsupPBMT; Section 3 presents our proposed method; Section 4 shows the experimental results; Section 5 describes the related work; Section 6 summarizes the paper.

2 The UnsupPBMT method

In this section, we will present the UnsupPBMT method we proposed in [10]. In the next section, we will point out the drawbacks of UnsupPBMT and propose a new unsupervised text simplification system to cope with these drawbacks.

The framework of the UnsupPBMT method is illustrated in Fig. 1. UnsupPBMT adopts a phrase-based machine translation (PBMT) system [15] as the underlying backbone model. UnsupPBMT models the simplification of an input sentence x into y according to: $P(y|x) = \arg\max_y P(x|y)P(y)$, where $P(x|y)$ is the probability that y would be simplified into x which is derived from so-called "phrase tables", and $P(y)$ is is the probability of y assigned by a language model. The execution process of UnsupPBMT is shown in Algorithm 1. UnsupPBMT first obtains some prior knowledge including word embeddings (line 1), word frequency (line 2), simple sentence set, and complex sentence set (line 3). UnsupPBMT initializes phrase tables and learns language models without the need for parallel text simplification corpus (lines 4−5), and then builds an initial UnsupPBMT method (line 6). Based on the initial UnsupPBMT system, UnsupPBMT builds a synthetic parallel corpus (line 7), which can turn the unsupervised problem into the supervised problem through iterative back-translation techniques [16] (lines 8−13).

Phrase tables While prior work for initializing phrase tables relied on a parallel corpus, UnsupPBMT only utilizes word embeddings and word frequency obtained from Wikipedia. UnsupPBMT adopts the Glove algorithm [17] to learn word embeddings. Word frequencies counted from Wikipedia will help to find the simplest words from a set of similar words. In general, the higher the frequency, the easier the word. Assume that $e(w)$ is the embedding of word w and $f(w)$ is the frequency of word w. Phrase tables PT are initialized by the scores between one word w_i to the other word w_j which is calculated by,

$$P(w_j \mid w_i) = \begin{cases} \dfrac{f(w_j)}{f(w_i)} \cos(e(w_i), e(w_j)), \\ \qquad \text{if } P(w_j \mid w_i) < 1, \\ 1, \text{otherwise}, \end{cases} \tag{1}$$

where cos is cosine similarity. If both words w_i and w_j have higher similarity and w_j has higher frequency than w_i, $P(w_j \mid w_i)$ has a high score. The underlying principle of Eq. (1) is that the words with high frequency should have higher

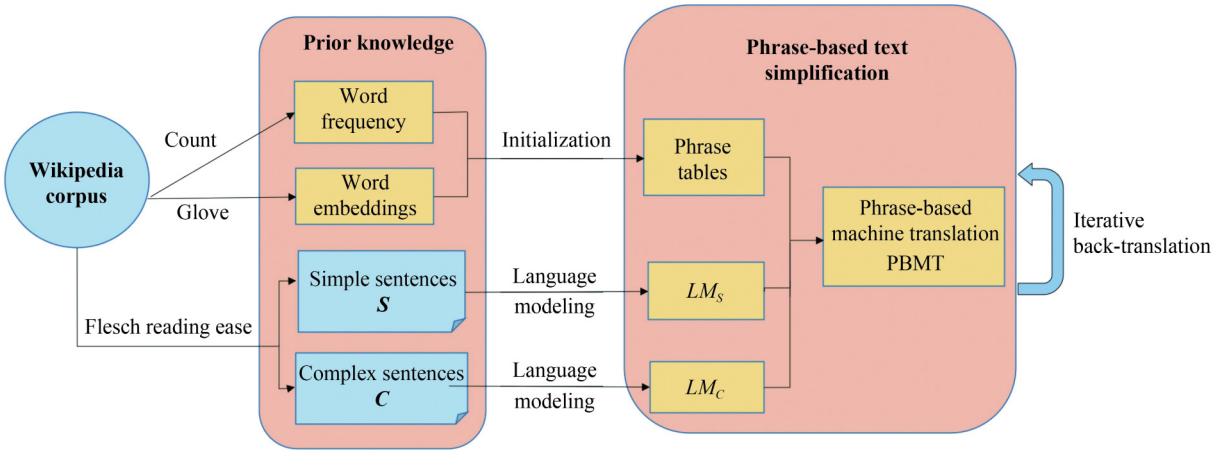


Fig. 1 Architecture of UnsupPBMT method

Algorithm 1 UnsupPBMT method

Learn word embedding using Glove from Wikipedia.
Compute word frequency from Wikipedia.
Gather simple corpus S and complex corpus C from Wikipedia using Flesch reading-ease score.
Initialize phrase tables(PT) using Eq.(1)
Learn two langugage models LM_S and LM_C from S and C using KenLM.
Combine (PT, LM_S and LM_C) to build $m^0_{C\to S}$ using PBMT.
Simplify C for yielding S^0 using $m^0_{C\to S}$.
for i=1 to N **do**
    Train model $m^i_{S\to C}$ using (S^{i-1}, C).
    Simplify S for yielding C^i using $m^i_{S\to C}$.
    Train model $m^i_{C\to S}$ using (C^{i-1}, S).
    Simplify C for yielding S^i using $m^i_{C\to S}$.
end for

scores than the words with low frequency when all of them have very high similarity values.

When initializing phrase tables, UnsupPBMT extracts the most frequent 50,000 words, and calculates their scores between each word and its 200 most similar words, resulting in a phrase table of 10 million entries.

Language modeling UnsupPBMT extracts simple sentence set S and complex sentence set C from Wikipedia to learn simple language modeling (LM_S) and complex language modeling (LM_C). UnsupPBMT chooses one text readability assessment measure (Flesch reading ease score (FRES) [18]) to calculate the scores of all sentences from Wikipedia texts. FRES grades the text from 0 to 100. Higher scores indicate sentences that are easier to read, and lower scores mean sentences that are more difficult to read. We delete millions of sentences whose scores around the median to establish a clear boundary between simple corpus and complex corpus. UnsupPBMT collects 10 million sentences whose scores less than 100 as simple corpus S, and 10 million sentences whose scores greater than 10 as complex corpus C.

The two sentence sets S and C are applied to learning two different n-gram language models (namely, LM_S and LM_C) by the KenLM method [19]. The two language models will remain constant in the whole training iterations in UnsupPBMT.

Iterative back-translation The initial UnsupPBMT system is named as $m^0_{C\to S}$ using the PBMT system based on the above phrase tables PT and two language models (LM_S and LM_C). The aim of iterative back-translation [16,20] reverses the daunting unsupervised task into a supervised learning task. After obtaining $m^0_{C\to S}$ system, UnsupPBMT builds a synthetic simple sentence set S^0 by simplifying the complex sentence set C using $m^0_{C\to S}$. Given the synthetic parallel complex-simple sentence pairs (S^0, C), UnsupPBMT trains and tunes a standard PBMT system $m^1_{S\to C}$ over it from simple sentence set to complex sentence set. Next, UnsupPBMT executes both generation and training processes but in the reverse direction.

3 The UnsupPBMT method using pre-trained language modeling for initialization

In [10], we found that UnsupPBMT can achieve good performance, in which the quality of phrase table plays a crucial role. UnsupPBMT initializes the phrase table by calculating the similarities between words using word embeddings. But, when choosing the 200 most similar words for each word, we found that the phrase tables in UnsupPBMT contain a lot of dissimilar words because word embedding modeling only considers the relevance between words.

In this section, we introduce a novel method by searching similar words with pre-trained language modeling. Intuitively, words that are interchangeable most of the time are likely to have similar meanings. We will use the pre-trained language modeling BERT to predict which words have similar meanings with the original word.

3.1 Overview

The overview of the process of obtaining similar words using BERT is illustrated in Fig. 2. Specifically, for each word in the vocabulary, we first search different sentences that contain this word from the Wikipedia corpus. Here, we search for many different sentences instead of one sentence, because word meanings are contextualized. If we only use one sentence to predict, the coverage of similar words is limited. For example, as shown in Fig. 2, the word "executes" in the second sentence does not contain "processes" and "terminates".

For each sentence, we use BERT masked language model (MLM) to produce similar words. For each sentence, we predict the top 10 most similar words using BERT. Finally, we rank these similar words by calculating their predicted counts in the sentence set. These similar words generated by BERT will replace the similar words using word embeddings to initialize the phrase tables. Below, we will explain the similar words generated by BERT in detail.

3.2 Masked similarity prediction

In this subsection, we will introduce how to get the prediction words for each sentence using the pre-trained masked language modeling BERT.

Generally, after replacing the original word into "[MASK]" symbol, existing methods [11,21] directly feed its contextualized vector $\boldsymbol{h} \in \mathbb{R}$ generated by the BERT encoder to the MLM head, which will output a probability distribution over the entire vocabulary V, indicating the likelihood of each word w appearing at this position. Then, the top-n predicted words are chosen as valid replacements for the original word.

$$p(w|\boldsymbol{h}) = \text{Softmax}(W_2\sigma(W_1\boldsymbol{h} + \boldsymbol{b})), \tag{2}$$

where $\sigma(\cdot)$ is the activation function; $W_1 \in \mathbb{R}^{h\times h}$, $\boldsymbol{b} \in \mathbb{R}^h$, and $W_2 \in \mathbb{R}^{V\times h}$ are learnable parameters that have been pre-trained with the MLM objective of BERT.

If we directly adopt Eq. (2) to obtain the similar words, it overlooks the following two problems:

(1) The predicted words are only related to the context regardless of the complex word itself.

(2) The predicted words take no account of simplicity. The aim of the task hopes to find the simple and similar words of

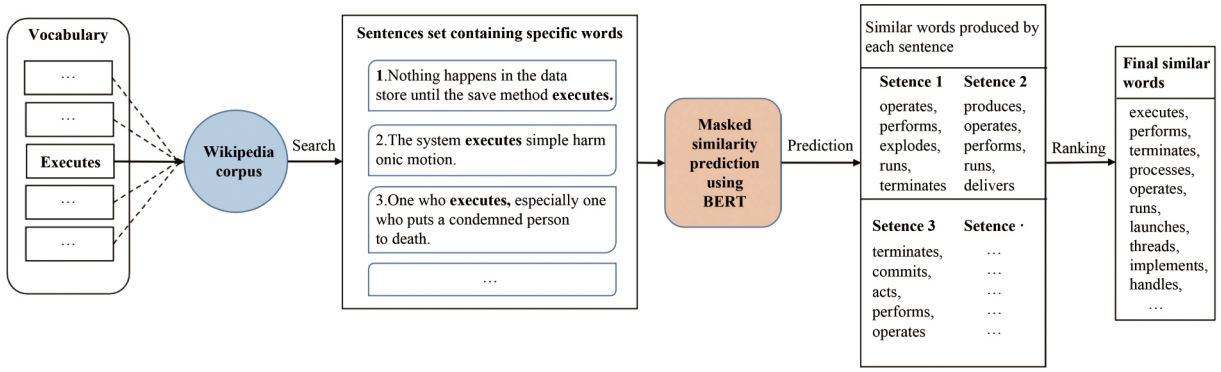


Fig. 2 The overview of the process of obtaining similar words using BERT

the original word.

For solving the first issue, we concatenate the original sentence and the masked sentence as a sentence pair, and feed the sentence pair into BERT to predict the "[MASK]" symbol. This is based on the fact that BERT is adept at dealing with sentence pairs because of the next sentence prediction (NSP) task adopted by BERT. The top words from the predictions are not only related to the complex word, but also are produced by considering the context.

To address the second issue, we will re-rank the predicted words based on simplicity and similarity, and choose the top-ranked words as the similar words.

Giving predicted words $C = \{c_1, c_2, ..., c_k\}$ for the original word, we incorporate the following three features to rank these words: probability, frequency, and similarity, where k is the number of predicted words.

Probability (r_{prob}) We choose the probability distribution of the vocabulary corresponding to the mask word as one feature. Because the prediction already incorporates the contextual information, the probability of prediction is a crucial feature that includes the information of both the context and the complex word itself. If the probability of one predicted word is higher, then it will have a higher ranking.

Frequency (r_{fre}) Frequency-based ranking strategy is one of the most popular choices by lexical simplification and quite effective [22]. In general, the more frequency a word is used, the most familiar it is to readers. We adopt the Zipf scale created from the SUBTLEX lists [23], because some experiments [24,25] revealed that word frequencies from this corpus correlate with human judgments on simplicity than many other more widely used corpora, such as Wikipedia. SUBTLEX is composed of over six million sentences extracted from subtitles of assorted movies. The Zipf frequency of a word is the base-10 logarithm of the number of times it appears per billion words.

Similarity (r_{sim}) The similarity between the original word and the predicted word is chosen as a feature. We choose the word embedding vectors used in UnsupPBMT, and choose the cosine similarity metric to compute the similarity. If the similarity value between the predicted word and the original word is greater, then the predicted word will have a higher ranking.

Each of the features captures one aspect of the suitability of the predicted word to replace the original word. We computes three different rankings (r_{fre}, r_{sim} and r_{prob}) according to their scores for all predicted words, respectively. The final ranking for all predicted words is computed as follows,

$$f\_r = \lambda_1 r_{fre} + \lambda_2 r_{prob} + \lambda_3 r_{sim}, \quad (3)$$

where f_r denotes the final ranking of C, the weights λ_1, λ_2 and λ_3 balance the relative importance of the different features, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Finally, we choose the top 10 rankings as the most similar words from the 20 predicted words in our method.

4 Experiments

4.1 Experimental setup

(1) DataSets

Three benchmark datasets in text simplification (WikiSmall, WikiLarge, and NewSela) are used to do experiments.

The WikiSmall [6] contains complex and simple sentence pairs by automatically aligning from English Wikipedia and Simple English Wikipedia. The training/development/test set contain 89,042/205/100 sentence pairs.

The WikiLarge [5] is also from English Wikipedia and Simple English Wikipedia, which incorporates the simplified data set created earlier, such as WikiSmall. The training set of WikiLarge has 296,402 sentence pairs. WikiLarge includes 8 (reference) simplifications for 2,359 sentences which are split into 2,000 for development and 359 for testing. Its validation and test sets are taken from Turkcorpus, where each original sentence has 8 human simplifications created by Amazon Mechanical Turk workers.

NewSela is a collection of 1,840 news articles written by professional editors at five reading levels for children. We use the standard split following Zhang and Lapata [5]. The training/development/test set contain 95,208/1,129/1,077 sentence pairs.

(2) Metrics

Following previous work, we choose two widely used evaluation metrics (SARI and FKGL) in the text simplification task.

SARI [4] evaluates the quality of the system output sentence by comparing the system output sentence with the original sentence and the reference sentence. Xu et al. [4] demonstrated that SARI correlates most closely to human judgments in the text simplification task. Therefore, SARI is commonly treated as the most important measurement.

FKGL [26] is used to measure the readability of the system

output text by considering the sentence length and word length. The lower value of FKGL indicates a simpler sentence.

(3) Methods for comparison

We compare our method UnsupPBMT-BERT with the following methods:

PBMT-R [3] is a machine translation sentence simplification method based on phrase table and syntactic reordering.

Hybrid [27] performs sentence segmentation and deletion operations based on text representation structure, and then uses PBMT-R model to simplify sentences.

EncDecA [2] is a vanilla Seq2Seq model with an attention mechanism.

Dress and **Dress-LS** [5] are Seq2Seq model combined with deep reinforcement learning framework.

EntPar [28] is based on multi-task learning.

Edit-NTS [29] is a supervised edit-based neural model.

ACCESS [1] parametrizes a Seq2Seq model on a given attribute of the target simplification, e.g., its length, by prepending a special token at the beginning of the source sentence.

UNMT and **UNTS** [8] are two unsupervised text simplification methods, which are inspired by unsupervised neural machine translation.

UnsupPBMT [10] is a state-of-the-art unsupervised method for text simplification based on a phrase-based machine translation system.

UnsupPBMT-BERT is the proposed method in the paper. In our experiments, we search 50 different sentences for each word. Finally, we select the top 20 similar words including the original word for WikiLarge and the top 100 words for WikiSmall and NewSela. For BERT modeling, we use BERT-Large, Uncased (Whole Word Masking) pretrained on BooksCorpus and English Wikipedia, which consists of 24-layer, 1024-hidden, 16-heads, 340M parameters.

The above eight methods (PBMT-R, Hybrid, EncDecA, Dress, Dress-LS, EntPar, Edit-NTS, and ACCESS) are supervised TS methods, and the rest four methods (UNMT, UNTS, UnsupPBMT, and UnsupPBMT-BERT) are unsupervised TS methods.

4.2 Comparison with existing methods

Table 2 summarizes the evaluation results of our method on three corpora.

We can see that our method UnsupPBMT-BERT obtains a SARI score of 40.10 on WikiLarge, surpassing UnsupPBMT's 39.08, and even outperforming most of the previous supervised methods, excluding ACCESS. These results show that our model has indeed learned the ability to simplify sentences. Compared with the other two unsupervised methods (UNMT and UNTS), UnsupPBMT-BERT significantly outperforms them. On the FKGL metric, UnsupPBMT-BERT achieves low FKGL scores compared with UnsupPBMT, which means that the simplified sentences by UnsupPBMT-BERT are indeed simpler. The results show that pre-trained language modeling is fit for finding high-similar words.

For the WikiSmall experiments in Table 2, our method UnsupPBMT-BERT outperforms UnsupPBMT on two metrics, and is also competitive with state-of-the-art supervised methods. It also verifies that pre-trained language modeling for initialization is better than word embedding modeling.

For the NewSela experiments in Tabel 2, our method UnsupPBMT-BERT also outperforms UnsupPBMT on two metrics, but underperforms most of the supervised methods. For the NewSela dataset, we notice that the length of the simplified sentences is less than the complex sentences. For the two methods UnsupPBMT and UnsupPBMT-BERT, they are more concerned about the replacement, not deletion. We observe that lexical simplification is important in WikiLarge dataset. Therefore, the improvement of our method is more obvious compared with the results on NewSela.

4.3 Qualitative study

All of the above experiments are quantitative analyses of UnsupPBMT-BERT. Here, we also qualitatively compare the similar words produced by our method (BERT) and

Table 2 Performance of baselines and our method on three corpora. ↑ The higher, the better. ↓ the lower, the better. _ indicates the results that is not found in the original paper

Method	WikiLarge		WikiSmall		NewSela	
	SARI ↑	FKGL ↓	SARI ↑	FKGL ↓	SARI ↑	FKGL ↓
Baselines						
Complex	28.70	8.11	4.34	12.40	2.74	8.65
Reference	49.89	8.26	63.62	8.96	70.25	3.48
Supervised methods						
PBMT-R (2012)	38.56	8.30	15.97	11.52	15.77	7.95
Hybrid (2014)	31.40	**4.70**	30.46	9.55	30.00	4.15
EncDecA (2017)	35.66	8.67	13.61	11.41	24.12	5.49
Dress (2017)	37.08	6.79	27.48	7.62	27.37	4.19
Dress-LS (2017)	37.27	6.62	27.24	7.55	26.63	4.21
EntPar (2018)	37.45	7.41	28.24	**6.93**	**32.98**	**1.38**
EditNTS (2019)	38.22	7.30	**32.35**	5.47	31.41	3.40
ACCESS (2020)	**41.87**	7.22	–	–	–	–
Unsupervised methods						
UNMT (2019)	33.72	8.23	–	–	–	–
UNTS (2019)	35.29	7.84	–	–	–	–
UnsupPBMT (2021)	39.08	8.26	25.12	10.66	23.75	7.36
UnsupPBMT-BERT	**40.10**	**7.52**	**29.08**	**7.04**	**27.36**	**5.55**

UnsupPBMT (Word2Vec).

Firstly, we randomly choose 100 words with low frequency from the SUBTLEX lists [23], and manually evaluate the generated similar words by BERT and Word2Vec for these 100 words. Here, we evaluate the top 50 similar words by BERT and Word2Vec, and check the proportion with which the generated words are similar. Table 3 shows the results of the experiments for the top 10, 20, 30, 40, and 50 words, respectively. We see that BERT obtains better results than Word2Vec. A high percentage of the generated words by BERT are synonymous, for example, the percentage is 0.52 for the top 10 words.

We choose the following five words (enchanting, speckled, gorgeous, excused, and ferocious) to show the generated 20 similar words in Table 4. We can see that Word2Vec produced more non-similar words compared with BERT, in which these words are shown in bold. We see that many simple synonyms are not generated by Word2Vec, e.g., "enchanting->beautiful", "speckled->spotted, "gorgeous->wonderful, "ferocious->deadly, etc. We also find that many morphological derivations of one word are most likely to occur in the similar words, e.g., "excuse" and "excuses" for "excused". Conversely, BERT produces more simple and high-similar words, less non-similar words.

4.4 Ablation study of our method

To further analyze the factors affecting UnsupPBMT-BERT, we do more experiments in this section. We analyze the following two parameters: number of sentences searched (Num(sentences)) and number of the final similar words (Num(SimilarWords)). Here, for each experiment, we vary one parameter and fix the other parameter.

(1) How many sentences are needed to be found (Num(sentences))?

We conducted experiments to set the number of sentences searched varying from 1 to 100. Figure 3 shows the results of UnsupPBMT-BERT on three datasets using two metrics.

We can see that the SARI value is very low when Num(sentences) equals 1, and gradually stabilized when Num(sentences) is changed from 10 to 100. The results verify that our strategy by searching for many different sentences can generate high-quality similar words because it brings diversity compared with the strategy by searching for only one sentence.

The FKGL value is stable under different sentences because the metric only measures the readability of the system output text by considering the sentence length and word length, without caring about the quality of the output text.

(2) How many similar words are needed to be produced (Num(SimilarWords))?

By varying the number of the final similar words, we study the effect of Num(SimilarWords) in UnsupPBMT-BERT. We set Num(SimilarWords) varying from 20 to 100. Figure 4 shows the results of UnsupPBMT-BERT.

We can see that the SARI value increases gradually and the FKGL value decreases gradually when Num(SimilarWords) increases, except the SARI value in WikiLarge. Our method can obtain more synonyms when Num(SimilarWords) increases, which brings a better initialization for the phrase tables. In contrast to other results, Our method achieves the best SARI value on WikiLarge dataset when the number of high-similar words equals 20. We found that the bad results are caused by the oversimplification of many complex words when Num(SimilarWords) increases. In the future, we will try to incorporate a control mechanism to alleviate this problem.

(3) Influence of ranking features

To determine the importance of each ranking feature, we make an ablation study by removing one feature in turn. The results are presented in Table 5. UnsupPBMT-BERT combi-

Table 3 The proportion with which the generated words are synonymous by manual evaluation

Method	The number of generated words				
	1−10	1−20	1−30	1−40	1−50
BERT	0.52	0.39	0.33	0.28	0.25
Word2Vec	0.32	0.24	0.20	0.17	0.15

Table 4 The similar words of different words are obtained by BERT and Word2Vec, respectively. Non-similar words and words with inconsistent part-of-speech are shown in bold

Word	Method	The produced similar words
Enchanting	BERT	enchanting, fascinating, beautiful, magical, delightful, charming, exquisite, haunting, attractive, engaging, exotic, elegant, attracting, romantic, **magnificent**, enjoyable, wonderful, appealing, **exceptional**, **accompanying**, splendid.
	Word2Vec	enchanting, captivating, charming, delightful, mesmerizing, **breathtaking**, bewitching, alluring, beguiling, **unforgettable**, **breath-taking**, **idyllic**, wondrous, **marvelous**, **charm**, **verdant**, exquisitely, lovely, quaint, **scenery**
Speckled	BERT	speckled, spotted, freckle, fleck, scattered, **streaked**, **splashed**, dotted, stained, **sprayed**, spot, smeared, splotch, **burl**, **strewn**, gleaming, spotting, banded, **conspicuous**, **dispersed**.
	Word2Vec	speckled, mottled, flecks, specks, **brownish**,**reddish**, **greenish**,**streaked**,**silvery**, **pinkish**, **dusky**, striped, **iridescent**, **whitish**,**bluish**,**yellowish**,**multicolored**, **gray**,**brown**,**blue-green**.
Gorgeous	BERT	gorgeous, beautiful, lovely, wonderful, excellent, **stunning**, handsome, attractive, sexy, graceful, pretty, flawless, magnificent, fine, exquisite, elegant,great, sophisticated, amazing, spectacular.
	Word2Vec	gorgeous, beautiful, lovely, **stunning**, fabulous, amazing, cute, charming, breathtaking, **stunningly**, splendid, chic, **lush**, alluring, **marvelous**, glamorous, **awesome**, classy, sweet, **incredible**.
Excused	BERT	excused, dismissed, omitted, exempt, forgiven, allowed, removed, spared, withdrew, **forbidden**, admitted, explained, **extended**, introduced, isolated, **interrupted**, expelled, **exhausted**, pardoned, authorized.
	Word2Vec	excused, dismissed, forgiven, removed, admitted, explained, **absences**, **obliged**, **absent**, **excuse**, excepted, authorized, isolated, expelled, **politely**, **excuses**, **regretted**, **objection**, **objected**, **punished**.
Ferocious	BERT	ferocious, fierce, ruthless, deadly, furious, formidable, vicious, brutal, savage, aggressive, grim, murderous, lethal, severe, **massive**, monstrous, **territorial**, enormous, **impressive**, violent.
	Word2Vec	ferocious, fierce, **fearsome**, vicious, ferocity, merciless, relentless, **snarling**, bloodthirsty, monstrous, menacing, furious, ravenous, **unstoppable**, terrifying, **fearless**, **fury**, **onslaught**, **unleashed**, **vengeful**.

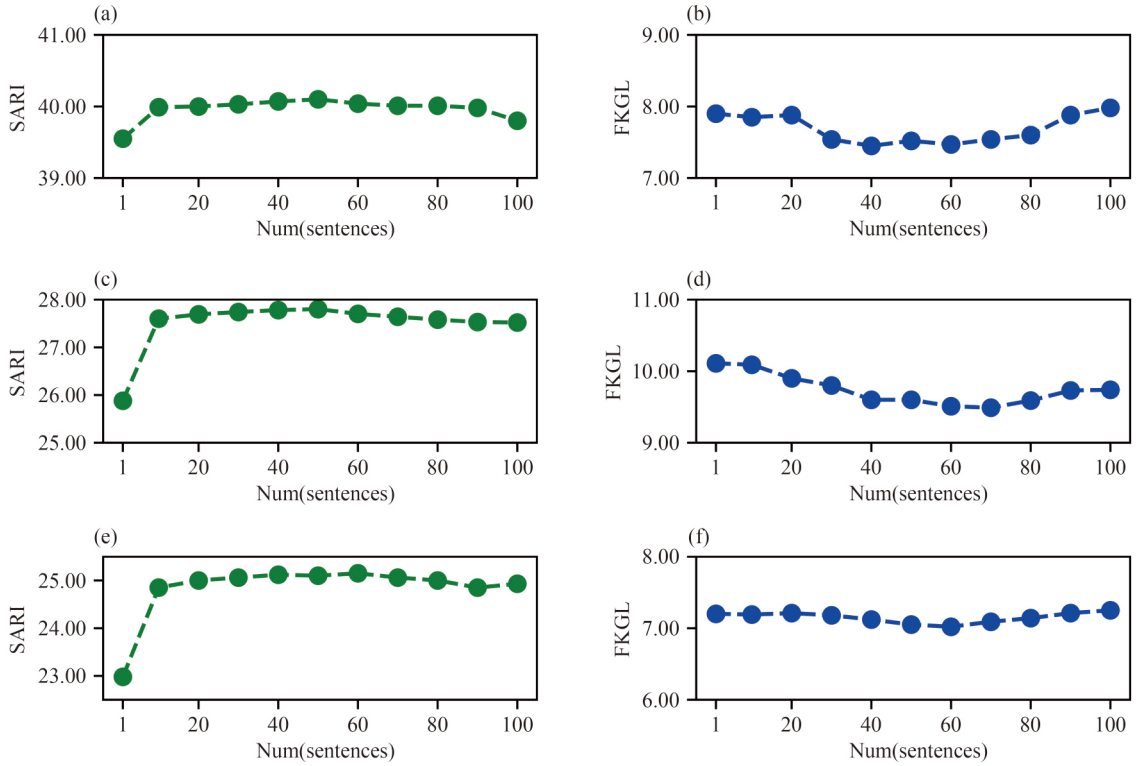


Fig. 3 Evaluation results of different number of sentences on three test sets. (a) WikiLarge; (b) WikiLarge; (c) WikiSmall; (d) WikiSmall; (e) NewSela; (f) NewSela

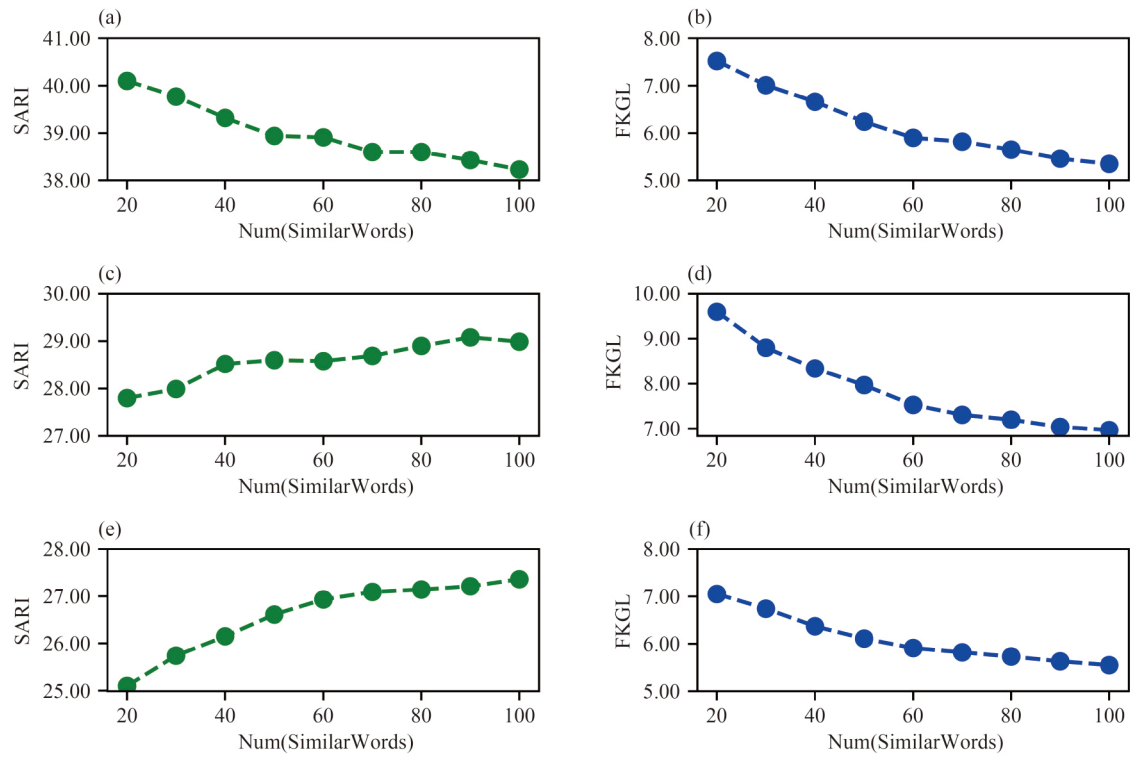


Fig. 4 Evaluation results of different number of SimilarWords on three test sets. (a) WikiLarge; (b) WikiLarge; (c) WikiSmall; (d) WikiSmall; (e) NewSela; (f) NewSela

ning all three features achieves the best SARI value, which means all features have a positive effect. Our method removing "Simlarity" feature achieves better results on FKGL metric, but it decreases the values of SARI, because our method removing "Simlarity" feature prefers to use the words with high frequency. In our experiments, all features in our method are regarded as equally important, which may not be the best option. In the future, we can combine these features

by using different weights to improve the performance of our method.

5 Related work

5.1 Pre-trained language models

Pre-trained language models (LMs) based deep neural networks include GPT [30], XLNet [31], BERT [32] and its variants [33,34], which push the performances of many NLP tasks to a new level. The main reasons have the following two aspects: (1) LMs learn generic linguistic features from large-scale unstructured text corpora, which can serve as a knowledge base. (2) LMs based on Transformer architecture capture high-order, long-range dependency in texts, which have strong feature representation power. In this paper, we utilize LMs as a knowledge base to generate high-similar words.

5.2 Supervised text simplification

Supervised text simplification methods treat text simplification tasks as monolingual machine translation tasks that translating from complex sentences to simplified sentences. Early attempts [3,27] rely on statistical machine translation systems to simplify texts, such as phrase-based machine translation. The trained parallel dataset is WikiSmall [6], which aligns sentences from the "ordinary" English Wikipedia and the "simple" English Wikipedia. Statistical text simplification has the advantages of simple and easy to interpret.

Recently, neural approaches based on the encoder-decoder framework have been developed for supervised text simplification. Zhang et al. [5] incorporated deep reinforcement learning into the encoder-decoder framework. Zhang et al. constructed the WikiLarge dataset by merging multiple small datasets from English Wikipedia and Simple English Wikipedia. Zhao et al. [26] integrated the Simple PPDB (A Paraphrase Database for Simplification) for simplification into the Transformer architecture, where PPDB is a paraphrase knowledge base that contains a wide range of real-world simplification rules. Dong et al. [29] adopted a neural programmer-interpreter approach to learn explicit edit operations (ADD, DELETE and KEEP). Scarton et al. [35] and Martin et al. [1] added artificial tokens to the beginning of each source sentence in the parallel corpus without changing the encoder-decoder architecture. The performance of supervised neural text simplification strongly relies on large amounts of parallel sentences. However, the availability of parallel text simplification corpora is scarce.

5.3 Unsupervised text simplification

To overcome the scarcity of parallel TS corpus, unsupervised TS without using any parallel corpus has attracted much attention. Unsupervised text simplification without using any parallel corpus simplifies the complex sentences to the simplified sentences. Surya et al. [8] proposed an unsupervised neural text simplification based on a shared encoder and two decoders, which only learn the neural network parameters from simple sentences set and complex sentences set. Narayan and Gardent [36] and Kumar et al. [9] are the pipeline-based unsupervised framework, where the pipeline of Narayan and Gardent is composed of lexical simplification, sentence splitting, and phrase deletion, the pipeline of Kumar et al. includes deletion, reordering, and lexical simplification. Martin et al. [37] adopted multilingual sentence embedding modeling LASER [38] to calculate the similarity between the sentences from 1 billion sentences from CCNET [39]. Since the aim of the two works is to find the most similar sentences from a large corpus, they cannot guarantee that the aligned sentences preserve the same meanings.

In [10], we introduced a phrase-based unsupervised text simplification UnsupPBMT, which only uses the ordinary English Wikipedia as a knowledge base without any simple corpus. The method achieves very good results on the widely used benchmarks. The key part of UnsupPBMT initializes the phrase tables without using any parallel corpus. Therefore, the quality of the phrase tables is crucial for the performance of the text simplification method. We found that UnsupPBMT aligns a large number of non-similar words to initialize the phrase tables, which will introduce noise into simplification. In this paper, we build our approach upon pre-trained LMs, which are used as general linguistic knowledge sources for predicting high-similar words.

Table 5 Ablation study results of the ranking features. "w/o" denotes "without"

	SARI	FKGL
UnsupPBMT-BERT	40.10	7.52
w/o BERT	39.33	7.33
w/o Similarity	39.18	5.57
w/o Frequency	39.37	7.52

6 Conclusion

In this paper, we propose an unsupervised Statistical Text Simplification Using Pre-trained Language Modeling BERT for Initialization, called UnsupPBMT-BERT, which can produce more high-quality similar words automatically. Our extensive experimental study shows that UnsupPBMT-BERT can achieve better performance than the other unsupervised text simplification methods on three datasets using two metrics. We show that the context is crucial for producing similar words but has been largely overlooked by the mainstreams of literature. In the future, we will try to construct a paraphrase database like PPDB [40] using the proposed method.

Acknowledgements This research was partially supported by the National Natural Science Foundation of China (Grant Nos. 62076217 and 61906060); and the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China (IRT17R32).

References

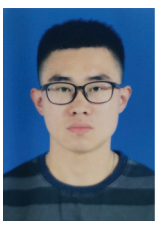
1. Martin L, de la Clergerie É, Sagot B, Bordes A. Controllable sentence simplification. In: Proceedings of the 12th Conference on Language Resources and Evaluation. 2020, 4689−4698
2. Nisioi S, Štajner S, Ponzetto S P, Dinu L P. Exploring neural text simplification models. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017, 85–91
3. Wubben S, van den Bosch A, Krahmer E. Sentence simplification by monolingual machine translation. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers.

2012, 1015−1024
4. Xu W, Napoles C, Pavlick E, Chen Q, Callison-Burch C. Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics, 2016, 4: 401–415
5. Zhang X, Lapata M. Sentence simplification with deep reinforcement learning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017, 584–594
6. Zhu Z, Bernhard D, Gurevych I. A monolingual tree-based translation model for sentence simplification. In: Proceedings of the 23rd International Conference on Computational Linguistics. 2010, 1353−1361
7. Xu W, Callison-Burch C, Napoles C. Problems in current text simplification research: new data can help. Transactions of the Association for Computational Linguistics, 2015, 3: 283–297
8. Surya S, Mishra A, Laha A, Jain P, Sankaranarayanan K. Unsupervised neural text simplification. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019, 2058−2068
9. Kumar D, Mou L, Golab L, Vechtomova O. Iterative edit-based unsupervised sentence simplification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020, 7918−7928
10. Qiang J, Wu X. Unsupervised statistical text simplification. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(4): 1802−1806
11. Meng Y, Zhang Y, Huang J, Xiong C, Ji H, Zhang C, Han J. Text classification using label names only: a language model self-training approach. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020, 9006−9017
12. Petroni F, Rocktäschel T, Lewis P, Bakhtin A, Wu Y, Miller A H, Riedel S. Language models as knowledge bases?. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019, 2463−2473
13. Roberts A, Raffel C, Shazeer N. How much knowledge can you pack into the parameters of a language model?. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020, 5418−5426
14. Zhang H, Khashabi D, Song Y, Roth D. TransOMCS: from linguistic graphs to commonsense knowledge. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. 2020, 4004−4010
15. Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E. Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. 2007, 177−180
16. Artetxe M, Labaka G, Agirre E, Cho K. Unsupervised neural machine translation. In: Proceedings of the 6th International Conference on Learning Representations. 2018
17. Pennington J, Socher R, Manning C. GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014, 1532−1543
18. Farr J N, Jenkins J J, Paterson D G. Simplification of flesch reading ease formula. Journal of Applied Psychology, 1951, 35(5): 333–337
19. Heafield K. KenLM: faster and smaller language model queries. In: Proceedings of the 6th Workshop on Statistical Machine Translation. 2011, 187−197
20. Lample G, Ott M, Conneau A, Denoyer L, Ranzato M. Phrase-based & neural unsupervised machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018, 5039−5049
21. Li D, Zhang Y, Peng H, Chen L, Brockett C, Sun M T, Dolan B. Contextualized perturbation for textual adversarial attack. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021, 5053−5069
22. Glavaš G, Štajner S. Simplifying lexical simplification: do we need simplified corpora?. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. 2015, 63−68
23. Brysbaert M, New B. Moving beyond Kučera and Francis: a critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. Behavior Research Methods, 2009, 41(4): 977–990
24. Qiang J, Li Y, Zhu Y, Yuan Y, Wu X. Lexical simplification with pretrained encoders. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(5): 8649–8656
25. Qiang J, Lv X, Li Y, Yuan Y, Wu X. Chinese lexical simplification. IEEE/ACV Transactions on Audio, Speech, and Language Processing, 2021, 29: 1819–1828
26. Zhao S, Meng R, He D, Andi S, Bambang P. Integrating transformer and paraphrase rules for sentence simplification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018, 3164−3173
27. Narayan S, Gardent C. Hybrid simplification using deep semantics and machine translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. 2014, 435−445
28. Guo H, Pasunuru R, Bansal M. Dynamic multi-level multi-task learning for sentence simplification. In: Proceedings of the 27th International Conference on Computational Linguistics. 2018, 462−476
29. Dong Y, Li Z, Rezagholizadeh M, Cheung J C K. EditNTS: an neural programmer-interpreter model for sentence simplification through explicit editing. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019, 3393−3402
30. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI Blog, 2019, 1(8): 9
31. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le Q V. XLNet: generalized autoregressive pretraining for language understanding. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). 2019, 5754−5764
32. Devlin J, Chang M W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1. 2019, 4171−4186
33. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT: a lite BERT for self-supervised learning of language representations. In: Proceedings of the 8th International Conference on Learning Representations. 2020
34. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2019, 7871−7880
35. Scarton C, Specia L. Learning simplifications for specific target audiences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. 2018, 712−718
36. Narayan S, Gardent C. Unsupervised sentence simplification using deep semantics. In: Proceedings of the 9th International Natural Language Generation Conference. 2015, 111−120
37. Martin L, Fan A, de la Clergerie É, Bordes A, Sagot B. MUSS: multilingual unsupervised sentence simplification by mining paraphrases. 2021, arXiv preprint arXiv: 2005.00352
38. Artetxe M, Labaka G, Agirre E. Unsupervised statistical machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018, 3632−3642
39. Wenzek G, Lachaux M A, Conneau A, Chaudhary V, Guzmán F, Joulin

A, Grave E. CCNET: extracting high quality monolingual datasets from web crawl data. In: Proceedings of the 12th Language Resources and Evaluation Conference. 2020, 4003−4012

40. Pavlick E, Callison-Burch C. Simple PPDB: a paraphrase database for simplification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016, 143−148

Jipeng Qiang is currently an associate professor in the School of Information Engineering, at Yangzhou University, China. He received his PhD degree in computer science and technology from Hefei University of Technology, China in 2016. He was a PhD visiting student in the Artificial Intelligence Lab at the University of Massachusetts Boston, USA from 2014 to 2016. He has published more than 50 papers, including AAAI, EMNLP, TKDE, TASLP, and TKDD. His research interests mainly include natural language processing and data mining.



Feng Zhang is currently working toward the MS degree of computer science at the Yangzhou University, China. He received his BS degree in computer science from Huaiyin Institute of Technology, China. His research interest is text simplification.

Yun Li is currently a professor in the School of Information Engineering, Yangzhou University, China. He received the MS degree in computer science and technology from Hefei University of Technology, China in 1991, and the PhD degree in control theory and control engineering from Shanghai University, China in 2005. He has published more than 100 scientific papers. His research interests include data mining and cloud computing.

Yunhao Yuan is currently an associate professor in the School of Information Engineering, Yangzhou University, China. He received the MEng degree in computer science and technology from Yangzhou University, China, in 2009, and the PhD degree in pattern recognition and intelligence system from Nanjing University of Science and Technology, China in 2013. His research interests include pattern recognition, data mining, and image processing.

Yi Zhu is currently an assistant professor in the School of Information Engineering, Yangzhou University, China. He received the BS degree from Anhui University, the MS degree from the University of Science and Technology of China, and the PhD degree from Hefei University of Technology, China. His research interests are in data mining and knowledge engineering. His research interests include data mining and recommendation systems.

Xindong Wu is a professor in the School of Computer Science and Information Engineering at the Hefei University of Technology, China, and the president of Mininglamp Academy of Sciences, Mininglamp, China, and a fellow of IEEE and AAAS. He received his BS and MS degrees in computer science from the Hefei University of Technology, China, and his PhD degree in artificial intelligence from the University of Edinburgh, UK. His research interests include data mining, big data analytics, and knowledge-based systems.