Unsupervised Statistical Text Simplification

Jipeng Qiang and Xindong Wu, *Fellow, IEEE*

Abstract—Most recent approaches for Text Simplification (TS) have drawn on insights from machine translation to learn simplification rewrites from the monolingual parallel corpus of complex and simple sentences, yet their effectiveness strongly relies on large amounts of parallel sentences. However, there has been a serious problem haunting TS for decades, that is, the availability of parallel TS corpora is scarce or not fit for the learning task. In this paper, we will focus on one especially useful and challenging problem of unsupervised TS without a single parallel sentence. To the best of our knowledge, we present the first unsupervised text simplification system based on phrase-based machine translation system, which leverages a careful initialization of phrase tables and language models. On the widely used WikiLarge and WikiSmall benchmarks, our system respectively obtains 39.08 and 25.12 SARI points, even outperforms some supervised baselines.

Index Terms—Text simplification, machine translation, unsupervised

1 Introduction

Automatic text simplification (TS), a research topic that started 20 years ago, now has taken on a central role in natural language processing research not only because of the interesting challenges it possesses but also because of its social implications [1]. Most recent approaches have addressed text simplification as a monolingual machine translation problem translating from complex sentences to simplified sentences [2], [3], [4], [5]. These approaches work very well only when provided with a massive parallel corpus of complex and simple sentences. Unfortunately, these approaches are currently limited by the scarcity of parallel corpus.

Two parallel simplification benchmarks WikiSmall [6] and WikiLarge [4] which align sentences from the 'ordinary' English Wikipedia and the 'simple' English Wikipedia have been criticized recently because they contain a large proportion of: inaccurate simplifications (not aligned or only partially aligned) and inadequate simplifications (not much simpler than complex sentence) [7], [8], which lead to systems that generalize poorly [2]. For example, two-sentence pairs from WikiLarge in Table 1 illustrate the existing problems. The pair of sentence 1 and sentence 2 respectively describes two totally different contents, denoted as inaccurate simplification pair. The pair of sentence 3 and sentence 4 is an inadequate simplification since sentence 4 is not simpler than sentence 3. By manually comparing the sentences pairs from randomly sampling 200 sentences pairs from WikiLarge, the percent of sentence pairs (not aligned) and sentence pairs (not simpler) are 19 and 32 percent, respectively. The authors [7] observed that about 50 percent of the sentence pairs in WikiLarge benchmark are not simplifications. Therefore, learning to simplify using machine translation techniques based on the existing parallel corpus is far from meeting our needs.

- *J. Qiang is with the Department of Computer Science, Yangzhou University, Yangzhou, Jiangsu 225127, P. R. China. E-mail: jpqiang@yzu.edu.cn.*
- *X. Wu is with the Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Ministry of Education, Hefei, Anhui 10084, China, and also with the Mininglamp Academy of Sciences, Mininglamp, Beijing 100084, China. E-mail: xwu@hfut.edu.cn.*

Manuscript received 23 Feb. 2019; revised 24 Aug. 2019; accepted 6 Oct. 2019. Date of publication 16 Oct. 2019; date of current version 5 Mar. 2021.
(Corresponding author: Jipeng Qiang.)
Recommended for acceptance by Y. Chang.
Digital Object Identifier no. 10.1109/TKDE.2019.2947679

In this paper, we focus on one especially challenging research problem of unsupervised TS without a single parallel sentence. We propose a novel phrase-based unsupervised TS system by leveraging a careful initialization of phrase tables and two language models. To the best of our knowledge, the system is the first unsupervised statistical text simplification system. Our method utilizes the 'ordinary' Wikipedia as a massive knowledge base. We acquire the following useful information from Wikipedia: word embeddings, word frequencies, simple corpus, and complex corpus. Because word embeddings capture semantic properties of words and word frequencies reflect the level of difficulty, we propose a novel method to populate phrase tables by incorporating word embeddings and word frequencies. We gather the simple corpus and complex corpus by sorting the sentences of Wikipedia using Flesch reading-ease score, where higher scores indicate sentences that are easier to read and lower scores mark sentences that are more difficult to read. A simple language model and a complex language model can be learned from these two sets, which can help to improve the quality of the simplification models by performing local substitutions and word reorderings.

Our model is intuitive and supported by the following obvious advantages: 1) Unsupervised. Our model is an unsupervised system, which only utilizes unlabeled text collected from the English Wikipedia dump. 2) Very simple. Our model only has fewer hyperparameters that correspond with hundreds of millions of parameters to learn in neural text simplification models. 3) Easy to interpret. Phrase tables present the transition probabilities between difficult phrases and simpler phrases. Many existing TS methods adopt sequence to sequence models, which shows little detail about the transformation. 4) High efficiency. Our model obtains a SARI score of 39.08 on the WikiLarge benchmark, even outperforming the previous best results of all supervised TS systems. Our model surpasses the unsupervised baselines by more than 4.4 and 13.5 SARI points on WikiLarge and WikiSmall benchmarks, respectively. Our code is publicly available[1]

2 Related Work

As complex and simple parallel corpora are available, especially, the 'ordinary' English Wikipedia (EW) in combination with the 'simple' English Wikipedia (SEW), text simplification systems using machine translation systems have dominated simplification research since 2010. Original studies often used standard statistical machine translation approaches to learn the simplification of a complex sentence into a simplified sentence using the Bayes Theorem. For example, the Moses standard phrase-based system (SMT) [9] was directly adopted to simplify sentences using 3,383 pairs of complex and simple sentences. Coster and Kauchak [10] also simplified complex sentences using standard SMT on 137,000 aligned pairs of sentences extracted from EW and SEW corpus. Wubben et al. [2] investigated the SMT approach to simplification even further by incorporating a dissimilarity-based reranking mechanism to choose among possible simplification solutions. Xu et al. [3] presented an effective adaptation of SMT techniques for TS. Except for SMT systems, Neural Machine Translation (NMT) is a newly-proposed deep learning model and achieves very impressive results on bilingual machine translation [11]. Therefore, the existing architectures in NMT are directly or indirectly used for text simplification [4], [5], [12].

All of the above work is supervised TS systems, whose performance strongly relies on the availability of large amounts of parallel sentences. Two parallel benchmarks WikiSmall [6] and WikiLarge

1. https://github.com/qiang2100/UnsuperPBMT

TABLE 1
Example Sentence Pairs (Complex-Simple) Aligned between English Wikipedia and Simple English Wikipedia from the Parallel Wikipedia Simplification Corpus

Not aligned	1. [Complex] Murtaugh is named after Mark Murtaugh, who oversaw a local irrigation project. 2. [Simple] Murtaugh is a city of Idaho in the United States.
Not simpler	3. [Complex] Villandraut is a commune in the Gironde department in Aquitaine in south-western France. 4. [Simple] Villandraut is a commune. It is found in the region Aquitaine in the Gironde department in the southwest of France.

[4] contain a large proportion of: inaccurate simplifications (not aligned or only partially aligned) ; inadequate simplifications (not much simpler) [7], [8]. These problems cause the difficulty of designing a good alignment algorithm for extracting parallel sentences from EW and SEW, which is highlighted by [13]. Therefore, the available parallel sentences are not fit for training a TS system, which seriously hindered the development of text simplification.

Therefore, in this paper, we hope to adopt a novel TS method in a completely unsupervised manner without using a single parallel sentence. The work in TS that is closest to ours are from [14], [15]. Yatskar et al [15] and Paetzold et al. [14] focused on unsupervised lexical simplification method that replaces complex words with simpler alternatives based on word embeddings, which is different from our paper. Text simplification as a monolingual machine translation problem learns various types of simplification operations from a parallel corpus. On bilingual machine translation task, unsupervised machine translation systems have attracted much attention in the past two years [16], [17], [18]. Unsupervised statistical machine translation and neural machine translation systems were proposed and achieved very good performance. There are three common elements for these systems: the inferred bilingual dictionary, language models for two languages, back-translation. But, when we try to apply these systems into TS, we found that they struggle to generate simplified sentences, leaving the input sentences unchanged, as each word is translated with itself when performing word-by-word translation. Different from bilingual machine translation, the concern of TS is the replacement of difficult or unknown phrases with simpler equivalents in a single language. Therefore, in this paper, we focus on designing a targeted unsupervised system for TS.

Inspired by unsupervised bilingual machine translation, designing an unsupervised statistical method for TS still have the following major challenges: 1) Lack of simple corpus and complex corpus. The 'ordinary' English Wikipedia includes a large number of simple sentences, making it not suitable for extracting directly complex corpus. Similarly, the 'simple' English Wikipedia includes a large number of complex sentences, making it not suitable as a source for simple corpus. 2) How to populate phrase tables. Supervised TS systems need sufficient parallel sentences for populating phrase tables, which become impossible without parallel sentences. With methods using the bilingual unsupervised system, only the score of the translation of one word to itself has high value, which is not fit for TS.

3 Unsupervised Text Simplification

The architecture of our unsupervised text simplification approach is illustrated in Figure 1. Our model utilizes a phrase-based machine translation (PBMT) system [9] as the underlying backbone model. PBMT 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 main idea of our approach is first to acquire some useful knowledge from the 'ordinary' English Wikipedia[2] in preparation for TS, and then populate phrase tables and learn language models for PBMT system, denoted as our initial PBMT system. Based on the initial PBMT system, we generate a synthetic parallel corpus, which can turn the unsupervised problem into the supervised problem through iterative back-translation.

Prior Knowledge. We acquire the following useful information from Wikipedia: word embeddings, word frequencies, simple corpus, and complex corpus, where word embeddings and word frequencies are used to populate phrase tables, and simple corpus and complex corpus are used to learn simplified language model and complex language model, respectively.

1) Word embeddings. Word embedding methods represent words as continuous vectors in a low dimensional space that can capture the lexical and semantic properties of words. Here, the Glove algorithm [19] chose by us is used to learn word embeddings from Wikipedia. After obtaining word embeddings, we calculate the similarities between words, which will help to find similar words for each word. We denote $e(w)$ is the embedding of word w.
2) Word frequencies. In TS, the measure of word complexity normally takes into account word frequencies. In general, the higher the frequency, the easier the word. Therefore, word frequencies calculated from Wikipedia will help to find the simplest words from a set of similar words. Here, we denote $f(w)$ is the frequency of word w.
3) Simple corpus S and complex corpus C. Wikipedia contains a large number of simple sentences and complex sentences. In TS, the Flesch reading ease score (FRES) [1] is designed to indicate how difficult a sentence in English is to understand, and is widely used to evaluate the performance of TS. The formula for FRES is given in Equation (1). Therefore, FRES is chosen to sort all sentences from Wikipedia texts, where higher scores indicate sentences that are easier to read and lower scores mark 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. Specifically, after ranking the sentences, we collect 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 sets will be used to learn a simple language model and a complex language model, respectively.

$$206.835 - 1.015 \times (\text{total words}) - 84.6 \times \left(\frac{\text{total syllables}}{\text{total words}}\right). \tag{1}$$

Phrase Tables. While prior work for populating phrase tables relied on a parallel corpus, here we propose a novel method based on word embeddings and word frequencies. Hereinafter, we will treat phrases as single words, but the same arguments also hold for longer n-grams. Phrase tables PT are populated with the scores of the simplification of one word w_i to the other word w_j.

$$P(w_j \mid w_i) = \begin{cases} \frac{f(w_j)}{f(w_i)} cos(e(w_i), e(w_j)), if\ P(w_j \mid w_i) < 1 \\ 1, otherwise. \end{cases}, \tag{2}$$

2. http://download.wikimedia.org

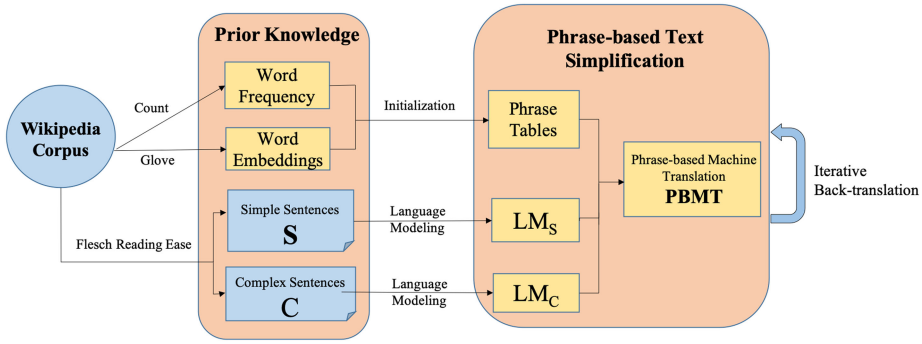

Fig. 1. Architecture of our text simplification system.

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 Equation 1 is that high frequency words are much easier to be simplified than low frequency words when all of them have very high similarity values. In addition, Equation 1 ensures that each phrase table entry always less than or equal to 1.

For example, suppose w_i='gigantic', using our method to populate phrase tables, except that $P(gigantic \mid gigantic)$ equals to 1, all of the following 'simple' words in phrase tables equal to 1: $P(huge \mid gigantic)$, $P(giant \mid gigantic)$, $P(bigger \mid gigantic)$, $P(vast \mid gigantic)$, and $P(enormous \mid gigantic)$. In this step, inappropriate words might also have higher similarity resulting in noisy phrase tables, but the following language model considers the probability of continuous multiple words that can help to correct some of the mistakes during the generation.

Language Model. The language model assigns a probability to a word sequence in the specified corpus. Both in the simple corpus S and complex corpus C, we respectively learn smoothed n-gram language models LM_S and LM_C using KenLM [20]. The two language models remain unchanged in the whole training iterations. A simple language model and a complex language model by the probability of a word sequence will help to improve the quality of the simplification models by performing local substitutions and word reorderings. The reordering model in the PBMT system accounts for different word orders across the corpus, scoring translation candidates according to the position of each translated phrase in the target corpus.

Iterative Back-Translation. Back-Translation for TS task is inspired by the success of back-translation for bilingual machine translation systems [16], [18]. Our initial seed PBMT system is denoted 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 goal of iterative back-translation turns the daunting unsupervised problem into a supervised learning task. After obtaining $m^0_{C\to S}$ system, we generate a synthetic simple corpus S^0 by simplifying the complex corpus C using $m^0_{C\to S}$. Given the synthetic parallel complex-simple sentences (S^0, C), we train and tune a standard PBMT system $m^1_{S\to C}$ over it from simple sentences to complex sentences. Next, we perform both generation and training process but in the reverse direction. This process is repeated iteratively, detailed in Algorithm 1.

In the beginning, many entries in the phrase tables are probably mistaken, due to noisy simplified sentences yielding by seed $m^0_{C\to S}$. Even then, the language model can help to correct some of the mistakes during the generation. As long as that happens, after each iteration, we will produce more accurate phrase tables. which makes the PBMT model stronger. At each iteration, it requires considerable time if we yield synthetic sentences using all source corpus. For accelerating the experiments, we randomly choose a subset of 10 million sentences from the source corpus to yield parallel sentences.

Algorithm 1. Unsupervised Text Simplification

Input: Wikipedia W
Output: TS models $\{m^0_{C\to S}, m^1_{C\to S}$ ..., $m^N_{C\to S}\}$

1: Learn word embeddings from W using Glove [19].
2: Count word frequency from W.
3: Gather simple corpus S and complex corpus C from W using Flesch reading-ease score.
4: Populate phrase tables PT using Equation (1).
5: Learn two language models LM_S and LM_C from S and C using KenLM.
6: Combine (PT, LM_S and LM_C) to build $m^0_{C\to S}$ using PBMT.
7: Simplify C for yielding S^0 using $m^0_{C\to S}$.
8: **for** i =1 to N do **do**
9: Train model $m^i_{S\to C}$ using (S^{i-1}, C).
10: Simplify S for yielding C^i using $m^i_{S\to C}$.
11: Train model $m^i_{C\to S}$ using (C^i, S).
12: Simplify C for yielding S^i using $m^i_{C\to S}$.
13: **end for**

4 Experiments

We design experiments to answer the following questions:

Q1. Effectiveness: Does our unsupervised model outperforms state-of-the-art competitors, even supervised approaches leveraging parallel sentences? Does our model really simplify complex sentences?

Q2. Unsupervised bilingual machine translation approaches: Do unsupervised bilingual machine translation approaches fit for text simplification?

4.1 Experiment Setup

Dataset. We use three widely used simplification datasets (WikiSmall, WikiLarge, and Newsela) to do experiments. The training/development/test set in WikiSmall, WikiLarge and Newseal have 89,042/205/100, 296,402/2000/359, 94,208/1,129/1,077 sentence pairs, respectively. The details of the three datasets are illustrated in this paper [4].

Metrics. Following previous work, three widely used metrics in text simplification are chosen in this paper [3], [21]. SARI [4] is a recent text-simplification metric by comparing the output against the simple and complex simplifications.[3] BLEU [11] is one traditional machine translation metric to assess the degree which translated simplifications differed from reference simplifications. Flesch

3. We used the implementation of SARI in [3].

TABLE 2
Automatic Evaluation on Three Dataset (WikiLarge, WikiSmall, and Newsela) Test Sets

		WikiLarge			WikiSmall			Newsela		
	Model	BLEU	FRES	SARI	BLEU	FRES	SARI	BLEU	FRES	SARI
Test	Complex	97.35	68.40	28.70	49.85	56.73	4.34	20.87	74.22	2.74
	Simple	97.41	70.94	49.89	100.00	69.07	63.62	100.00	93.54	70.25
Supervised	PBMT-R	81.11	74.55	**38.56**	46.31	61.27	15.97	18.19	79.81	15.77
	Hybrid	48.97	**85.53**	31.40	**53.94**	**70.04**	**30.46**	14.46	87.70	**30.00**
	EncDecA	**88.85**	72.18	35.66	47.93	60.03	13.61	21.70	86.92	24.12
	DRESS	77.18	76.35	37.08	34.53	69.91	27.48	**23.21**	**90.54**	27.37
Unsupervised	NMT	**86.63**	73.25	33.43	**44.48**	56.21	10.36	**18.94**	78.56	15.12
	SMT	85.05	66.65	34.66	43.67	57.41	11.60	17.17	74.86	13.01
	PBMT (Iter. 0)	57.89	**80.84**	**39.08**	26.03	**76.50**	**25.12**	14.93	86.86	23.75
	PBMT (Iter. 5)	79.16	79.45	38.97	39.19	75.17	24.65	17.35	**87.72**	**24.29**

Bold face highlights the best number for supervised and unsupervised TS methods, respectively. The scores of orignial complex and simple sentences in test set are shown the first two lines. The results of supervised methods are collected from [4]. PBMT (Iter. 0) just uses the unsupervised phrase table without back-translation.

reading ease score measures the readability of the output [22]. A higher FRES represents simpler output.[4]

Comparison Systems. We will compare our method with the followings: 1) Supervised: PBMT-R, Hybrid, EncDecA, Dress. PBMT-R is a phrase-based method with a reranking post-processing step [2]. Hybrid performs sentence splitting and deletion operations based on discourse representation structures and then simplifies sentences with PBMT-R [23]. NMT is a basic attention-based encoder-decoder model [5]. DRESS is an encoder-decoder model coupled with a deep reinforcement learning framework [4]. 2) Unsupervised: NMT and SMT. We respectively choose one from unsupervised neural machine translation method [18] and statistical machine translation method [16]. Due to only one language in TS, we delete this step of mapping word embeddings from the source language to the target language in NMT and SMT, and adopt the same word embeddings for simple and complex sentences.

For our model, we use Moses scripts [9] for tokenization. Our model is trained with true-casing and chooses 300 as the dimension of word embeddings. When populating phrase tables, we consider the most frequent 50,000 words, and align each of them to its 200 most similar words, resulting in a phrase table of 10 million phrase tables which we score using Equation 2. At each iteration, we translate 1 million sentences randomly sampled from S or C based on the direction of simplification. Except for initialization, we use phrase tables with phrases up to length 4.

4.2 Results and Qualitative Study

Results. Table 2 shows the results of all models on three datasets. Unsupervised TS systems (NMT, SMT, and PBMT) are only trained once using the knowledge acquired from Wikipedia, and then simplify the test set of three datasets. Supervised TS systems separately need to learn from the train set of each dataset. Since different metrics for evaluation TS can vary considerably across datasets. We report the results on the complex and simple of the test set, which shows on the first two lines of Table 2. By looking at the results of the source complex sentences using the BLEU metric, it outperforms all automatic TS systems. This is because BELU was designed to evaluate bilingual translation systems. When applied to monolingual tasks like simplification, BLEU does not take into account anything about the differences between the input and the references. Therefore, BLEU is not well suited for assessing simplicity for a lexical [3] nor a structural [24] point of view.

The left blocks in Table 2 summarize the results of our automatic evaluation on the WikiLarge dataset. As can be seen, our model PBMT obtains a SARI score of 39.08, even outperforming the previous best result of all supervised TS systems, which indicates that the model has indeed learned to simplify the complex sentences. On the FRES metric, our model achieves better results than supervised approaches, except Hybrid. If we check out the results of Hybrid, the simplified sentences of Hybrid only contain several words, not real sentences, resulting in higher FRES scores. On the BLEU metric, our method worse than unsupervised NMT and SMT, because unsupervised NMT and SMT made little change of the source found by checking out their results.

The middle blocks in Table 2 report the results on the WikiSmall dataset. Our model surpasses unsupervised baselines by more than 13.5 SARI points. FRES and BLEU follow a similar pattern as on WikiLarge. Compared with supervised TS methods, our model obtains the best SARI and FRES scores, except Hybrid. Incorporating iterative back-translation, our model obtains better BLEU scores and slightly lower SARI and FRES scores. The right blocks in Table 2 report the results on Newsela. Our model outperforms unsupervised NMT and SMT systems on FRES and SARI metrics. Compared with supervised TS systems, our model still has good performance. DRESS achieved higher FRES scores on Newsela dataset, not WikiLarge and WikiSmall, because Newsela is a better parallel dataset that can be better used by supervised neural text simplification system DRESS.

After iterative back-translation, PBMT (Iter 5) performs better than PBMT without back-translation (Iter 0) using BELU on all datasets, although PBMT (Iter 5) performs a little worse than PBMT (Iter 0) using FRES and SARI on WikiLarge and WikiSmall. This is because generated simplified sentences are getting close to the original sentences when increasing the number of iterations. Phrase tables learned from the synthetic parallel sentences can help to correct some wrong entries. But it also enhances the probability between each word and itself, namely, it can cause that one complex word is translated to itself not its simple equivalent. Therefore, we need to make a tradeoff about the iterative number of back-translation. Overall, the back-translation strategy for PBMT method can bring good results. In conclusion, we can see that PBMT outperforms previous unsupervised baselines on three datasets, even supervised baselines, which indicate that our method is effective at creating simpler output. Even before iterative back-translation, our model significantly outperforms the baselines and can be trained in a few minutes.

Qualitative Study. Table 3 shows example simplification from WikiLarge dataset on all methods. Unsupervised NMT and SMT did not make any simplification to the source complex sentence. Supervised PBMT-R, EncDecA, and DRESS have made certain simplifications, e.g., "vanished" of the source sentence is simplified into "disappeared". Our model reduces more linguistic complexity of the source

4. FRES's implementation is in https://github.com/nltk/nltk_contrib/tree/master/nltk_contrib/readability

TABLE 3
System Output for One Sentence from WikiLarge

Complex	Many species had vanished by the end of the nineteenth century, with European settlement.
Simple	with Euopean settlement many species have been vanished.
PBMT-R	Many species had **just disappeared** by the end of the **19th** century, with European settlement.
Hybrid	species had vanished , .
EncDecA	Many species had **disappeared** by the end of the nineteenth century , with European settlement .
DRESS	Many species had **disappeared** by the end of the nineteenth century .
NMT	Many species had vanished by the end of the nineteenth century, with European settlement.
SMT	Many species had vanished by the end of the century , nineteenth with European settlement .
PBMT (Iter. 0)	Many species had **gone** by the end of the **19th** century, with European settlers.
PBMT (Iter. 5)	Many species had **gone** by the end of the **19th** century, with European settlers.

'Complex' and 'Simple' are the source and reference sentences in the test set. Substitutions are shown in bold

sentence, while still retaining its original information and meaning. From all the results, we can see that the "vanished" can be simplified as "disappeared" or "gone", and the "nineteenth" can be simplified as "19th". For SMT and our model, these simplifications are both present in the unsupervised phrase tables. In phrase tables of SMT, the conditional probability from "vanished" to "vanished", "disappeared" and "gone" are: $P(vanished \mid vanished) = 0.999$, $P(disappeared \mid vanished) = 1.359\text{e}{-3}$ and $P(gone \mid vanished) = 3.848e{-9}$. These indicate that unsupervised SMT is impossible to make a change. In phrase tables of our model, the three conditional probability are $P(vanished \mid vanished) = 1$, $P(disappeared \mid vanished) = 1$ and $P(gone \mid vanished) = 1$. Under the same conditional probability, our model is easy to generate simplified sentences under the help of a simple language model. This verifies that the initial phrase tables are very important for PBMT.

5 Conclusion

Considering the scarcity of parallel simplification data, we make a novel attempt for unsupervised text simplification task. We propose a novel phrase-based unsupervised text simplification system, which only uses the ordinary English Wikipedia as a knowledge base without any simple corpus. We test the proposed model on WikiLarge, WikiSmall and Newsela datasets. The experimental results show that our model achieves significant improvement, even outperforms supervised text simplification systems.

Acknowledgments

This research is partially supported by the National Key Research and Development Program of China under grant 2016YFB1000900; the National Natural Science Foundation of China under grants 61703362 and 91746209; the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under grant IRT17R32; and the Natural Science Foundation of Jiangsu Province of China under grant BK20170513.

References

[1] H. Saggion, "Automatic text simplification," *Synthesis Lectures Human Lang. Technol.*, vol. 10, no. 1, pp. 1–137, 2017.
[2] S. Wubben, A. V. D. Bosch, and E. Krahmer, "Sentence simplification by monolingual machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. Int. Joint Conf. Natural Lang. Process.*, 2012, pp. 1015–1024.
[3] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, "Optimizing statistical machine translation for text simplification," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 401–415, 2016.
[4] X. Zhang and M. Lapata, "Sentence simplification with deep reinforcement learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 584–594.
[5] S. Nisioi, S. Štajner, S. P. Ponzetto, and L. P. Dinu, "Exploring neural text simplification models," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, vol. 2, pp. 85–91.
[6] Z. Zhu, D. Bernhard, and I. Gurevych, "A monolingual tree-based translation model for sentence simplification," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 1353–1361.
[7] W. Xu, C. Callison-Burch, and C. Napoles, "Problems in current text simplification research: New data can help," *Trans. Assoc. Comput. Linguistics*, vol. 3, no. 1, pp. 283–297, 2015.
[8] S. Štajner, H. Béchara, and H. Saggion, "A deeper exploration of the standard pb-smt approach to text simplification and its evaluation," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics/7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 823–828.
[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al., "Moses: Open source toolkit for statistical machine translation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics Companion Vol. Proc. Demo Poster Sessions*, 2007, pp. 177–180.
[10] W. Coster and D. Kauchak, "Simple english wikipedia: A new text simplification task," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Human Lang. Technol.*, 2011, pp. 665–669.
[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 6–17.
[12] T. Wang, P. Chen, J. Rochford, and J. Qiang, "Text simplification using neural machine translation," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 4270–4271.
[13] W. Hwang, H. Hajishirzi, M. Ostendorf, and W. Wu, "Aligning sentences from standard wikipedia to simple wikipedia," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2015, pp. 211–217.
[14] G. H. Paetzold and L. Specia, "Unsupervised lexical simplification for non-native speakers," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3761–3767.
[15] 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. Human Lang. Technol.: Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 365–368.
[16] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato, "Phrase-based & neural unsupervised machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 5039–5049.
[17] M. Artetxe, G. Labaka, and E. Agirre, "Unsupervised statistical machine translation," *Int. Conf. Learn. Representations*, pp. 3632–3642, 2018.
[18] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 73–84.
[19] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
[20] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proc. 6th Workshop Statistical Mach. Transl.*, 2011, pp. 187–197.
[21] K. Woodsend and M. Lapata, "Learning to simplify sentences with quasi-synchronous grammar and integer programming," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 409–420.
[22] P. J. Kincaid, R. P. Fishburne, R. J. L. Richard, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," Tech. Rep., DTIC Document, pp. 8–75, 1975.
[23] S. Narayan and C. Gardent, "Hybrid simplification using deep semantics and machine translation," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 435–445.
[24] E. Sulem, O. Abend, and A. Rappoport, "Simple and effective text simplification using semantic and neural methods," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 162–173.