Lexical Substitution is Practical for Rare Word Simplification

Takumi Maruyama
Nagaoka University of Technology
maruyama@jnlp.org

Kazuhide Yamamoto
Nagaoka University of Technology
yamamoto@jnlp.org

Abstract

We attempted to generate a sentence by using the concept of core vocabulary whilst preserving the original meaning of the sentence in the text simplification subtask. The correlation between word simplicity and frequency shows that many complex words are less frequently used. Hence, accurately simplifying sentences containing rare words is important in the simplification task. We explored a simplification model that works robustly. The machine translation approach and the lexical substitution were evaluated in the dataset that includes rare words (referred to as RARE) and the dataset that does not include rare words (referred to as NORMAL). The machine translation approach exhibits the best performance in the NORMAL dataset, but it exhibits very poor performance in the RARE dataset. Meanwhile, the lexical substitution model works robustly in both the datasets, and it displays fluent as well as adequate outputs. These results imply that for practical text simplification systems, accumulating human knowledge is important even though its construction is costly.

1 Introduction

The number of foreigners who are visiting Japan has been increasing over the years. Japan hosts approximately 28 million visitors in a year[1]. In addition, approximately 2.32 million foreigners are living in the country[2], and this number is increasing by time. According to a survey conducted by the National Institute for Japanese Language and Linguistics, the number of people who can understand basic Japanese exceeds the number of people who can understand English (Iwata, 2010). Hence, a simplified text is one of the important methods of providing information to foreigners.

The level of simplicity of a certain text is determined by various factors, such as the sentence length, the vocabulary used and the discourse structure. Thus, objectively defining the level of text simplicity is difficult. Therefore, we focus on the extent of the vocabulary used in a text. According to the studies emphasising on the requirement of large vocabulary to understand a text, Laufer (1992) and Nation (2006) showed that reading comprehension is correlated with the ration of vocabulary (referred to as text covering ratio) known by a reader with regard to a certain text. That is, if a text is represented only by a certain known vocabulary, then understanding that text becomes easy. We challenge to simplify the original sentences with a vocabulary of 2,000 words (referred to as the core vocabulary) whilst preserving its original meaning.

In practical text simplification systems, simplifying rare words is very important. The correlation between word simplicity and frequency shows that many complex words are used less frequently. However, no knowledge is available on the most robust model that works for rare words. Therefore, to determine a robust system in this task, we compare the machine translation approach and lexical substitution by using the paraphrasing dictionary in the evaluation dataset containing rare words.

[1] https://www.jnto.go.jp/jpn/statistics/visitor_trends
[2] https://www.e-stat.go.jp

32nd Pacific Asia Conference on Language, Information and Computation
Hong Kong, 1-3 December 2018

2 Related Works

Automatic text simplification is a task that reduces the complexity of vocabulary and expressions whilst preserving the main meaning of a text. This technique can be used to render many text resources available for a wide range of readers, including children, non-native speakers and disabled people. As a pre-processing step, text simplification can improve the performance of natural language processing tasks, such as parsing (Chandrasekar et al., 1996), summarisation (Siddharthan et al., 2004; Xu and Grishman, 2009), semantic role labelling (Vickrey and Koller, 2008), information extraction (Miwa et al., 2010) and machine translation (Chen et al., 2012; Štajner and Popvić, 2016). Automatic text simplification involves several subtasks, such as complex word identification, lexical simplification, syntactic simplification, sentence splitting and sentence compression. Various simplification approaches are employed based on context, sentence length and syntactic structure of the source sentence. Generally, multiple simplification approaches work together to simplify a text. Research for automatic text simplification is generally divided into three systems: rule-based, lexical simplification and machine translation.

Rule-based systems use rules that are manually created for syntactic simplification and substitute difficult words by using a predefined vocabulary. By analysing a syntactic structure, a sentence with a particular structure or a complex structure can be transformed into a simple structure (Siddharthan and Angrosh, 2014; Lee et al., 2017).

Recently, machine translation approaches for text simplification present good performance (Wubben et al., 2012; Nisioi et al., 2017; Zhang and Lapata, 2017). The original and simplified sentences can be assumed as two different languages. This approach is a process of translating the original language into a simplified one. This is also known as monolingual machine translation. The data used in these studies are sourced from the Simple English Wikipedia corpus (Wubben et al., 2012; Zhu et al., 2010; Kauchak, 2013) and the Newsela corpus (Xu et al., 2015). Although these datasets are composed of simple vocabulary, they do not explicitly incorporate vocabulary restrictions. Our research concentrates on vocabulary restriction, which is different from their research.

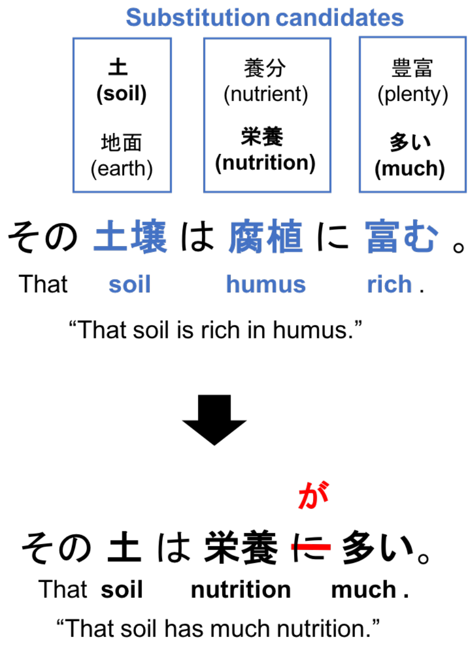

Figure 1: Example of a sentence containing complex words, and the words' substituted candidates. In Japanese, substituting words surrounding a complex word together with the complex word is sometimes necessary.

Lexical simplification systems simplify texts mainly by substituting complex words with simpler alternatives (Paetzold and Specia, 2016; Paetzold and Specia, 2017). Lexical simplification involves the following processes: identification of complex words, generating synonyms or similar phrases by using various similarity measures and ranking as well as selecting the best candidate word. Kajiwara and Yamamoto (2015) used several Japanese paraphrasing datasets for generating substitutions. Hading et al. (2016) also utilised Japanese thesaurus and dependency-based word embeddings. In Japanese, substituting particles surrounding a verb is sometimes necessary when the verb is substituted. In these studies, only one complex word is substituted; hence, lexical substitution accompanied by the replacement of particles cannot be performed. In Figure 1, the complex word "富む (rich)" is substituted by "多い (much)". However, "その土は栄養に多い" is not fluent. When replacing "富む (rich)" with

"多い" (much)", the particle "に" must be substituted by"が" together with "富む (rich)". The problem as to why a substitution candidate does not fit the context is also described by Hading et al. (2016) and Kodaira et al. (2016). To address this issue, we use a simple paraphrasing dictionary that covers alternatives for words surrounding the complex word in the substitution generation step.

3 Methods

3.1 Machine translation approach

We implemented the encoder-decoder model in (Nisioi et al., 2017). We defined the architecture into two LSTM layers (Hochreiter and Schmidhuber, 1997), 500 hidden units and 0.3 dropout probability (Srivastava et al., 2014). The vocabulary size is defined to 20,000 . We trained the model for 15 epochs by using Adam optimiser (Kingma and Ba, 2015). Further, we saved the current state of the model and predicted the perplexity of the model on the development set at the end of each epoch. We selected the model resulting from the epoch with the best perplexity to avoid over-fitting. Additionally, we employed global attention and input feeding described by Luong et al. (2015) for the decoder. The architecture is illustrated in Figure 2.

For the attention layer, we compute the context vector c_t by using the information obtained from the hidden states of the source sentence, and by computing a weighted average with the alignment weights a_t. The new hidden state $\tilde{h_t}$ is computed using a concatenation of the previous hidden state h_t and the context vector c_t.

$$\tilde{h_t} = \tanh W[c_t; h_t] \tag{1}$$

The global alignment weights are computed using a softmax function over the general scoring method for attention.

$$a_t(s) = \frac{\exp(h_t^{\mathrm{T}} W_a \bar{h_s})}{\sum_{s'} \exp(h_t^{\mathrm{T}} W_{as'} \bar{h_s})} \tag{2}$$

Input feeding is a process that sends the previous hidden state, obtained by using the alignment method, to the input at the subsequent step.

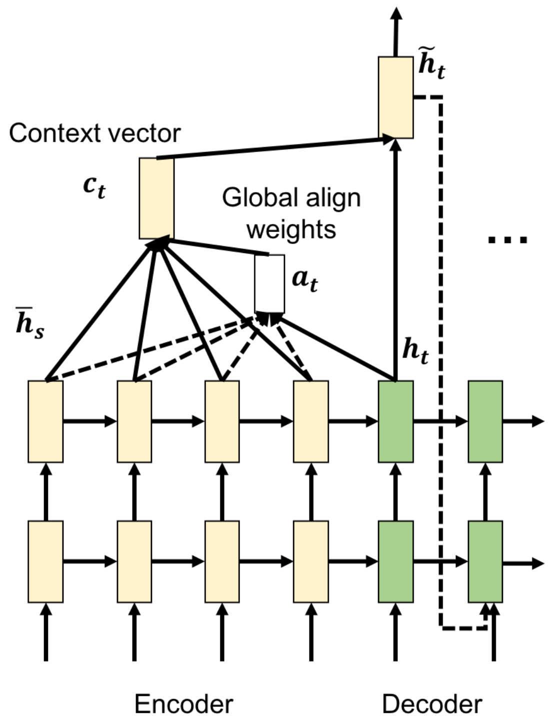

Figure 2: Architecture of model

3.2 Context-based lexical substitution

Lexical simplification is generally performed using the following four steps: complex word identification, substitution generation and substitution selection and ranking (Paetzold and Specia, 2015). Our definition of simplicity is based on the inclusion of the core vocabulary in the simplified text. In the complex word identification step, we checked whether all words in the input sentences exist in the core vocabulary list. We defined words other than in the core vocabulary as complex words. In the substitution generation step, we generated the substitution candidates from the Japanese simplified dictionary as described in section 4.1. We also added a copy of the original word into the substitution candidates because sometimes all candidates generated by the dictionary do not fit the context. In the substitution selection and ranking steps, we ranked the candidates generated from the substitution generation step based on various features, such as fluency (section 3.2.1), semantic similarity (section 3.2.2), contextual similarity (section 3.2.3) and edit distance (section 3.2.4). Then, we selected the best candidate

Algorithm 1 Ranking(ts)

1: $subs \leftarrow \emptyset$
2: **for** $each\ token\ t \in ts$ **do**
3: $all\_rank \leftarrow \emptyset$
4: $cands \leftarrow obtain\_candadates(t)$
5: **for** $each\ feature\ f$ **do**
6: $scores \leftarrow \emptyset$
7: **for** $each\ cand \in cands$ **do**
8: $scores \leftarrow scores \cup f(cand)$
9: **end for**
10: $rank \leftarrow ranking(score)$
11: $all\_rank \leftarrow all\_rank \cup rank$
12: **end for**
13: $wavg\_rank \leftarrow weighted\_avg(all\_rank)$
14: $best\_cand \leftarrow argmax_{cands}(wavg\_rank)$
15: $subs \leftarrow subs \cup (t, best\_cands)$
16: **end for**
17: **return** subs

with regard to the ranking results obtained.

We employed the ranking algorithm of Glavaš and Štajner (2015). The overall simplification algorithm is provided in Algorithm 1. First, we obtained the substitution candidates for each complex word in the original sentences (line 4). Then, we computed the features for each of the substitution candidates (lines 5-9), and ranked them according to feature scores (line 10). Finally, we chose the best candidate as the one with the highest weighted average rank in terms of the overall features (line 14). When calculating the weighted average, we empirically defined the fluency weight to 0.6, semantic similarity weight to 0.1, context similarity weight to 0.2 and edit distance weight to 0.1.

3.2.1 Fluency

For fluency calculation, we employed perplexity obtained from bi-directional language model (BiLM). This model combines forward and backward language models.

Given a sequence of N tokens (t_1, t_2, ..., t_N), a forward LM computes the probability of the sequence. The model predicts the token t_k considering the past sequence (t_1, ..., t_{k-1}).

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, ..., t_{k-1}) \quad (3)$$

A backward LM is similar to a forward LM except that it is provided with the reversed input of a sequence. That is, this model predicts the previous token t_k considering the future sequence (t_{k+1}, ..., t_N).

$$p(t_1, t_2, ..., t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, ..., t_N) \quad (4)$$

BiLM is trained to maximise the following log-likelihood which combines the forward and backward directions.

$$\sum_{k=1}^{N} \{\log p(t_k | t_1, t_2, ..., t_{k-1}; \overrightarrow{\theta}_{fwdLM}) + \log p(t_k | t_{k+1}, t_{k+2}, ..., t_N; \overleftarrow{\theta}_{bwdLM})\} \quad (5)$$

, where $\overrightarrow{\theta}_{fwdLM}$ and $\overleftarrow{\theta}_{bwdLM}$ are hyperparameters of forward and backward LMs, respectively. We defined the architecture as two LSTM layers with 256 hidden units and 0.2 dropout probability. This model is trained on Balanced Corpus of Contemporary Written Japanese [3] (BCCWJ).

3.2.2 Semantic similarity

This feature is computed as the cosine similarity between the vector of the original word and that of the substitution candidate. To obtain a word vector, we employed NWJC2vec (Asahara, 2018), which is a large-scale pre-trained word embedding constructed from NINJAL Web Japanese Corpus (Asahara et al., 2014).

3.2.3 Contextual similarity

As semantic similarity does not distinguish the senses of polysemous words, considering only the semantic similarity between the original and candidate word may lead to selecting a synonym of the wrong sense. Therefore, we compute this feature by averaging the semantic similarities between a substitution candidate and each word based on the context of the original word.

[3] BCCWJ , which is a corpus of various Japanese texts, including books, magazines, newspapers, white papers, blogs, net bulletin boards, textbooks and law.

	Maruyama and Yamamoto (2018)		Katsuta and Yamamoto (2018)	
	Original	Simplified	Original	Simplified
Total #sentences	50,000	50,000	35,000	35,000
Total #tokens	490,021	516,881	434,544	504,850
Vocabulary size	8,786	2,238	16,587	3,939
Avg. #characters per sentence	14.79	15.35	19.18	21.46
Avg. #words per sentence	9.80	10.34	12.42	14.42

Table 1: Corpora statistics. We presented the number of words in the vocabulary after changing them to the basic form based on the UniDic dictionary. This vocabulary size also includes words, such as proper nouns and symbols. Therefore, the vocabulary size of the simplified version is more than 2,000 words.

$$csim(\omega, c) = \frac{1}{|C(\omega)|} \sum_{\omega' \in C(\omega)} cos(v_{\omega'}, v_{\omega}) \quad (6)$$

,

where $C(\omega)$ is a set of context words of the original word ω and v_ω is the NWJC2vec vector of the word ω. We defined the size of the context window to three around the original word.

3.2.4 Edit distance

We added a copy of the original word into the substitution candidates. When ranking according to the fluency or the semantic and context similarities, the output that does not change the original word becomes more advantageous. Therefore, we added the edit distance between the original and simplified sentences into a ranking feature to adopt the simplified sentences preferentially.

4 Experiments

4.1 Datasets

Japanese Simplified Corpus. Japanese language has two simplification corpora, i.e. the simplified Japanese parallel corpus (Maruyama and Yamamoto, 2018) and the crowdsourced corpus (Katsuta and Yamamoto, 2018). The simplified Japanese parallel corpus is constructed by five native Japanese speakers. Maruyama and Yamamoto (2018) asked the annotators to simplify a part of the Tanaka corpus[4] with 2,000-word vocabulary[5] as selected by them. These 2,000 words and named entities cover approximately 80% of BCCWJ. In addition, their corpora have high scores in fluency and meaning preservation based on a manual evaluation. That is, their core vocabulary can cover a wide range of expressions in Japanese. The crowdsourced corpus is constructed by seven annotators employed via crowdsourcing. Katsuta and Yamamoto (2018) asked the annotators to simplify by using only the core vocabulary. The statistics of the two corpora are shown in Table 1. We used 85,000 sentences consisting of the simplified Japanese parallel corpus and the crowdsourced corpus for training the machine translation model, and for conducting an evaluation.

To evaluate the model's effectiveness on rare words, we divided the simplified corpus into two datasets: a dataset containing rare words (RARE) and a dataset that does not contain rare words (NORMAL). Here, we define rare words as word sets with text coverage of less than 3% in the corpus. Each rare word appears five times or less in the simplified corpus. We randomly selected 2,000 sentences from the RARE dataset and the 2,000 sentences from the NORMAL dataset for evaluation. Development data were also constructed in the same manner. We used other 77,000 sentences as the training data for the machine translation approach.

Japanese Simplified Dictionary. Pavlick and Callison-Burch (2016) constructed a simple paraphrasing database (simple PPDB) for text simplification. Thereafter, Kajiwara and Komachi (2017) constructed a simple PPDB for Japanese language. This dataset extracts complex and simple word pairs from the paraphrasing database for Japanese (PPDB:

[4]http://www.edrdg.org/wiki/index.php/Tanaka_Corpus

[5]http://box.jnlp.org/easy-japanese/words2

Example of sentence	Original	Simplified
彼女は私に、お腹がすいていると ささやい た。	ささやい た (whispered)	小さな 声 で 言っ た (talked in a small voice)
(She whispered to me that she was hungry.)	ささやいた (whispered)	言った (talked)
ひき逃げの犯人は翌日父親に 伴わ れて自首してきました。	伴わ れ て (be accompanied)	連れ られ て (be followed)
(The hit-and-run culprit surrendered himself to the police accompanied by his father the following day.)	に 伴わ れ て (be accompanied by)	と 一緒 に (together with)
彼がその事実を否定したというのは はたして 本当かしら。	はたして (really)	実際 に (actually)
(I really wonder that he denies the truth.)	はたして (really)	#deletion
自分の才能を思う存分生かすには、自分にもっと ふさわしい 職業、新しい職場を見つけることです。	ふさわしい (suitable)	適切 な (appropriate)
In order to utilize the most of your talent, it is necessary to find a more suitable job and a new workplace.	ふさわしい (suitable)	合っ た (fit)

Table 2: Examples of Japanese simplified dictionary. Column 1 indicates an example of a sentence that includes a complex word (underlined). Column 2 indicates the words to be rewritten in the example of a sentence by the annotators. Column 3 indicates the words which the annotators simplified from Column 2. “#deletion” denotes that the target word is deleted

Japanese). The simple PPDB is attached to paraphrasing probability and three-level word simplicity based on the education vocabulary table for Japanese. As a handcrafted paraphrasing dictionary for text simplification, there are lexical substitution datasets constructed by Kajiwara and Yamamoto (2015) and evaluation datasets constructed by Kodaira et al. (2016) for lexical simplification. However, these datasets cannot be applied to our task because they do not contain vocabulary restriction.

Hence, we manually constructed a paraphrase dictionary where the simplified side consists of only the core vocabulary. We defined the target words to 20,000 high-frequency words in BCCWJ and Tanaka corpus. These words do not include proper nouns, onomatopoeia, numbers and symbols. We asked seven annotators who employed crowdsourcing to simplify these 20,000 words. We divided the 20,000 words into four files such that each file contains 5,000 words, and assigned one file to each annotator. We presented each annotator with the target words, their parts of speech and examples of sentences containing the target words. The annotators listed similar words or phrases of the target words based on the above information. We allowed the annotators to simplify not only the target words but also the surrounding words. In addition, we allowed the deletion of target words if the action renders fluent and simple expressions compared to substituting it with simpler alternatives. An example of our constructed paraphrase dictionary is shown in Table 2.

4.2 Evaluation metrics

We evaluated the model’s output based on three metrics:1) BLEU with NIST smoothing described by Chen and Cherry (2014), which is a traditional machine translation evaluation metric; 2) SARI (Xu et al., 2016), which is a recent text-simplification metric; and 3) core vocabulary ratio (Core) contained in a sentence, which is considered to be a characteristic of simplicity in this study. SARI has a positive correlation with simplicity, and BLEU has a positive

Fluency	
4	The sentence is clear.
3	The sentence has several unclear parts, but the overall meaning can be understood.
2	Although a sentence is not clear, its meaning can be inferred.
1	The sentence is not clear and its meaning cannot be understood totally.
Adequency	
4	The meanings of the two sentences are the same.
3	The main meanings of the two sentences are the same.
2	The meanings of the two sentences are different, but they partially match.
1	The meanings of the two sentences are different.

Table 3: Criteria for fluency score and meaning preservation score

correlation with fluency and meaning preservation (Vu et al., 2018).

In addition, we manually evaluated the randomly selected 50 sentences from the evaluation data of the RARE and NORMAL datasets related to fluency and adequacy. We defined the evaluation criteria to four stages (i.e. higher marks indicate better output) based on the criteria shown in Table 3. We also counted the total number of changes (Total) generated by each system, considering the change of phrase as one change. Instances involving deletion of complex words are considered as a single change. The changes that preserve the original meaning and render the sentences to be more simpler to understand are marked as *Correct*.

5 Result and Discussion

The results of the automatic and manual evaluations are shown in Tables 4 and 5, respectively. In the tables, "Original" denotes the input sentences and "EncDecAttn" denotes the model of the machine translation approach described in Section 3.1. Furthermore, "EncDec" denotes the encoder-decoder model which does not use attention mechanism and input-feeding. This model corresponds to the model used by Maruyama and Yamamoto (2017), and its hyperparameters are defined to the same hyperparameters as the "EncDecAttn" model. Finally, "LexSub" denotes the lexical substitution model as described in Section 3.2.

5.1 Robustness of the lexical substitution model

Table 4 illustrates that the "EncDecAttn" has the highest scores of BLEU and SARI in the NORMAL dataset. Table 5 illustrates that "EncDecAttn" outputs clear sentences sufficient to understand and preserve the main meaning of the original sentence. Moreover, "EncDecAttn" performs a large number of changes with very few mistakes.Hence, "EncDecAttn" is the best model with regard to the NORMAL dataset.

However, in the RARE dataset, the fluency and adequacy scores of the "EncDecAttn" are low, i.e. 2.57 and 2.14, respectively. These scores indicate that the outputs of "EncDecAttn" are not clear and cannot preserve the meaning of the original sentence. Therefore, the RARE dataset is a critical defect for practical text simplification systems to generate a sentence that differ from the meaning of the original sentence. Although the model output contains many changes, but the ratio of correct changes is low, i.e. 31.7%. These results convey that "EncDecAttn" does not work for data containing rare words.

Meanwhile, "LexSub" has the highest BLEU score in the RARE dataset.Moreover, it has the largest ratio of correct changes as well as the highest score in terms of fluency and adequacy with regard to both, NORMAL and RARE datasets. Hence, LexSub works robustly in both the datasets. As mentioned in Section 1, simplifying rare words in the text simplification task is important. The output results of "LexSub" are more apt for a practical system than "EncDec" or "EncDecAttn".

Models	NORMAL			RARE		
	BLEU	SARI	Core	BLEU	SARI	Core
Original	66.70	27.34	90.9%	39.26	21.73	82.1%
EncDec	63.59	40.00	99.4%	30.52	38.54	99.2%
EncDecAttn	73.73	44.79	99.3%	41.10	45.33	98.7%
LexSub	68.21	32.63	93.8%	41.67	32.38	87.0%

Table 4: Automatic evaluation.

Models	NORMAL				RARE			
	Changes		Scores		Changes		Scores	
	Total	Correct	Fluency	Adequacy	Total	Correct	Fluency	Adequacy
EncDec	55	58.2%	3.24	3.00	83	19.3%	2.48	1.90
EncDecAttn	42	73.8%	3.38	3.45	82	31.7%	2.57	2.14
LexSub	22	81.8%	3.68	3.73	26	84.6%	3.82	3.73
Reference	40	100%	3.9	3.84	71	100%	3.82	3.66

Table 5: Manual evaluation

5.2 Few changes in the input sentences of the lexical substitution

The core vocabulary ratio of "LexSub" in the NORMAL and RARE datasets is 93.8% and 87.0%, respectively. "LexSub" simply improves the core vocabulary ratio by approximately three percentage points in the NORMAL dataset and five percentage points in the RARE dataset. In addition,very few changes were witnessed with regard to "LexSub" in both the datasets. Thus, we believe that the candidates that fit the context of the original sentences are not generated. "LexSub" obtains only three candidates at the maximum for one complex word from the paraphrasing dictionary described in 4.1. If our simplified dictionary is extended in the future, then this problem can be addressed. These results imply that steadily accumulating human knowledge is important.

6 Conclusions and Future work

We attempted to generate simplified sentences by using the concept of core vocabulary whilst preserving the original meaning of the sentences in the text simplification subtask.

To explore a simplification model that works robustly for sentences containing rare words, we employed two methods: machine translation approach and lexical substitution. The substitution model generates candidates based on a paraphrasing dictionary that covers words surrounding complex words. The results show that this approach presents the best performance overall, but it is not suitable for sentences that contain rare words. Meanwhile, lexical substitution model can robustly simplify sentences, even those that include rare words as well.

This model can accurately simplify complex words in both the datasets, and the number of changes encountered is the lowest. This is because the model obtains few substitution candidates for one complex word from the paraphrasing dictionary.

We believe that for practical text simplifications, accumulating human knowledge is important, although its construction is costly. In future, we would increase the number of substitution candidates. Moreover, we would construct a robust model by combining the machine translation approach and the lexical substitution method based on human knowledge.

Acknowledgements

We are grateful to JSPS KAKENHI Grants-in-Aid for Challenging Research (Exploratory) Grant ID 17K18481 for their support with regard to conducting this study.

References

Masayuki Asahara, Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi. 2014. Archiving and analysing techniques of the ultra-large-scale web-based corpus project of ninjal, japan. *Alexandria: The Journal of National and International Library and Information Issues*, 25(1-2):129–148.

Masayuki Asahara. 2018. NWJC2Vec: Word embedding dataset from 'NINJAL Web Japanese Corpus'. *Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication*, 24(2):7–25, Feb.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods tbr text simplification. *Proceedings of the 16th International Conference on Computational Linguistics*, pages 1041–1044.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.

Han-Bin Chen, Hen-Hsen Huang, Hsin-Hsi Chen, and Ching-Ting Tan. 2012. A simplification: Translation-restoration framework for cross-domain smt applications. *24th International Conference on Computational Linguistics: Technical Papers*, pages 545–560.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*.

Muhaimin Hading, Yuji Matsumoto, and Maki Sakamoto. 2016. Japanese lexical simplification for non-native speakers. *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Kazunari Iwata. 2010. The preference for english in linguistic services: Japanese for living: Country wide survey and hiroshima. *The Japanese Journal Language of in Society*, pages 81–94.

Tomoyuki Kajiwara and Mamoru Komachi. 2017. Simple ppdb: Japanese. *Proceedings of the 23rd Natural Language Processing of Japan*.

Kajiwara Kajiwara and Kazuhide Yamamoto. 2015. Evaluation dataset and system for japanese lexical simplification. *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop*, pages 35–40.

Akihiro Katsuta and Kazuhide Yamamoto. 2018. Crowdsourced corpus of sentence simplification with core vocabulary. *The 11th International Conference on Language Resources and Evaluation*, pages 461–466.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

P. Diederik Kingma and Lei Jimmy Ba. 2015. Adam: A method for stochastic optimization. *Proceeding of the 3rd International Conference for Learning Representations*.

Tomonori Kodaira, Tomoyuki Kajiwara, and Mamoru Komachi. 2016. Controlled and balanced dataset for japanese lexical simplification. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics – Student Research Workshop*, pages 1–7.

Batia Laufer. 1992. How much lexisis necessary for reading comprehension? *Vocabulary and Applied Linguistics*, pages 126–132.

John Lee, Buddhika J. K, and Pathirage Don. 2017. Splitting complex english sentences. *Proceedings of the 15th International Conference on Parsing Technologies*.

Minh-Thang Luong, Hieu Pham, and D. Christopher Manning. 2015. Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Takumi Maruyama and Kazuhide Yamamoto. 2017. Sentence simplification with core vocabulary. *Proceedings of the International Conference on Asian Language Processing*, pages 363–366.

Takumi Maruyama and Kazuhide Yamamoto. 2018. Simplified corpus with core vocabulary. *The 11th International Conference on Language Resources and Evaluation*, pages 1153–1160.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 788–796.

I.S.P Nation. 2006. How large a vocabulary is needed for reading and listening. *The Canadian Modern Language Review/La Revue canadienne des langues vivantes*.

Sergiu Nisioi, Sanja Štajner, Ponzetto Simone Paolo, and P. Liviu Dinu. 2017. Exploring neural text simplification models. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 85–91.

Henrique Gustavo Paetzold and Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. *Proceedings of ACL-IJCNLP 2015 System Demonstrations*.

Henrique Gustavo Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3761–3767.

Henrique Gustavo Paetzold and Lucia Specia. 2017. Lexical simplification with neural ranking. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple ppdb: A paraphrase database for simplification. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Advaith Siddharthan and A. M. Angrosh. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. *Proceedings of the 20th international conference on Computational Linguistics*, pages 86–91.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15 (2014)*, pages 1929–1958.

Sanja Štajner and Maja Popvić. 2016. Can text simplification help machine translation? *Baltic J. Modern Computing*, 4(2).

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. Sentence simplification with memory-augmented neural networks. *Proceedings of NAACL-HLT 2018*, pages 79–85.

Sander Wubben, den Antal van Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

Wei Xu and Ralph Grishman. 2009. A parse-and-trim approach with information significance for chinese sentence compression. *Proceedings of the 2009 Workshop on Language Generation and Summarisation*, pages 48–55.

Wei Xu, Callison Chris Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361.