Chinese Lexical Simplification

Jipeng Qiang, Xinyu Lu, Yun Li, Yunhao Yuan, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Lexical simplification has attracted much attention in many languages, which is the process of replacing complex words in a given sentence with simpler alternatives of equivalent meaning. Although the richness of vocabulary in Chinese makes the text very difficult to read for children and non-native speakers, there is no research work for the Chinese lexical simplification (CLS) task. To circumvent difficulties in acquiring annotations, we manually create the first benchmark dataset for CLS, which can be used for evaluating the lexical simplification systems automatically. To acquire a more thorough comparison, we present five different types of methods as baselines to generate substitute candidates for the complex word that includes synonym-based approach, word embedding-based approach, BERT-based approach, sememe-based approach, and a hybrid approach. Finally, we design the experimental evaluation of these baselines and discuss their advantages and disadvantages. To our best knowledge, this is the first study for CLS task.**

***Index Terms*—Lexical simplification, BERT, unsupervised, pretrained language model.**

I. Introduction

LEXICAL Simplification (LS) aims at replacing complex words with simpler alternatives without changing the meaning of the sentence, which can help various groups of people, including children [1], non-native speakers [2], people with cognitive disabilities [3], to understand text better. For example, the sentence "John composed these verses in 1995" could be lexically simplified into "John wrote the poems in 1995". LS task has been applied to different languages, such as English [2], [4]–[8], Japanese [9], [10], Spanish [11], [12], Swedish [13] and Portuguese [14].

Manuscript received September 10, 2020; revised December 22, 2020 and March 29, 2021; accepted April 30, 2021. Date of publication May 24, 2021; date of current version June 4, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 62076217, 91746209, and 61703362, by the National Key Research and Development Program of China under Grant 2016YFB1000900, by the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under Grant IRT17R32, and by the Natural Science Foundation of Jiangsu Province of China under Grant BK20170513. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zoraida Callejas. *(Jipeng Qiang and Xinyu Lu contributed equally to this work.)(Corresponding author: Yun Li.)*

Jipeng Qiang, Xinyu Lu, Yun Li, and Yunhao Yuan are with the Department of Computer Science, Yangzhou 225009, China (e-mail: qjp2100@gmail.com; 181303216@yzu.edu.cn; liyun@yzu.edu.cn; yhyuan@yzu.edu.cn).

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei University of Technology, Hefei 230009, China, and also with the Mininglamp Academy of Sciences, Mininglamp Technology, Beijing 100084, China (e-mail: xwu@hfut.edu.cn).

Digital Object Identifier 10.1109/TASLP.2021.3078361

Chinese, the only existing pictographic language in the modern world, is one of the most difficult languages to learn [15], [16]. There are more than 200 000 commonly used words in Chinese that are composed of 5000 characters. For example, for a simple Chinese word "Qizi" (Wife), there are dozens of equivalent meaning, such as "Lǎopo," "Póniáng," "Xífù," "Nèirén," "Háitaniáng," "Duìxiàng," "Furén," "Àiren," "Tàitai" and so on. The complexity and richness of words in Chinese text tend to make these people (children, non-native speakers, etc) feel extremely difficult. These suggest that Chinese lexical simplification system is an invaluable tool for improving text accessibility. However, there has been no published work on Chinese lexical simplification so far. Therefore, we focus on the Chinese lexical simplification (CLS) problem in this paper.

The first challenge of CLS is the lack of human annotation. We first construct a benchmark dataset HanLS for CLS that can be used for both training and evaluation, as well as to accelerate the research on this topic. Firstly, we request two native speakers with teaching experience to give some target words as the list of content words (nouns, verbs, adjectives, and adverbs), and search some sentences containing the target words. Given a sentence and a word to be simplified, we then asked six annotators to give its simpler variants of that word that are appropriate in the context of the sentence.

On the English lexical simplification task, the substitutes are composed of merely one word in most cases. The second challenge in CLS task is the substitutes are made up of different number of characters. There have been no published approaches on CLS so far. For providing a comprehensive comparison, we propose five different types of methods as baselines to generate substitutes. (1) Synonym dictionary-based approach: it obtains substitute candidates by picking synonyms from a manually curated lexical dictionary. (2) Word embedding-based approach: it uses the similarity of word embeddings to generate substitute words. (3) Pretrained language model-based approach: we adopt pre-trained language model BERT [17] that masks the complex word of the original sentence for feeding into BERT to predict the masked token. (4) Sememe-based approach: we design a word substitution method based on sememes, the minimum semantic units, which can retain more potential valid substitutes for complex words. (5) One hybrid method: we extract candidate substitutions by combining the synonym dictionary and the pretrained language model-based approach. After obtaining the substitute candidates, we utilize the following four features to select the best substitute: language modeling based on BERT, word frequency, word similarity, and Hownet similarity, which respectively capture one aspect of the suitability of the candidate word to replace the complex word.

The contributions of this work are three-fold:

1) We focus on the Chinese lexical simplification (CLS) task and create manually the first benchmark dataset HanLS for CLS that can be used to evaluate the CLS approaches automatically.

2) We propose five different benchmarks for the CLS task, which contains two classic methods (Synonym dictionary and Word embedding) and three latest methods (BERT, Sememe, and Hybrid).

3) Experimental results show that these baselines (Synonym dictionary, Pretrained language model, and Hybrid) output lexical simplifications that are grammatically correct and semantically appropriate on HanLS.

The dataset and baselines to accelerate the research on this topic are available at https://www.github.com/luxinyu1/ChineseLS.

II. Related Work

Lexical simplification (LS) as a sub-task of text simplification focuses to simplify complex words of one sentence with simpler variants. Most current researches are focused on English lexical simplification. We will introduce English LS methods in detail, briefly explain other language LS methods, and finally present some work related to Chinese LS. Besides, we will present the common datasets for each language LS task. All these datasets contain instances that are composed of a sentence, a target complex word, and a set of suitable substitutions provided by humans for their simplicity.

English LS and its benchmarks: The popular lexical simplification approaches were rule-based, in which each rule contains a complex word and its simple synonyms [18], [19]. Rule-based systems usually identified synonyms from WordNet or other linguistic databases for a predefined set of complex words and selected the "simplest" from these synonyms based on the frequency of word or length of word [1], [20]. Some LS systems tried to extract rules from parallel corpora [21]–[23]. In an effort to avoid the requirement of lexical resources or parallel corpora, LS systems based on word embeddings were proposed [5]–[7], [24]. They extracted the top words as candidate substitutions whose vectors are closer in terms of cosine similarity with the complex word. Pre-training language models [17], [25] have attracted wide attention and have shown to be effective for improving many downstream natural language processing tasks. The recent LS methods are based on BERT [8], [26] to generate suitable simplifications for complex words.

There are three widely used datasets for English LS, which are LexMTurk [23], BenchLS [2] and NNSeval [27]. LexMTurk is composed of 500 instances annotated by 50 Amazon Mechanical "turkers". BenchLS is composed of 929 instances for English, which is from LexMTurk and LSeval [1]. The LSeval contains 429 instances, in which each complex word was annotated by 46 turkers and 9 Ph.D. students. NNSeval is composed of 239 instances for English, which is a filtered version of BenchLS.

Other language LS: Most of the other language LS methods are often based on linguistic databases to find simpler candidate substitutes for complex words. The PorSimples project provides a LS method for Brazilian Portuguese, which uses sets of related words provided by the databases Tep 2.0 and PAPEL [14]. Bott *et al.* [11] use the Spanish OpenTheaurus to find synonyms for complex words in Spanish. Keskisrkk [28] used a thesaurus SynLex for the Swedish language to find synonyms for complex words. Kajiwara *et al.* [9] taken advantage of dictionaries that provide word descriptions. The method extracts candidate substitutions from a complex word's definition. They constructed a dataset from the newswire corpus for the evaluation of Japanese lexical simplification. Afterward, Kodaira *et al.* [29] proposed a new controlled and balanced dataset for Japanese lexical simplification with a high correlation with human judgment.

Chinese LS: To our best knowledge, there is no work about Chinese LS. The most relevant work with Chinese LS is Chinese text readability assessment [30]. Text readability assessment is used to measure the difficulty level of the given text to assist the selection of suitable reading materials for learners [31]. Automatic text readability measures are composed of formula-based methods and classification methods using various features, including word features, sentence features, etc. When the difficulty level of the text is obtained, the next step is to simplify the original text for reducing the difficulty of the text and meeting the needs of different users. However, Chinese LS task receives little attention, and we cannot obtain publicly available methods and datasets. Therefore, in this paper, we will first construct a Chinese LS dataset for evaluation and propose some different LS systems to simplify Chinese sentence.

Other related tasks: Many tasks in NLP need to generate the substitutes for one or several words in one sentence. We will analyze their similarities and differences below.

Grammatical Error Correction (GEC) [32] is typically formulated as a sentence correction task. A GEC system takes a potentially erroneous sentence as input and is expected to transform it to its corrected version. Most recent GEC systems are based on the seq2seq framework and are trained with error-corrected sentence pairs. Due to massive training data, the state-of-the-art GEC system can achieve human-level performance in GEC benchmarks and be practically used for correcting grammatical errors. There is no demand for GEC that the substitutes for the original word must be simpler alternatives. LS is commonly treated as an unsupervised task, as no labeled training dataset is in English or other language LS tasks. The first step of LS is to identify the specific words in a given sentence that should be simplified, and GEC does not need to perform this step. Compared with human-level performance, the state-of-the-art LS systems have very big upgrade space.

In recent years, Adversarial Text Generation (ATG) [33] for natural language processing (NLP) tasks has attracted much attention, whose aims to perturb input text to trigger errors in machine learning models, while keeping the output close to the original. The task first needs to find the vulnerable words in one given input sentence for the target model, then generate substitutes for the vulnerable words. Both LS and ATG need to identify specific words, e.g., complex words for LS and vulnerable words for ATG. After replacing the specific words with the substitutes, both tasks need to guarantee the fluency and semantically preservation in the generated adversarial samples. But, compared with the original words, the substitutes in LS are simpler alternatives of equivalent meaning, and the substitutes in LS can only mislead the target model.

III. A Dataset

After referring to the construction of existing English and Japanese lexical simplification datasets, we create a dataset HanLS for Chinese lexical simplification task annotated by three undergraduates and three graduate students. These students are all native Chinese speakers. As all annotators are all native Chinese speakers, in which two annotators have teach experience for children. Therefore, the lexical simplification dataset is more suitable for evaluating the performance of lexical simplification methods designing for the children. We follow these steps below.

1) **Extracting sentences**: We define complex words as "High Level" words in the worldwide popular Chinese HSK vocabulary [34]. The 600 high-level words (nouns, verbs, adjectives, and adverbs) are chosen by two native speakers with teaching experience based on their experience and intuition. We aim to create a balanced corpus and control sentences that have only one complex word. Then, sentences that include a complex word are randomly extracted from these two sources: Modern Chinese corpus of the State Language Commission and Chinese translation corpus.[1] Following previous work, 10 sentences including each complex word are collected. Annotators chose one sentence for each complex word under each POS tag by controlling the number of complex words in each sentence.

2) **Providing substitutes**: Simplification candidates were collected from five native speakers. For each instance, the annotators wrote substitutes that did not change the sense of the sentence. When providing a substitute, an annotator could refer to a dictionary but was not supposed to ask the other annotators for an opinion. When an annotator could not think of a paraphrase, they were permitted to supply no entry. These annotators ranked the various substitutes provided for the complex word according to how simple they were in contexts.

3) **Merging All Annotations**: All annotations were merged into one dataset by averaging the annotations from all annotators. An example from this dataset is explained below. Given one example, we suppose it has one substitute x. When the following rankings (1,2,2,4,1) were obtained from five annotators, the average rank of x was 2. The final integrated ranking for each instance is obtained by rearranging the average ranks of these substitutes in ascending order.

The merged dataset was evaluated by a new annotator. The annotator rated a substitute as inappropriate based on the following two criteria: i) A substitute is inappropriate if the sentence becomes unnatural after replacing the target word; ii) A substitute is inappropriate if the meaning of the sentence is changed after replacing the target word. Finally, the dataset has 524 instances where each instance has an average of 8.51 substitutes, denoted as HanLS. The complex words in HanLS contain nouns 166, verbs 160, adjectives 134, and adverbs 64, which are composed of one character 9, two characters 472, three characters 13, and four characters 30, respectively. Figure 1 shows an example of the dataset. Here, the complex word has 9 substitutes and we only show four of them.

[1][Online]. Available: https://github.com/brightmart/nlp_chinese_corpus

Sentence: 句子太长, 念起来拗口.
(The sentence is too long, it does **not** read **sleekly**.)
Substitute list: 别扭, 不顺口, 难, 不通顺
(hardly, not smoothly, difficultly, not fluently)

Fig. 1. An example of annotation in the dataset HanLS. The word with red color is the complex word.

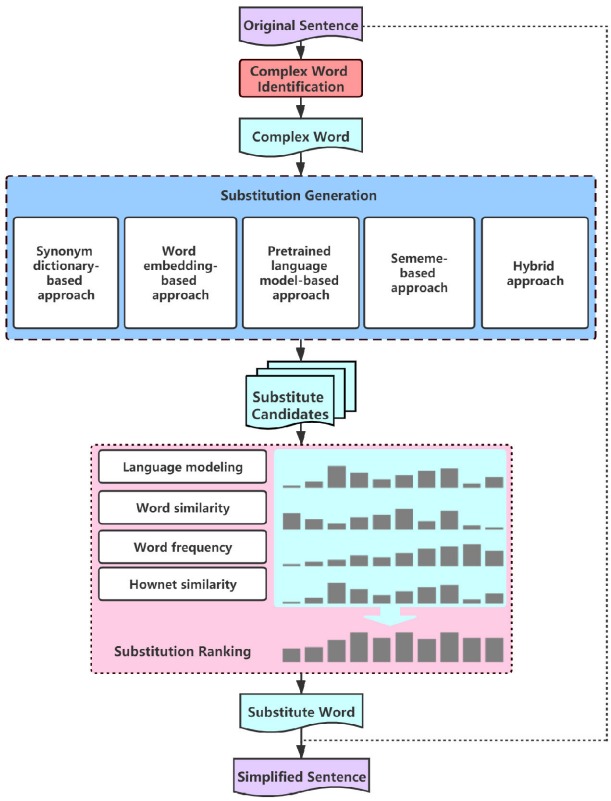

Fig. 2. Chinese lexical simplification framework.

IV. Approaches

Following the steps of English lexical simplification [5], [27], Chinese lexical simplification system also includes the following three steps: complex word identification, substitution generation, and substitution ranking. In the complex word identification (CWI) step, the goal is to select the words in a given sentence that should be simplified. The aim of Substitution Generation (SG) is to produce substitute candidates for complex words. We present five different methods for SG. Giving substitute candidates of the complex word, the Substitution Ranking (SR) of the lexical simplification is to decide which one of the candidate substitutions that fits the context of the complex word is the simplest. We adopt four high-quality features to rank the substitutes. The structure of our framework is shown in Figure 2.

A. Complex Word Identification

Complex Word Identification (CWI) is used to identify the complex words from a given input sentence. We provide two unsupervised methods to identify the complex words from a single Chinese sentence.

The first is grounded on the HSK(Chinese Proficiency Test) graded vocabulary sheet,[2] and we consider the word in the top-level(HSK-6) or not recorded as complex words.

The second method is based on word frequency, which is counted from a large and well-rounded corpus. We set a threshold and if the frequency of a word is lower than the threshold, we consider it as a complex word.

B. Substitution Generation (SG)

An ideal SG strategy will be able to find all words that can replace a given target complex word in all contexts in which it may appear. For providing a comprehensive comparison, we provide five different types of approaches to generate substitutes for Chinese LS task and discuss their advantages and disadvantages. For the following substitution generation methods, in our experiments, we filter these substitutes that are not in the dictionary (Modern Chinese Word List).

1) Three Baselines: **(1) Synonym dictionary-based approach**: Most LS approaches [11], [14] utilized synonym dictionary for SG, e.g., WordNet for English and OpenThesaurus for Spanish. For Chinese SG, we choose a synonym thesaurus HIT-Cilin [35] for generating substitutes, which contains 77 371 distinct words. The advantage of the method is simple and easy to implement. Besides that constructing a synonym dictionary is expensive and time-consuming, it is impossible to cover all the words.

2) Word embedding-based approach: Word embedding-based approaches [2] was used for English SG, which first obtains the vector representation for each word from the pretrained word embedding model and extracts the top k words as substitutes whose embeddings vector has the highest cosine similarity with the vector of the complex word. Here, we use the pretrained Chinese word vectors[3] using Word2Vector algorithm [36], and extract the top 10 words as substitutes. The advantage of the method is the pretrained word embedding model is easily accessible because it only needs an ordinary large amount of text corpus. The substitute candidates contain not only similar words, but also highly related words and words with opposite meanings.

3) Sememe-based approach: The meaning of a word can be represented by the composition of its sememes, where sememe is defined as the minimum indivisible semantic unit of human languages defined by linguists [37]. Sememes have been successfully used for many NLP tasks including semantic composition [38], pretrained language model [39], etc. This is the first attempt to apply sememe for lexical simplification.

In practical NLP applications, Sememe knowledge bases are built based on sememes, in which Hownet[4] is the most famous one [40]. In contrast to WordNet focusing on the relations between senses, it annotates each word with one or more relevant sememes. We first introduce how words, senses and sememes are organized in HowNet. In HowNet, a word

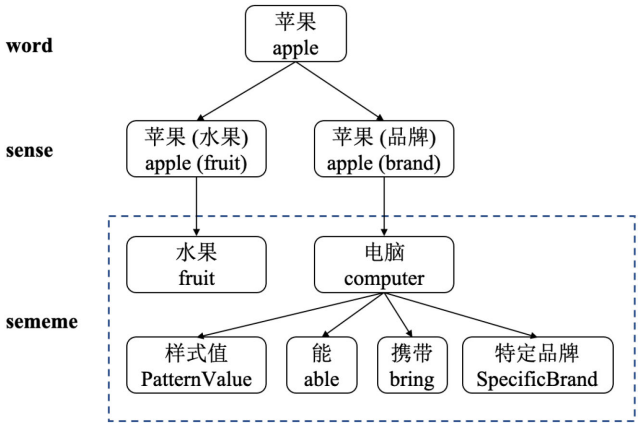

Fig. 3. An example of HowNet.

may have various senses, and each sense has several sememes describing the exact meaning of sense. As illustrated in Figure 3, the word $apple$ has two senses including $apple(fruit)$ and $apple(brand)$ in HowNet. The sense $apple(fruit)$ only has one sememe $fruit$, and the sense $apple(brand)$ has five sememes including $computer$, "$PatternValue$," "$able$," "$bring$" and "$SpecificBrand$". There are about 2000 sememes and over 100 thousand labeled Chinese and English words in HowNet.

In Hownet, the sememes of a word can accurately describe the meaning of the word. Therefore, the words owning the same sememe annotations should share the same meanings, and they can act as substitute candidates for each other. In our sememe-based method, a word w can be substituted by another word $w*$ only if one sense of w has the same sememe annotations as one sense of $w*$.

Compared with the word embedding and language model-based substitution methods, sememe-based approach cannot generate antonyms words as substitutes, although antonyms words could be high similar words. Compared with the synonym-based method, sememe-based method generates more substitute words.

2) BERT-Based Method: Recent English LS method [8], [26] adopted pretrained language model BERT to produce substitutes. BERT is a bi-directional language model trained by two training objectives: masked language modeling (MLM) and next sentence prediction (NSP). Unlike a traditional language modeling objective of predicting the next word in a sequence given the history, MLM predicts missing tokens in a sequence given its left and right context. In contrast to English LS task, we cannot directly utilize Chinese pretrained BERT model for Chinese SG. Because English has a natural space as a separator, we only mask the word w of the sentence S using one special symbol "[MASK]" to obtain the probability distribution of the vocabulary corresponding to the mask word.

In Chinese, a word is composed of one or more characters. For one complex word is composed of four characters, the possible substitutes may be one character, two characters, three characters, and four characters. We need to use different numbers of [MASK] symbol to replace the complex word. Therefore, predicting the [MASK] symbols is not only the target complex word predictiona (cloze task) but also a generating task.

[2][Online]. Available: http://www.chinesetest.cn/userfiles/file/HSK/HSK-2012.xls

[3][Online]. Available: https://github.com/Embedding/Chinese-Word-Vectors

[4][Online]. Available: http://www.keenage.com/

Specifically, we use less than or equal to the number of [MASK] symbols to replace the original word, and combine all the results as substitutes. The original sentence S replaced with [MASK] symbols is denoted as S'. Based on BERT is adept at dealing with sentence pairs, we feed the sentence pair $\{S, S'\}$ into the BERT $\mathcal{M}$ to obtain output prediction $\mathcal{P} = \mathcal{M}(S, S')$. If S' contains many [MASK] symbols, we aim to generate many semantic-consistent substitutes by utilizing the corresponding prediction for each [MASK] symbol. Therefore, we propose two different strategies: full-permutation of top-K predictions and local beam search.

1) Full-Permutation of top-K predictions: Given the [MASK] symbols $\{\text{MASK}_1, \ldots, \text{MASK}_t\}$, we list all possible combinations from the prediction $P^{t\times k}$ of $\mathcal{P}$, which is k^t character combinations. We use the perplexity of all combinations to get top-K combinations, in which these combinations that are not a natural word are filtered out. We filter these words that are not in the Modern Chinese Word List [41]. The advantage of this strategy does not need to additionally adopt BERT.

2) Local Beam Search: All new characters are simultaneously generated based on all prediction for the first strategy. Due to the nature of BERT, this strategy suffers from a conditional independence problem in which the predicted characters are conditional-independently generated and are agnostic of each other. This can result in generating repeating or inconsistent new characters at each generation round.

To address this weak-dependency issue, a native approach to perform beam search would be to maintain a priority queue of top k candidate character series predictions when moving from the leftmost slot to the rightmost slot. This is followed by a ranking step to select the top k most likely series among the Vk series to grow. Based on such a native approach is expensive, as the runtime complexity takes $\mathcal{O}(t * k * V)$. We design a customized beam search method for our model, called Local Beam Search (IBE). Prediction in IBE is limited to the top k character candidates, and thus the beam search procedure as described is applied on the narrow band of K instead of the full vocabulary V. Every time we make a search, we find the optimal k characters from the k^2 paths, not the $k * V$ paths in beam search. This reduces the computation to $\mathcal{O}(t * k^2)$.

3) Hybrid Method: We design a simple hybrid approach for Chinese SG, which combines the synonym dictionary-based approach and the pretrained language model-based approach. Specifically, if the complex word is included in HIT-Cilin synonym dictionary, we use the synonym dictionary-based approach to generate substitutes, else we use the pretrained language modelbased approach.

C. Substitution Ranking (SR)

Giving substitute candidates $C = c_1, c_2, \ldots, c_n$, we choose four different features (word frequency, word similarity, language model and HowNet similarity) to rank these substitutes, where n is the number of substitute candidates. Each of the features captures one aspect of the suitability of the candidate word to replace the complex word. We compute four different rankings (r_{wf}, r_{ws}, r_{lm} and $rank_{hs}$) according to their scores for all substitutes, respectively. The final ranking for all substitutes is computed as follows,

$$f\_r = \lambda_1 r_{wf} + \lambda_2 r_{ws} + \lambda_3 r_{lm} + \lambda_4 r_{hs} \tag{1}$$

where f_r denotes the final rankings of C, the weights $\lambda_1, \lambda_2, \lambda_3$, and λ_4 balance the relative importance of the different features, and $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$.

Because the complex word also could be one of the substitutes, we choose the top two substitutes with the average rank scores over all features. When deciding whether simplifying a word is necessary, we account for this implicitly by performing the simplification only if the substitute has lower information content than the complex word. Specifically, if the first substitute is not the complex word w, we will replace the complex word w with the first substitute. Otherwise, if the first substitute is the complex word w, we will choose the second substitute only if the second substitute has a higher frequency than the complex word.

1) Language modeling: The feature aims to evaluate the fluency of substitute in a given sentence. We do not choose traditional n-gram language modeling, and we choose the pretrained language model BERT to compute the probability of a sentence or sequence of words.

Let $W = w_{-m}, \ldots, w_{-1}, w, w_1, \ldots, w_m$ be the context of the original word w. We adopt a new strategy to compute the likelihood of W. We first replace the original word w with the substitution candidate c, and composed of one new sequence $W' = w_{-m}, \ldots, w_{-1}, c, w_1, \ldots, w_m$.

We then mask one word w_i of W' from front to back and feed into BERT to compute the cross-entropy loss of the mask word.

$$loss(w_i) = -\sum_{i=1}^{V} \mathbb{I}\{y_i = w_i\} \times \log p_{BERT}(y_i = w_i | W'_{\setminus w_i}) \tag{2}$$

where V is the vocabulary, $\mathbb{I}\{\cdot\}$ is the indicator function, and p_{BERT} is the BERT output distribution (conditioned on the W' excluding word w_i).

The language loss of the sequence W' is the average of all words,

$$loss(W') = \sum_{i=-m}^{i=m} loss(w_i)/len(W') \tag{3}$$

where $len(W')$ is the number of charactets in W'

Finally, we rank all substitute candidates based on the corresponding sequence loss $loss(W')$. The lower the loss, the better substitute is the candidate for the original word. We use as context a symmetric window of size five around the complex word.

2) Word similarity: We obtain the vector representation of each word using the pretrained word embedding model, and compute the similarity between the complex word and each substitute. The higher the similarity value, the higher the ranking.

3) Word Frequency: Frequency-based substitute ranking strategy is one of the most popular choices by English lexical simplification. In general, the more frequency a word is used, the most familiar it is to readers. In this work, we adopt the

word frequency which is calculated from one big corpus[5] which contains more than 2.5 hundred million characters. The higher the frequency value, the higher the ranking. We test many word frequency files from different corpora, and this one we adopted is proved to be the best one.

4) Hownet similarity: In addition to the word similarity using word embeddings, we choose a new word similarity method based on Hownet, which has been proved that it has a good performance in antonym and synonym similarity calculation for Chinese words [42]. Hownet-based similarity based on the sememes computes the similarity between the complex word and the substitutes, which provides a good complimentary for the following situation. When the substitute candidates are antonyms and semantically related but not similar words, the two features (language model and word similarity) probably lose their effectiveness. If the similarity value between the candidate and the complex word is greater, then the candidate will have a higher ranking.

V. Experiments

We design experiments to answer the following three questions:

Q1. The quality of the created Chinese lexical dataset HanLS: Is the results of manual evaluation consistent with that of annotated dataset HanLS?

Q2. The difference of the proposed five substitution generation methods: The evaluation metrics from previous English LS task are used to verify the effectiveness of these different SG methods on HanLS.

Q3. The factors of affecting the CLS system: We conduct experiments on HanLS to verify the influence of some key parameters (substitution generation methods and substitution ranking features) on the whole CLS system.

Here, the proposed CLS methods are called as synonym dictionary-based method (**Synonym**), word embedding-based approach (**Embedding**), pretrained language model-based approach (**BERT**), sememe-based approach (**Sememe**) and a hybrid approach (**Hybrid**). BERT based approach has two different strategies for substitution generation: full-permutation of top-K predictions and local beam search. They are denoted as BERT_TopK and BERT_BS, where BERT_BS is the default setting of BERT. In all experiments, we use BERT-Base, Chinese pretrained model.[6]

In lexical simplification task, complex word identification module is often regarded as an independent task to evaluation. We do not evaluate the two proposed complex word identification methods. We choose the first CWI method as the default setting in the open-source Chinese LS system.

A. Evaluation of the Quality of the Dataset HanLS

Considering the richness of Chinese vocabulary, it is very difficult to give all reasonable substitutes for each complex word in HanLS although we find many people to label this dataset.

[5][Online]. Available: https://github.com/liangqi/chinese-frequency-word-list

[6][Online]. Available: https://huggingface.co/bert-base-chinese

TABLE I
The Comparative Results of Manual Evaluation and Automatic Evaluation for All Methods (Embedding, Sememe, BERT, Hybrid and Synonym) on HanLS

	Embedding	Sememe	BERT	Hybrid	Synonym
Num_Cha	472	442	503	470	379
Acc_Manual	0.708	0.799	0.827	0.864	0.917
Acc_Gold	0.623	0.692	0.716	0.785	0.854

In this experiment, we will verify the comprehensiveness of the annotated reasonable substitutes in HanLS to determine the quality of this dataset. Each lexical simplification method can generate one substitute for each complex word in each instance of HanLS, where the substitute can be the complex word itself or other words. We can judge whether the generating substitute is correct or not by manual evaluation or the annotated substitutes in the HanLS. If the evaluation results between manual evaluation and the annotated substitutes in the HanLS are substantially in agreement, we can assume that HanLS is one high-quality dataset for Chinese lexical simplification. Additionally, it should be noted that we only consider these instances in which the complex word is changed by the system, rather than all instances in HanLS, because we cannot evaluate the annotated substitutes for these instances with no replacement.

The total number of instances in HanLS is 524. We adopt the following three metrics.

Num_Cha: The number of instances in which the replacement for the complex word by the lexical simplification system is not the complex word itself.

Acc_Manual: The proportion of instances in which the replacement for the complex word is not the original word and correct by manual evaluation.

Acc_Gold: The proportion with which the replacement of the original word is not the original word and is in the gold standard. The ACC_Gold value is automatically calculated without human intervention.

The results are shown in Table I. From the ranking order of these five methods, we can see that the results of the manual evaluation are in accordance with the results of automatic evaluation. The average proportion of instances in which the results of the manual evaluation is the same as the results of the automatic evaluation is above 85%. Synonym achieves the best values using Manual and Auto. But it only generates the substitutes for 379 instances, which also means that many complex words are replaced by the original word itself. We conclude that HanLS is a high-quality dataset in which the annotated substitutes are reasonable and comprehensiveness. Below, we will give a detailed comparison of the baselines we proposed using HanLS.

B. Evaluation of Substitution Generation

We use the following four metrics from the previous English LS task [8], [27] to evaluate the performance of the SG method. Suppose that there are m samples in test set, where the complex word of the i-th sample is w_i, the set of the annotated substitutes for w_i is o_i, and the set of the generated substitute candidates is q_i. Here, we use $\#(o_i)$ and $\#(q_i)$ are denoted as the number of words in o_i and q_i, respectively.

TABLE II
SUBSTITUTION GENERATION EVALUATION RESULTS. THE NUMBER WITHIN THE PARENTHESIS IN FIRST LINE PRERESENTS THE TOTOAL NUMBER OF THE INSTANCES OR ALL GENERATED CANDIDATES

SG methods	Potential (524)	Precision (4460)	Recall (4460)	F1
Synonym	81.49%	40.68%	**27.42%**	**32.76%**
Embedding	72.14%	19.70%	35.36%	25.30%
Sememe	72.14%	30.76%	13.24%	18.51%
BERT_TopK	80.15%	22.08%	19.70%	20.82%
BERT_BS	88.93%	31.41%	26.23%	28.59%
Hybrid	**90.46%**	**42.90%**	26.40%	32.69%

Potential: The proportion of instances for which at least one of the substitutes generated is in the gold-standard.

$$Precision = \frac{\sum_{i=1}^{m} \mathbb{I}\{any(o_i) \in q_i\}}{m} \tag{4}$$

Here, $\mathbb{I}$ is the indicator function, if at least one of the substitutes generated o_i is in the annotated substitutes q_i, $\mathbb{I}\{any(o_i) \in q_i\}$ is set to 1, else 0.

Precision: The proportion of generated substitute candidates that are in the annotated substitutes.

$$Precision = \frac{\sum_{i=1}^{m} \#(o_i \cap q_i)}{\sum_{i=1}^{m} \#q_i} \tag{5}$$

Recall: The proportion of annotated substitutes that are included in the generated substitution candidates.

$$Recall = \frac{\sum_{i=1}^{m} \#(o_i \cap q_i)}{\sum_{i=1}^{m} \#o_i} \tag{6}$$

F1: The harmonic mean between Precision and Recall.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

The results are shown in Table II. We can see that the two methods (Synonym and BERT) are more effective than the two methods (Embeddings and Sememe). Embedding has the lowest Precision value because the generated substitutes contain many semantically related but not similar words. For sememe-based method, it generates dozens or even hundreds of substitutes for many instances, which results in the poorest Recall value. Synonym-based method is a simple but powerful method, which can be easily understood and deployed to different languages. But both Synonym and Sememe have a big limitation that is their coverage. For example, we can find that many common used words do not occur in this dictionary, e.g., "yuánzhù(Assistance)," "xíngnáng(Luggage)" and "kepò(break up)" for Synonym dictionary, "xiǎnyǒu(rare)," "chúnshǔ(purely)" and "huangmán(wild)" for Sememe. BERT-based method without relying on linguistic databases offers impressive results, mainly because it considers the context of the complex word when generating substitute candidates. BERT_BS outperforms BERT_TopK, which verifies that local beam search strategy can obtain better substitutes than full-permutation of top-K predictions strategy. The hybrid method offers the highest Potential and Precision.

Overall, BERT_TopK and Hybrid-based methods offer the best Potential. BERT_TopK method provides a good balance precision and recall using only pretrained language model trained over raw text. Based on the nature of the strategies discussed and the results of our benchmark, it is likely to conclude that the combination of different strategies can create competitive substitution generators.

TABLE III
EVALUATION RESULTS OF LS SYSTEMS INCLUDING SUBSTITUTION GENERATION AND SUBSTITUTTION RANKING

	Precision	Recall
Synonym	74.43%	64.69%
Embedding	60.50%	56.11%
Sememe	59.35%	58.40%
BERT_TopK	68.51%	60.69%
BERT_BS	73.09%	68.70%
Hybrid	**80.73%**	**70.42%**

C. System Evaluation

Besides, we use these two previous metrics to evaluate the performance of the full pipeline. Suppose that there are m samples in test set, where the complex word of the i-th sample is w_i, the set of the annotated substitutes for w_i is o_i, and the replacement of the original word is t_i.

Precision (PRE): The proportion with which the replacement of the original word is either the original word itself or is in the gold standard.

$$Precision = \frac{\sum_{i=1}^{m} (\mathbb{I}\{t_i = w_i\} \| \mathbb{I}\{t_i \in o_i\})}{m} \tag{8}$$

where, if t_i and w_i are the same word, $\mathbb{I}\{t_i = w_i\}$ is set to 1, else 0; if t_i belonging to o_i is set to 1, else 0.

Accuracy (ACC): The proportion with which the replacement of the original word is not the original word and is in the gold standard.

$$Accuracy = \frac{\sum_{i=1}^{m} 1_{t_i \in o_i}}{m} \tag{9}$$

The experimental results are shown in Table III. We compare the full pipeline results of the five methods. Hybrid attains the highest Accuracy and Precision. BERT-based methods also achieve satisfying experiment results, especially BERT_BS. Although the results of Synonym are very encouraging, the main drawback of Synonym is its coverage. The best English LS method [8] on its benchmark dataset NNSeval obtained a Precision score of 0.526 and an Accuracy score of 0.436. Compare with English LS task, we can find that the three approaches (Synonym, BERT_BS, and Hybrid) on Chinese LS task can be served as strong baselines.

D. Ablation Study

To further analyze the advantages and disadvantages of all approaches, we do more experiments in this section.

1) Influence of Ranking Features

To determine the importance of each ranking feature, we make an ablation study by removing one feature in turn. The methodologies for substitution generation and substitution ranking are highly overlapped. For example, the following generation/ranking pairs (Embedding, Similarity), (BERT, Language Model), and (Sememe, Hownet) use the same information resource. Whether one information resource used both substitution

TABLE IV
ABLATION STUDY RESULTS OF THE RANKING FEATURES. "W/O" DENOTES "WITHOUT"

	Synonym		Embedding		Pretrained		Sememe		Hybrid	
	PRE	ACC	PRE	ACC	PRE	ACC	PRE	ACC	PRE	ACC
w/o Language	72.33%	62.60%	58.78%	54.39%	69.27%	65.27%	58.40%	57.44%	77.86%	67.56%
w/o Similarity	70.99%	64.12%	**60.88%**	**56.49%**	66.03%	63.93%	49.24%	48.28%	76.72%	69.27%
w/o Frequency	70.42%	48.66%	56.68%	52.29%	72.90%	62.60%	55.92%	54.96%	76.15%	53.82%
w/o Hownet	73.47%	63.93%	57.44%	53.05%	67.75%	64.69%	59.35%	58.40%	79.77%	69.66%
Full	**74.43%**	**64.69%**	60.50%	56.11%	**73.09%**	**68.70%**	**59.35%**	**58.40%**	**80.73%**	**70.42%**

generation and substitution ranking affects the performance. The results are presented in Table IV.

We first analyze the influence of each feature on the performance of each lexical simplification method. We can see that all approaches combining all four features achieve the best results, excluding similarity feature for Embedding, which means all features have a positive effect. Embedding removing Similarity feature produces almost identical results with Embedding combining all features. Word Embedding-based approach has already used word embeddings to generate substitute candidates which lead to similarity feature that does not affect substitution ranking. (BERT, Language Model) and (Sememe, Hownet) have separate concerns. For example, Language model in candidate ranking is used to compute the probability of a sentence or sequence of words, and Pre-trained in candidate generation is just for a word. Hownet in candidate ranking is used to compute the similarity between two words by considering all senses, and Sememe in candidate generation only considers one sense. For the two pairs, we can see that the two models owning the all features have the best results.

2) Influence of Different Pre-trained Models for Substitute Generation

From the above experiments, we know pre-trained modeling BERT achieves great results for CLS. Now, we will use more pre-trained models to do experiments. We choose the following pre-trained models:

BERT: 12-layer, 768-hidden, 12-heads, 110 M parameters. BERT using in the paper randomly selects Chinese characters to mask.

Ernie-1.0[7] 12-layer, 768-hidden, 12-heads, 110 M parameters. ERNIE is designed to learn language representation enhanced by knowledge masking strategies, which include entity-level masking and phrase-level masking.

Roberta[8] 12-layer, 768-hidden, 12-heads, 110 M parameters. RoBERTa is trained with dynamic masking, full-sentences without NSP loss, large mini-batches and a larger byte-level BPE.

ELECTRA[9] 12-layer, 768-hidden, 12-heads, 102 M parameters. ELECTRA employs a new generator-discriminator framework. The generator is typically a small MLM that learns to predict the original words of the masked tokens. The discriminator is trained to discriminate whether the input token is replaced by the generator. Note that, we only use the generator here for MLM task.

[7]https://github.com/PaddlePaddle/ERNIE

[8]https://huggingface.co/hfl/chinese-roberta-wwm-ext

[9]https://huggingface.co/hfl/chinese-electra-base-generator

TABLE V
INFLUENCE OF DIFFERENT PRE-TRAINED MODELS

	SG			SR	
	PRE	RE	F1	PRE	ACC
BERT	**31.41%**	**26.23%**	**28.59%**	**73.09%**	**68.70%**
Ernie-1.0	25.11%	22.42%	23.69%	65.46%	61.64%
Roberta-WWM	25.15 %	25.25%	25.20%	67.56%	63.74%
ELECTRA	15.32%	13.43%	14.31%	50.19%	45.23%
Macbert	15.41%	24.47%	18.91%	60.88%	54.01%

Macbert[10] 12-layer, 768-hidden, 12-heads, 102 M parameters. Instead of masking with [MASK] token, which never appears in the fine-tuning stage, Macbert uses similar words for the masking purpose. Macbert uses a percentage of 15% input words for masking, where 80% will replace with similar words, 10% replace with a random word, and keep with original words for the rest of 10%.

Table V shows the results of the experiments using different pre-trained models on HanLS dataset. We can see that BERT based modeling obtains all the highest values over the four other models. Compared with other pre-trained models, ELECTRA and Macbert are unsuited for use in the LS task. ELECTRA adopts a generator-discriminator framework and uses a pre-training task called replaced token detection. The generator is trained to perform masked language modeling, and the discriminator is trained to distinguish tokens in the data from tokens that have been replaced by generator samples. After pre-training, ELECTRA throws out the generator and fine-tunes the discriminator on downstream tasks. In contrast to ELECTRA, we use the generator to generate the substitute candidates which reduces the effect of ELECTRA. In Macbert, instead of masking with "[MASK]" token, which never appears in the fine-tuning stage, it uses similar words for the masking purpose. Although Ernie-1.0 and Roberta-WWM outperform BERT on many tasks, pre-trained BERT model is more fit for lexical simplification. If in the future a better Bert model is available, one can try to replace the Bert model in this paper to further improve the performance of LS system.

3) Error Analysis

In this subsection, we analyze all proposed approaches to understand the sources of its errors. We use PLUMBErr tool [6] to assess all steps taken by LS systems, and identify five types of errors.

1) NoError: No error during simplification.

2) NoCandi: No candidate substitutions are produced.

3) NoSimplerCandi: No simpler candidates are produced.

[10]https://huggingface.co/hfl/chinese-macbert-base

TABLE VI
ERROR CATEGORISATION RESULTS OF THE BASELINES

	NoCandi	3 (NoSimplerCandi)	4 (ChangedMeaning)	5 (NoSimplify)	1 (NoError)
Synonym	97(18%)	51(10%)	185(35%)	50(10%)	289(55%)
Embedding	146(28%)	111(21%)	230(44%)	85(16%)	209(40%)
BERT	58(11%)	**30(6%)**	164 (31%)	**23(4%)**	**337(64%)**
Sememe	146(28%)	56(11%)	218(42%)	39(7%)	267(51%)
Hybrid	**50(10%)**	57(11%)	**155(30%)**	51(10%)	318(61%)

4) ChangedMeaning: Replacement compromises the sentence's grammaticality or meaning.

5) NoSimpify: Replacement does not simplify the word.

Errors of NoCandi and NoSimplerCandi are made during Substitution Generation, and error ChangedMeaning and ChangedMeaning during Substitution Ranking. Table VI shows the count and proportion (in brackets) of instances in HanLS in which each error was made. It shows that BERT correctly simplifies the largest number of problems while making the fewest errors of NoSimplerCandi and NoSimplify. However, it can be noticed that BERT makes many errors of ChangedMeaning. Hybrid makes the fewest error of NoCandi and ChangedMeaning. Embedding making the most mistakes for each step is the worst method compared with other methods. By analyzing the output produced after each step, we found that this is caused by producing many semantically related but not similar words as substitute candidates. Synonym and Sememe make few errors of NoSimplerCandi and NoSimplify, but they make many errors of NoCandi and ChangedMeaning. They are based on linguistic databases, in which many complex words cannot be found in the databases. Overall, the results are in accordance with the conclusions of the above experiments.

VI. CONCLUSION

In this paper, we manually built a dataset for the performance evaluation of Chinese lexical simplification (CLS) system automatically. We proposed five different methods to generate the substitute candidates and introduced four high-quality features to rank the substitute candidates. Experiment results have shown that synonym-based approach, BERT-based approach, and hybrid method achieved better results. We believe the proposed CLS systems will serve as strong baselines and the created dataset can accelerate the research on this topic for future research. Despite some initial positive results on a difficult task, we note that the performance of CLS system can be affected by substitution generation and substitution ranking. In the future, we will incorporate some prior knowledge into pre-trained language model for CLS.

REFERENCES

[1] J. De Belder and M.-F. Moens, "Text simplification for children," in *Proc. SIGIR Workshop Accessible Search Syst. Workshop*, 2010, pp. 19–26.

[2] G. H. Paetzold and L. Specia, "Unsupervised lexical simplification for non-native speakers," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3761–3767.

[3] H. Saggion, "Automatic text simplification," *Synth. Lectures Hum. Lang. Technol.*, vol. 10, no. 1, pp. 1–137, 2017.

[4] J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait, "Practical simplification of english newspaper text to assist aphasic readers," in *Proc. AAAI Workshop Integrating Artif. Intell. Assistive Technol.*, 1998, pp. 7–10.

[5] G. Glavaš and S. Štajner, "Simplifying lexical simplification: Do we need simplified corpora?" in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 63–68.

[6] G. Paetzold and L. Specia, *"Lexical simplification with neural ranking*," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, Short Papers, 2017, pp. 34–40.

[7] S. Gooding and E. Kochmar, "Recursive context-aware lexical simplification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4853–4863.

[8] J. Qiang, Y. Li, Y. Zhu, Y. Yuan, and X. Wu, "Lexical simplification with pretrained encoders," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8649–8656.

[9] T. Kajiwara, H. Matsumoto, and K. Yamamoto, "Selecting proper lexical paraphrase for children," in *Proc. 25th Conf. Comput. Linguistics Speech Process.*, 2013, pp. 59–73.

[10] T. Kajiwara and K. Yamamoto, "Evaluation dataset and system for japanese lexical simplification," in *Proc. ACL-IJCNLP Student Res. Workshop*, 2015, pp. 35–40.

[11] S. Bott, L. Rello, B. Drndarević, and H. Saggion, "Can spanish be simpler? lexsis: Lexical simplification for spanish," in *Proc. COLING*, 2012, pp. 357–374.

[12] L. Rello, R. Baeza-Yates, L. Dempere-Marco, and H. Saggion, "Frequent words improve readability and short words improve understandability for people with dyslexia," in *Proc. IFIP Conf. Human Comput. Interaction*, 2013, pp. 203–219.

[13] E. Rennes and A. Jönsson, "A tool for automatic simplification of swedish texts," in *Proc. 20th Nordic Conf. Comput. Linguistics*, 2015, pp. 317–320.

[14] S. M. Aluísio and C. Gasperin, "Fostering digital inclusion and accessibility: The porsimples project for simplification of portuguese texts," in *Proc. NAACL HLT Young Investigators Workshop Comput. Approaches Lang. Amer., Assoc. Comput. Linguistics*, 2010, pp. 46–53.

[15] J. Yang, "What makes learning chinese characters difficult? the voice of students from english secondary schools," *J. Chin. Writing Syst.* vol. 2, no. 1, 2018, pp. 35–41.

[16] S. W. Wong *et al.*, "Perception of native english reduced forms in chinese learners: Its role in listening comprehension and its phonological correlates," *TESOL Quart.* vol. 51, no. 1, pp. 7–31, 2017.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in: NAACL, 2018.

[18] E. Pavlick and C. Callison-Burch, "Simple Ppdb: A paraphrase database for simplification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2016, pp. 143–148.

[19] M. Maddela and W. Xu, "A word-complexity lexicon and a neural readability ranking model for lexical simplification," in *Proc. EMNLP*, 2018, pp. 3749–3760.

[20] S. Devlin and J. Tait, "The use of a psycholinguistic database in the simpli cation of text for aphasic readers," *Linguistic Databases*, vol. 1, pp. 161–173, 1998.

[21] O. Biran, S. Brody, and N. Elhadad, "Putting it simply: A context-aware approach to lexical simplification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.*, 2011, pp. 496–501.

[22] M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee, "For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia," in *Proc. NAACL*, 2010, pp. 365–368.

[23] C. Horn, C. Manduca, and D. Kauchak, "Learning a lexical simplifier using wikipedia," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, 2014, pp. 458–463.

[24] J. Qiang and X. Wu, "Unsupervised statistical text simplification," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1802–1806, Apr. 2021.

[25] Y. Sun *et al.*, "ERNIE: Enhanced representation through knowledge integration," 2019, *arXiv:1904.09223*. http://arxiv.org/abs/1904.09223

[26] W. Zhou, T. Ge, K. Xu, F. Wei, and M. Zhou, "BERT-Based lexical substitution," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3368–3373.

[27] G. H. Paetzold and L. Specia, "A survey on lexical simplification," *J. Artif. Intell. Res.*, vol. 60, pp. 549–593, 2017.
[28] R. Keskisarkka, *Automatic Text Simplification Via Synonym Replacement* (2012).
[29] T. Kodaira, T. Kajiwara, and M. Komachi, "Controlled and balanced dataset for japanese lexical simplification," in: *Proc. ACL Student Res. Workshop*, *Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 1–7. [Online]. Available: https://www.aclweb.org/anthology/P16-3001
[30] H. Liu, S. Li, J. Zhao, Z. Bao, and X. Bai, "Chinese teaching material readability assessment with contextual information," in *Proc. Int. Conf. Asian Lang. Process.*, 2017, pp. 66–69.
[31] K. Collinsthompson, "Computational assessment text readability: A survey current future research," *ITL - Int. J. Appl. Linguistics*, vol. 165, no. 2, pp. 97–135, 2014.
[32] S. Zhang, H. Huang, J. Liu, and H. Li, "Spelling error correction with soft-masked BERT," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, Assoc. Comput. Linguistics*, 2020, pp. 882–890. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.82
[33] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu, "BERT-ATTACK: Adversarial attack against BERT using BERT," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Assoc. Comput. Linguistics*, 2020, pp. 6193–6202.
[34] J. Zhao, B. A. Zhang, and J. Cheng, "Some suggestions on the revision of the outline of the graded vocabulary for HSK," *Chin. Teach. the World* (2003).
[35] J. Mei *et al.*, Tongyici cilin (extended), *HIT IR-Lab* (1996).
[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[37] L. Bloomfield, "A set of postulates for the science of language," *Language* vol. 2, no. 3, pp. 153–164, 1926.
[38] F. Qi *et al.*, "Modeling semantic compositionality with sememe knowledge," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5706–5715.
[39] Y. Zhang, C. Yang, Z. Zhou, and Z. Liu, "Enhancing transformer with sememe knowledge," in *Proc. 5th Workshop Representation Learn. NLP, Assoc. Comput. Linguistics*, 2020, pp. 177–184.
[40] Z. Dong, Q. Dong, and C. Hao, "Hownet and the computation of meaning," in World Scientific, 2006.
[41] L. Yuming, "On green paper on language situation in china," *Appl. Linguistics*, vol. 1, 2007.
[42] Q. Liu, "Word similarity computing based on hownet," *Comput. Linguistics Chin. Lang. Process.*, vol. 7, no. 2, pp. 59–76, 2002.

Jipeng Qiang received the Ph.D. degree in computer science and technology from the Hefei University of Technology, Hefei, China, in 2016. He is currently an Associate Professor with the School of Information Engineering, Yangzhou University, Jiangsu, China. From 2014 to 2016, he was a Visiting Ph.D. Student with the Artificial Intelligence Lab, University of Massachusetts Boston, Boston, MA, USA. He has authored or coauthored more than 40 papers, including AAAI, TKDE, and TKDD. His research interests mainly include topic modeling and natural language processing. He was the recipient of two grants from the National Natural Science Foundation of China, one grant from the Natural Science Foundation of Jiangsu Province of China, one grant from Natural Science Foundation of the Higher Education Institutions of Jiangsu Province of China.

Xinyu Lu is currently working toward the B.E. degree with the Department of Computer Science and Technology, Yangzhou University, Jiangsu, China. His research focuses on text simplification.

Yun Li received the M.S. degree in computer science and technology from the Hefei University of Technology, Hefei, China, in 1991, and the Ph.D. degree in control theory and control engineering from Shanghai University, Shanghai, China, in 2005. He is currently a Professor with the School of Information Engineering, Yangzhou University, Jiangsu, China. He has authored or coauthored more than 100 scientific papers. His research interests include data mining and cloud computing.

Yunhao Yuan received the M.Eng. degree in computer science and technology from Yangzhou University, Jiangsu, China, in 2009 and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2013. He is currently an Associate Professor with the School of information Engineering, Yangzhou University. His research interests include pattern recognition, data mining, and image processing.

Xindong Wu (Fellow, IEEE) received the B.S. and M.S. degrees in computer science from the Hefei University of Technology, Hefei, China and the Ph.D. degree in artificial intelligence from The University of Edinburgh, Edinburgh, U.K. He is currently a Yangtze River Scholar with the School of Computer Science and Information Engineering, Hefei University of Technology, and the President of Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, China. His research interests include data mining, big data analytics, knowledge-based systems, and web information exploration. He is currently the Steering Committee Chair of the IEEE International Conference on Data Mining, the Editor-in-Chief of *Knowledge and Information Systems* (by Springer), and a series Editor-in-Chief of the Springer Book Series on *Advanced Information and Knowledge Processing*. Between 2005 and 2008, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING by the IEEE Computer Society. He was the Program Committee Chair or co-Chair for the 2003 IEEE International Conference on Data Mining, the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, and the 19th ACM Conference on *Information and Knowledge Management*. He is a Fellow of an AAAS.