Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 3114–3120
Marseille, 11–16 May 2020

Word Complexity Estimation for Japanese Lexical Simplification

Daiki Nishihara[†], **Tomoyuki Kajiwara**[‡]
[†] Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
[‡] Institute for Datability Science, Osaka University, Osaka, Japan
nishihara.daiki@ist.osaka-u.ac.jp, kajiwara@ids.osaka-u.ac.jp

Abstract

We introduce three language resources for Japanese lexical simplification: 1) a large-scale word complexity lexicon, 2) the first synonym lexicon for converting complex words to simpler ones, and 3) the first toolkit for developing and benchmarking Japanese lexical simplification system. Our word complexity lexicon is expanded to a broader vocabulary using a classifier trained on a small, high-quality word complexity lexicon created by Japanese language teachers. Based on this word complexity estimator, we extracted simplified word pairs from a large-scale synonym lexicon and constructed a simplified synonym lexicon useful for lexical simplification. In addition, we developed a Python library that implements automatic evaluation and key methods in each subtask to ease the construction of a lexical simplification pipeline. Experimental results show that the proposed method based on our lexicon achieves the highest performance of Japanese lexical simplification. The current lexical simplification is mainly studied in English, which is rich in language resources such as lexicons and toolkits. The language resources constructed in this study will help advance the lexical simplification system in Japanese.

Keywords: Lexical simplification, word complexity, lexicon

1. Introduction

Text simplification (Shardlow, 2014) that rewrites a sentence into an easy-to-understand form for language learners (Petersen and Ostendorf, 2007) and children (Belder and Moens, 2010) is attracting attention. In particular, lexical simplification (Paetzold and Specia, 2017a), which paraphrases complex words into simpler ones according to the context while maintaining the syntactic structure of the input sentence, is being actively researched (mainly in English). The lexical simplification system is useful not only for assisting the reading comprehension of language learners and children but also for the preprocessing of natural language processing applications such as machine translation (Štajner and Popović, 2016).

Lexical simplification has been studied mainly in English, which is rich in language resources such as a word complexity lexicon (Maddela and Xu, 2018), a paraphrase database from complex phrases to simpler ones (Pavlick and Callison-Burch, 2016) and a toolkit (Paetzold and Specia, 2015). Such language resources are not available for other languages, including Japanese.

Lexical simplification consists of the following four subtasks (Shardlow, 2014; Paetzold and Specia, 2017a):

Complex word identification: Task of deciding which words of a given sentence may not be understood by a given target audience and hence must be simplified.

Substitution generation: Task of finding words or expressions that could replace the target complex word.

Substitution selection: Task of deciding which of the generated candidate substitutions can replace the complex word without compromising the sentence's grammar or meaning in a given context.

Substitution ranking: Task of ranking the remaining candidate substitutions of a given complex word by their simplicity.

As shown in Figure 1, these subtasks can be broadly divided into word complexity estimation tasks and lexical substitution tasks. This study focuses on word complexity estimation tasks and constructs language resources for Japanese lexical simplification.[1] Our contributions are the construction of the following three language resources in Japanese:

- a large-scale word complexity lexicon,
- a simplified synonym lexicon from complex words into simpler ones,
- and a toolkit for developing and benchmarking a lexical simplification system.

2. Related Work

This section outlines the lexica used for lexical simplification and the main approaches in lexical simplification.

2.1. Lexica for Lexical Simplification

Simple PPDB (Pavlick and Callison-Burch, 2016), a subset of a large-scale paraphrase database (Ganitkevitch et al., 2013; Pavlick et al., 2015), is a language resource that extracts synonymous phrase pairs from complex phrases into simpler ones for lexical simplification. Pavlick and Callison-Burch (2016) trained a classifier to estimate which phrase in paraphrase pair is simpler based on a logistic regression model with features such as number of characters, number of words, frequency, part of speech, and word embeddings. 4.5 million simplified synonymous phrase pairs[2] are publicly available which were extracted by applying this complexity estimator to the paraphrase database.[3]

[1] https://github.com/nishihara-daiki/lsj/tree/lrec2020
[2] http://www.seas.upenn.edu/~nlp/resources/simple-ppdb.tgz
[3] http://paraphrase.org

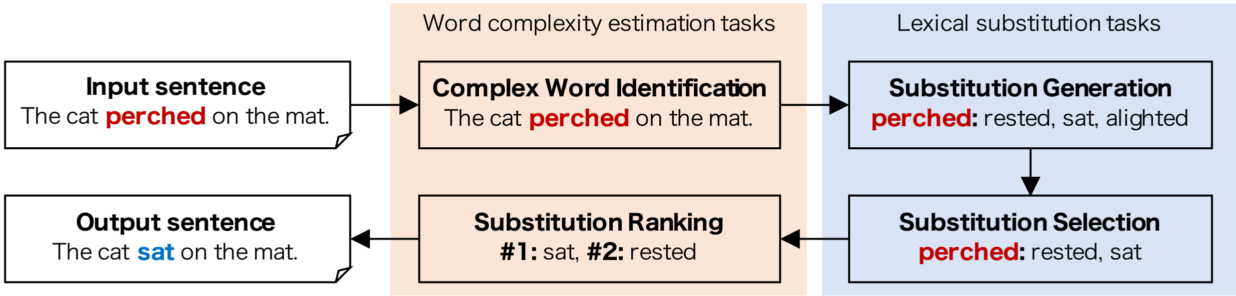

Figure 1: Lexical simplification pipeline.

Maddela and Xu (2018) built a word complexity lexicon[4] where 11 non-native English speakers annotated 6 levels of complexity for the most frequent $15,000$ words in the Google 1T Ngram Corpus.[5] In addition, they trained a neural network-based complexity estimator with features adding word complexity to Pavlick and Callison-Burch (2016)'s features, and published a Simple PPDB++, a large-scale simplified paraphrase database that has more than 10 million pairs.

In Japanese, word complexity lexica such as Vocabulary List of Japanese Language Proficiency Test (JLPT)[6] and Japanese Education Vocabulary List (JEV)[7] are available. JLPT contains approximately $8k$ words with 4 levels of complexity and JEV has almost $18k$ words with 6 levels of complexity (Easy (first / second part), Medium (first / second part), Difficult (first / second part)). We train a word complexity estimator using JEV, which has a larger vocabulary and extend it to a larger-scale word complexity lexicon.

2.2. Lexical Simplification Approaches

Early studies of lexical simplification (Devlin and Tait, 1998; Belder and Moens, 2010) obtained synonyms of the target word from WordNet (Miller, 1995) and replaced the most frequently used word with the target word. Many subsequent systems consist of similar pipelines: a candidate acquisition step (lexical substitution task in Figure 1) and a ranking step (word complexity estimation task in Figure 1).[8]

Approaches for acquiring substitution candidates include lexicon-based methods and distributional similarity-based methods. Devlin and Tait (1998), a pioneer in lexicon-based methods, has acquired synonyms from WordNet. Recent studies (Pavlick and Callison-Burch, 2016; Maddela and Xu, 2018) uses Simple PPDB, which can handle phrases as well as words. Biran et al. (2011) used distributional similarity for the first time to select semantically close substitutions from synonyms obtained from WordNet. Recent studies (Glavaš and Štajner, 2015; Paetzold and Specia, 2016) uses word embeddings such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014).

Approaches for ranking substitution candidates include frequency-based methods and average ranking-based methods. Devlin and Tait (1998), a pioneer in frequency-based methods, ranked substitutions by counting the 1-gram frequencies. A recent study (Paetzold and Specia, 2016) uses the 5-gram frequency that considers context words around the target word. Glavaš and Štajner (2015) proposed an average ranking approach that ranks substitutions by word similarity, context similarity, 1-gram frequency, and 5-gram frequency.

In English, a toolkit (Paetzold and Specia, 2015) for developing and benchmarking a lexical simplification system is available that implements each of these methods.[9] In this study, we release a Japanese version of the toolkit based on an evaluation dataset (Kodaira et al., 2016) for Japanese lexical simplification.

3. Constructing a Word Complexity Lexicon

The Japanese Educational Vocabulary List (JEV)[7] is a high-quality word complexity lexicon constructed by Japanese language teachers. Because it is constructed manually, however, the number of recorded vocabularies is small. We construct a large-scale Japanese word complexity lexicon based on JEV for use in natural language processing applications such as lexical simplification.

3.1. Proposed Method

In order to construct a large-scale Japanese word complexity lexicon, we train a multi-class classifier that estimates the three levels of complexity (Easy, Medium, and Difficult) in JEV. We use four types of features: part of speech, character frequency, word frequency, and word embedding. For each word, we obtain character frequency features by measuring both maximum and minimum values of each character's frequency in the word.

The number of characters used in previous studies in English (Pavlick and Callison-Burch, 2016; Maddela and Xu, 2018) did not contribute to our Japanese word complexity estimation task. This is due to the fact that the word in Japanese is composed of three types of characters: Hiragana, Katakana, and Kanji. The phonetic characters Hiragana and Katakana are simple characters composed of 48 types, but the ideographic characters Kanji are complex characters composed of tens of thousands of types. Therefore, a word composed of Hiragana or

[4]https://github.com/mounicam/lexical_simplification
[5]https://catalog.ldc.upenn.edu/LDC2006T13
[6]http://www7a.biglobe.ne.jp/nifongo/data/
[7]http://jhlee.sakura.ne.jp/JEV.html
[8]Since the target words are given in the evaluation dataset, many methods do not implement the CWI subtask in Figure 1.
[9]https://github.com/ghpaetzold/LEXenstein

	Easy	Medium	Difficult	Total
Training	662	$5,255$	$4,641$	$10,558$
Development	61	667	591	$1,319$
Test	80	642	597	$1,319$
OOV	413	$2,201$	$2,110$	$4,724$
Total	$1,216$	$8,765$	$7,939$	$17,920$

Table 1: The number of words in JEV. Unused "OOV" are words that do not appear in at least one of these sources: Wikipedia, TWC, BCCWJ, or Asahi Shimbun word vector.

Katakana characters is an easy word even if it has a large number of characters; on the contrary, a word composed of Kanji characters may be a difficult word even if it has a small number of characters.

3.2. Experimental Settings

Japanese Wikipedia[10] was used to count the frequency of characters and words. We used WikiExtractor[11] to extract text from Wikipedia, and MeCab (Kudo et al., 2004) with IPADIC-2.7.0 for tokenization. In addition to Wikipedia, we used the word frequencies of the Tsukuba Web Corpus (TWC)[12] and the Balanced Corpus of Contemporary Written Japanese (BCCWJ)[13] (Maekawa et al., 2010) because counting word frequencies from multiple corpora improves the estimation of word complexity (Kajiwara and Komachi, 2018). Since the BCCWJ tags each word with a part-of-speech, we use it as a part-of-speech feature. We used a 300-dimensional Skip-gram model (Mikolov et al., 2013) of the Asahi Shimbun word vector[14] as our word embedding feature. This word embedding model is trained on approximately 8 million newspaper articles. Note that the performance of the Skip-gram model was better than the CBOW model (Mikolov et al., 2013) and the GloVe model (Pennington et al., 2014) included with the Skip-gram model.

As shown in Table 1, JEV was divided into a training set, development set and test set at a ratio of 8:1:1 for our experiment. Note that there are fewer easy words compared to medium and difficult words.

The SVM[15] with RBF kernel was used for our classifier, and its C and γ parameters were selected from $\{0.001, 0.01, 0.1, 1.0, 10.0\}$ to achieve the best performance in the development set. Note that the SVM classifier is better than the random forest classifier and the multi-layer perceptron for our experiments.

3.3. Experimental Results

Table 2 shows the classification accuracy of word complexity estimation on the test set. Our method, which considers four types of features, achieved higher accuracy than the baseline and the comparison methods, which considers only each feature. Our ablation analysis in the lower part of Table 2 shows that all the features contribute to the performance improvement as the performance decreases even if a feature is removed.

Features	Accuracy
Baseline: the most frequent class	0.487
Comparison: part-of-speech (POS)	0.516
Comparison: character frequency (CF)	0.549
Comparison: word frequency (WF)	0.646
Comparison: word embedding (WE)	0.703
Ours: POS+CF+WF+WE	**0.763**
Ours w/o POS	0.757
Ours w/o CF	0.735
Ours w/o WF	0.716
Ours w/o WE	0.671

Table 2: Accuracy for word complexity estimation.

Words	Complexity
先生 (teacher)	Easy
信用 (confidence)	Medium
胎盤 (placenta)	Difficult

Table 3: Examples of word complexity lexicon.

3.4. Word Complexity Lexicon for Japanese

We estimated the word complexity of $40,605$ words that are shared vocabularies of Wikipedia, TWC, BCCWJ, and Asahi Shimbun word vector; these sources were used for feature extraction in our experiment and constructed a word complexity lexicon was constructed. Table 3 shows examples of the word complexity lexicon.

4. Constructing a Simplified Synonym Lexicon for Lexical Simplification

Following the simple PPDB (Pavlick and Nenkova, 2015; Pavlick and Callison-Burch, 2016), we automatically construct a simplified synonym lexicon for lexical simplification that extracts complex-to-simple synonyms from a large-scale paraphrase database. In this study, we only convert from complex words to simpler synonyms because our complexity estimator is for words. Estimating a phrasal complexity is reserved for future work.

4.1. Proposed Method

We construct a synonym lexicon for lexical simplification by translating synonym pairs from complex words into simpler ones found in the paraphrase database. In order to identify complex and simple word pairs, each word pair is classified into the following three classes: "a destination word is simpler", "a source word is simpler", or "a word pair has the same complexity". We propose two approaches, pointwise and pairwise methods, to extract synonym pairs from complex words into simpler ones.

[10]https://dumps.wikimedia.org/jawiki/20190801/
[11]https://github.com/attardi/wikiextractor
[12]http://nlt.tsukuba.lagoinst.info/
[13]https://pj.ninjal.ac.jp/corpus_center/bccwj/freq-list.html
[14]https://cl.asahi.com/api_data/wordembedding.html
[15]https://scikit-learn.org/stable/index.html

	A destination word is simpler				A source word is simpler				The same complexity				
	D→M	D→E	M→E	Subtotal	M→D	E→M	E→D	Subtotal	E→E	M→M	D→D	Subtotal	Total
Train	2,668	308	357	3,333	2,638	366	329	3,333	28	1,867	1,439	3,334	10,000
Dev	279	21	33	333	275	30	28	333	2	181	151	334	1,000
Test	269	28	36	333	262	36	35	333	5	167	162	334	1,000
Total	3,216	357	426	3,999	3,175	432	392	3,999	35	2,215	1,752	4,002	12,000

Table 4: The number of word pairs sampled from JEV by complexity: **E**asy, **M**edium and **D**ifficult.

Features	Accuracy
Baseline: the most frequent class	0.334
Comparison: pairwise (POS)	0.418
Comparison: pairwise (CF)	0.435
Comparison: pairwise (WF)	0.535
Comparison: pairwise (WE)	0.540
Ours A: pointwise	**0.616**
Ours B: pairwise (POS+CF+WF+WE)	0.610
Ours C: Ours B with difference features	0.599
Ours B w/o POS	0.607
Ours B w/o CF	0.593
Ours B w/o WF	0.522
Ours B w/o WE	0.549

Table 5: Accuracy for word pair complexity estimation.

In the pointwise method, we assign a complexity level to each word using our word complexity lexicon constructed in Section 3.4. Word pairs with one of the following relationships are classified as "a destination word is simpler": Difficult → Medium, Difficult → Easy, or Medium → Easy. Similarly, word pairs classified as "a source word is simple" have one of these relationships: Medium → Difficult, Easy → Medium, or Easy → Difficult. All other pairs are classified as "a word pair has the same complexity". That is, the pointwise method (Ours A) does not train a classifier.

For the pairwise method, we train a multi-class classifier using four types of features: part-of-speech, character frequency, word frequency, and word embedding, similar to Section 3.1 We consider two methods of feature extraction: 1) concatenating word features (Ours B), and 2) considering the features of the differences (Ours C). We use the difference of word frequency and the difference of word embeddings for each dimension as our difference features.

4.2. Experimental Settings

The experimental settings for features and classifiers are the same as in Section 3.2 For training the classifiers, as shown in Table 4, word pairs belonging to the class of "a destination word is simpler", "a source word is simpler", or "a word pair has the same complexity" were randomly sampled from JEV at a ratio of 1:1:1. Note that these word pairs are not synonyms (i.e., the sample includes pairs of words with different meanings). We extracted 10k, 1k, and 1k word pairs for the training set, development set, and test set, respectively.

Word pairs	Complexity
講評 → レビュー (Review)	Difficult → Medium
頸部 → 首 (Neck)	Difficult → Easy
食塩 → 塩 (Salt)	Medium → Easy

Table 6: Examples of simplified synonym lexicon.

4.3. Experimental Results

Table 5 shows the classification accuracy for complexity estimation of word pairs in the test set. We expected the pairwise method to perform better than the pointwise method, which considers the information of each word independently, but contrary to our assumptions, the pointwise method achieved the highest performance. In addition, we expected that the difference features which consider the relationship between words work effectively in the pairwise method, but contrary to our assumptions, the difference features did not contribute to the complexity estimation of word pairs. Our ablation analysis in the lower part of Table 5 shows that all features contribute to performance improvement in the pairwise method.

4.4. Simplified Synonym Lexicon for Japanese

A simplified synonym lexicon was constructed from the 10-best version of PPDB:Japanese[16] (Mizukami et al., 2014) using our pointwise method. PPDB:Japanese is a large-scale paraphrase database in Japanese created from bilingual corpus and has $860k$ synonym pairs.

Table 6 shows examples of the lexicon. The simplified synonym pairs extracted by pointwise and pairwise methods without the difference features have $46,815$ and $59,401$ word pairs, respectively.

5. Toolkit for Lexical Simplification

Following previous study (Paetzold and Specia, 2015) in English, this section describes a toolkit that implements typical methods for each lexical simplification subtask shown in Figure 1 for Japanese. This Python library makes it easy to construct a Japanese lexical simplification system.

5.1. Acquiring Substitutions

As mentioned in Section 2.2, approaches for acquiring substitution include lexicon-based methods and distributional similarity-based methods. This toolkit uses either method to acquire substitution candidates.

[16] https://ahcweb01.naist.jp/old/resource/jppdb/

	Candidate acquisition	Ranking	Accuracy	Precision	Changed Proportion
LIGHT-LS	word2vec	Average ranking	11.35	25.93	43.76
Ours	PPDB: pointwise	5-gram language model	17.18	25.36	67.75
Ours	PPDB: pointwise	Average ranking	**18.64**	**27.51**	67.75
Ours	PPDB: pairwise	5-gram language model	14.59	21.33	**68.40**
Ours	PPDB: pairwise	Average ranking	15.40	22.51	**68.40**

Table 7: Evaluation results of Japanese lexical simplification.

To support the lexicon-based approach (Devlin and Tait, 1998; Pavlick and Callison-Burch, 2016; Maddela and Xu, 2018), we can input a text file in the format of the tab-separated values of synonym pairs. In order to improve the precision, we can further refine the substitution candidates using paraphrase probabilities or cosine similarity between word embeddings.

To support the distributional similarity-based approach (Glavaš and Štajner, 2015; Paetzold and Specia, 2016), we can input a text or binary file in the word2vec format of word embeddings. Following previous studies (Glavaš and Štajner, 2015; Paetzold and Specia, 2016), we select the top 10 substitution candidates whose word embeddings are most similar to that of the target word.

5.2. Ranking Substitutions

As mentioned in Section 2.2, approaches for ranking substitution include frequency-based methods and average ranking-based methods. This toolkit uses either method to select a simple synonym.

To support the frequency-based approach (Devlin and Tait, 1998; Belder and Moens, 2010; Specia et al., 2012; Paetzold and Specia, 2016), we can input a text or binary file in the ARPA format of N-gram language model. Following previous study (Paetzold and Specia, 2016), when using the 5-gram language model, for example, two words before and after the target word are considered in the context. It ranks all substitution candidates based on the likelihood of the language model score, and outputs the candidate with the highest likelihood.

To support the average ranking-based method (Glavaš and Štajner, 2015), we can input a word2vec format of word embeddings, an ARPA format of N-gram language model, and a word frequency list in the format of the tab-separated values. Following the previous study (Glavaš and Štajner, 2015), we rank the substitution candidates using the four rankings of word similarity, context similarity, 1-gram and 5-gram frequency. The candidate with the highest average ranking is the generated output.

5.3. Evaluations

We evaluated the performance of each method using an evaluation dataset (Kodaira et al., 2016) for Japanese lexical simplification. This dataset contains one complex word in all $2,010$ sentences extracted from BCCWJ (Maekawa et al., 2010). Five annotators provide an average of 4.3 words of simple substitutions for each complex word. We used three evaluation metrics: accuracy, precision, and changed proportion, which are commonly used in English lexical simplification(Horn et al., 2014; Glavaš and Štajner, 2015; Paetzold and Specia, 2017a).

Accuracy: The ratio with which the highest ranking candidate is not the target word itself and is in the gold-standard.

Precision: The ratio with which the highest ranking candidate is either the target word itself or is in the gold-standard.

Changed Proportion: The ratio with which the highest ranking candidate is not the target word itself.

A comparison method is LIGHT-LS (Glavaš and Štajner, 2015), which can extract features from raw corpus. LIGHT-LS acquires the top 10 words with the highest cosine similarity between word embeddings as substitution candidates, and select the best candidate by average ranking based on four features: word similarity, context similarity, 1-gram and 5-gram frequency. According to a survey paper on lexical simplification (Paetzold and Specia, 2017a), LIGHT-LS achieves the highest accuracy in the English benchmark dataset.

Our method acquired substitution candidates from the simplified synonym lexicon constructed in Section 4.4 We used the 5-gram language model and Glavaš and Štajner (2015)'s average ranking for the candidate ranking.

The experimental settings are the same as in Section 3.2 and Section 4.2 We used the Skip-gram model (Mikolov et al., 2013) of the Asahi Shimbun word vector[14] as word embeddings. We used KenLM (Heafield, 2011) to train a 5-gram language model on Japanese Wikipedia[10] for candidate ranking. The 1-gram frequency was also calculated from Japanese Wikipedia.

Table 7 shows that our method based on the simplified synonym lexicon outperformed LIGHT-LS (Glavaš and Štajner, 2015) in all evaluation metrics. As described in Section 4.3, since the pointwise method achieves higher performance than the pairwise method for estimating word pair complexity, the simplified synonym lexicon constructed using the pointwise method achieves higher precision and accuracy in this experiment. Since LIGHT-LS cannot rank high-frequency words at the top, it often outputs the target words as is.[17] As a result, LIGHT-LS has a low changed proportion.

[17]Because of heuristics to increase precision in Algorithm 1 (Glavaš and Štajner, 2015).

6. Conclusion

We constructed three language resources for Japanese lexical simplification. Experimental results show that our large-scale word complexity lexicon and simplified synonym lexicon contribute to improving the performance of Japanese lexical simplification. In addition, our toolkit helps to build and evaluate a lexical simplification system that consists of substitution candidates' acquisition and ranking.

In future studies, we will expand the target of complexity estimation to multi-word expressions and phrases to enable more diverse simplifications. Additionally, candidate acquisition methods based on word alignment (Horn et al., 2014) on parallel corpora (Maruyama and Yamamoto, 2018; Katsuta and Yamamoto, 2018) and contextualized word embeddings (Qiang et al., 2019) such as BERT (Devlin et al., 2019) and ranking methods based on learning to rank (Horn et al., 2014) and deep learning (Paetzold and Specia, 2017b) will be implemented in our toolkit so that more approaches can be accessed.

Bibliographical References

Belder, J. D. and Moens, M.-F. (2010). Text Simplification for Children. In *Proceedings of the SIGIR Workshop on Accessible Search Systems*, pages 19–26.

Biran, O., Brody, S., and Elhadad, N. (2011). Putting it Simply: a Context-Aware Approach to Lexical Simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501.

Devlin, S. and Tait, J. (1998). The use of a Psycholinguistic Database in the Simplification of Text for Aphasic Readers. *Linguistic Databases*, pages 161–173.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Ganitkevitch, J., Durme, B. V., and Callison-Burch, C. (2013). PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Glavaš, G. and Štajner, S. (2015). Simplifying Lexical Simplification: Do We Need Simplified Corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 63–68.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197.

Horn, C., Manduca, C., and Kauchak, D. (2014). Learning a Lexical Simplifier Using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 458–463.

Kajiwara, T. and Komachi, M. (2018). Complex Word Identification Based on Frequency in a Learner Corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 195–199.

Katsuta, A. and Yamamoto, K. (2018). Crowdsourced Corpus of Sentence Simplification with Core Vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 461–466.

Kodaira, T., Kajiwara, T., and Komachi, M. (2016). Controlled and Balanced Dataset for Japanese Lexical Simplification. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 1–7.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237.

Maddela, M. and Xu, W. (2018). A Word-Complexity Lexicon and A Neural Readability Ranking Model for Lexical Simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3749–3760.

Maekawa, K., Yamazaki, M., Maruyama, T., Yamaguchi, M., Ogura, H., Kashino, W., Ogiso, T., Koiso, H., and Den, Y. (2010). Design, Compilation, and Preliminary Analyses of Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 1483–1486.

Maruyama, T. and Yamamoto, K. (2018). Simplified Corpus with Core Vocabulary. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 1153–1160.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at the International Conference on Learning Representations*, pages 1–12.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Mizukami, M., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2014). Building a Free, General-Domain Paraphrase Database for Japanese. In *Proceedings of the 17th Oriental COCOSDA Conference*, pages 129–133.

Paetzold, G. H. and Specia, L. (2015). LEXenstein: A Framework for Lexical Simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 85–90.

Paetzold, G. H. and Specia, L. (2016). Unsupervised Lexical Simplification for Non-Native Speakers. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 3761–3767.

Paetzold, G. H. and Specia, L. (2017a). A Survey on Lexical Simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Paetzold, G. H. and Specia, L. (2017b). Lexical Simplification with Neural Ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 34–40.

Pavlick, E. and Callison-Burch, C. (2016). Simple PPDB: A Paraphrase Database for Simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 143–148.

Pavlick, E. and Nenkova, A. (2015). Inducing Lexical Style Properties for Paraphrase and Genre Differentiation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 218–224.

Pavlick, E., Rastogi, P., Ganitkevitch, J., Durme, B. V., and Callison-Burch, C. (2015). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 425–430.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Petersen, S. E. and Ostendorf, M. (2007). Text Simplification for Language Learners: A Corpus Analysis. In *Proceedings of Workshop on Speech and Language Technology in Education*, pages 69–72.

Qiang, J., Li, Y., Zhu, Y., Yuan, Y., and Wu, X. (2019). A Simple BERT-Based Approach for Lexical Simplification. *arXiv:1907.06226*.

Shardlow, M. (2014). A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing*, pages 58–70.

Specia, L., Jauhar, S. K., and Mihalcea, R. (2012). SemEval-2012 Task 1: English Lexical Simplification. In *Proceedings of the *SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pages 347–355.

Štajner, S. and Popović, M. (2016). Can Text Simplification Help Machine Translation? *Baltic Journal of Modern Computing*, 4(2):230–242.