Personalizing Lexical Simplification

John Lee, Chak Yan Yeung
Department of Linguistics and Translation
City University of Hong Kong
jsylee@cityu.edu.hk, chak.yeung@my.cityu.edu.hk

Abstract

Given an input text from the user, a lexical simplification (LS) system makes the text easier to understand by substituting difficult words with simpler words. The best substitution may vary from one user to another, given individual differences in vocabulary proficiency level. Most current systems, however, do not consider these variations, and are instead trained to find one optimal substitution or list of substitutions for all users. This paper measures the benefits of using complex word identification (CWI) models to personalize an LS system. Experimental results show that even a simple CWI model, based on graded vocabulary lists, can help reduce the number of unnecessary simplifications and complex words in the output for learners of English at different proficiency levels.

1 Introduction

Lexical simplification (LS) is the task of replacing difficult words with simple words in a text, while preserving its meaning and grammaticality. It aims to produce output text that is easier to understand for readers with special needs, such as language learners, children (Kajiwara et al., 2013), and those with language disabilities (Devlin and Tait, 1998; Carroll et al., 1999). Table 1 shows an example input sentence to an LS system, and the ranked list of possible substitutions for the target word, i.e., the word that should be simplified. Most LS systems first perform complex word identification (CWI) to detect target words (i.e., "avoid" in this case), and then find appropriate substitutions for them (i.e., "prevent", "stop", etc., in order of preference).

CWI is thus an important first step in the LS pipeline. On the one hand, an overly conservative CWI model would fail to detect many complex words, leaving them unsimplified and limiting the utility of the LS system. On the other hand, an overly aggressive CWI model would be prone to misidentify simple words as complex, leading to unnecessary simplifications and increasing the risk of substitution errors. In an error analysis on LS systems, CWI-related error categories turned out to be among the most frequent (Shardlow, 2014). CWI has been receiving increasing attention in recent years, including a recent SemEval shared task (Paetzold and Specia, 2016b). Since the test set was annotated by a single learner, however, CWI performance on language learners at different levels of vocabulary proficiency continues to be under-explored.

Indeed, most LS evaluations assume one best substitution or one fixed ranked list of substitutions (cf. Table 1), and do not take into account variations in vocabulary knowledge among users. This "one-size-fits-all" approach is suboptimal since word complexity is in the eye of the beholder: a word that is complex for a low-proficiency user may be perfectly familiar to a high-proficiency user, or even to a low-proficiency user whose native language has a cognate word. As a case in point, consider the dataset in the SemEval 2016 CWI shared task. The Krippendorff's Alpha agreement was 0.244 among the 20 annotators. Further, suppose one builds an oracle CWI system on the test set in the shared task, and apply it on the Japanese learners of English in the dataset constructed by Ehara et al. (2010). For the

Proceedings of the 27th International Conference on Computational Linguistics, pages 224–232
Santa Fe, New Mexico, USA, August 20-26, 2018.

Input sentence	Gold ranked list of substitutions
Typically, a fast shutter speed will require a larger aperture to ensure sufficient light exposure, and a slow shutter speed will require a smaller aperture to *avoid* excessive exposure.	1. prevent 2. stop 3. {dodge, miss, evade, escape} 4. {elude, limit, avert, bypass, deter}

Table 1: An input sentence to a lexical simplification system, and the gold ranked list of substitutions for the target word, "avoid". This example is taken from the BenchLS dataset (Paetzold and Specia, 2016a).

least proficient learner in this dataset, the oracle would fail to identify 35.30% of the complex words; for the most proficient learner, it would cause false alarm for 93.67% of the non-complex words.

To address "the expected heterogeneity among non-native speakers with different language backgrounds and proficiency levels" (Paetzold and Specia, 2016b), this paper argues for the use of personalized CWI models to improve LS performance. We present the first quantitative evaluation of personalized LS on learners at varying levels of English proficiency. Further, we demonstrate that even a simple CWI model, based on graded vocabulary lists, can reduce the number of unnecessary simplifications and complex words in the output text.

The rest of the paper is organized as follows. The next section summarizes previous LS research, focusing on CWI and Substitution Ranking, where we will attempt personalization. Section 3 gives details on our data. Section 4 describes our approach and baselines. Section 5 defines the evaluation metrics. Section 6 presents experimental results and discusses the extent to which LS systems can benefit from personalized CWI. Finally, Section 7 concludes.

2 Previous work

Most lexical simplification (LS) systems adopt a pipeline architecture (Shardlow, 2014; Paetzold and Specia, 2016b). The pipeline typically begins with **Complex Word Identification** (CWI) to find target words to be simplified. A **Substitution Generation** component then generates candidate replacements for these complex words. These substitutions can be learned, for example, from standard Wikipedia and Simple Wikipedia (Horn et al., 2014), or with word embedding models (Glavaš and Štajner, 2015; Paetzold and Specia, 2016c). The **Substitution Selection** step then discards candidates that may distort the meaning of the text or affect its grammaticality, and retains those that best fits the context. Lastly, **Substitution Ranking** determines the best output by ranking the remaining candidates by simplicity.

LS research has mostly adopted the user-independent approach. We now review previous work in two components of the pipeline to which we will attempt to add personalization: CWI (Section 2.1) and Substitution Ranking (Section 2.2).

2.1 Complex word identification

The complex word identification (CWI) task classifies words in a text as either "complex" or "non-complex". Complex words are those that are difficult for a non-native speaker to understand; non-complex words are those that are not (Paetzold and Specia, 2016b; Yimam et al., 2017). In the 2016 SemEval CWI shared task, the best team, which combined various lexicon-based, threshold-based and machine learning voter sub-systems, achieved a precision of 0.147 and recall of 0.769 (Paetzold and Specia, 2016b). Overall, word frequencies were found to give the most reliable prediction for word complexity. The shared task was not designed to test performance on users at different proficiency levels, since the test set was annotated by a single learner.

To date, most CWI research has taken the user-independent approach, with only a few published studies on personalized CWI. Zeng et al. (2005) showed that demographic features can help improve CWI performance for individual users in the medical domain. Laufer and Nation (1999) proposed the "word sampling" method. Using a ten-level proficiency scale, with 1000 words at each level, this method samples a fixed number of words from the learner as the training set, and then labels unseen words based on their proximity to these words. Ehara et al. (2012; 2014) described a two-step algorithm, mainly using

word frequency statistics as features. In the first step, all words are organized as a multiple complete graph. The k most informative nodes, or words, are selected by a graph-based active learning approach. The learner then rates his/her knowledge of these k words on a five-point scale (see Section 3.1). Lee and Yeung (2018) followed the same procedure to create a training set for Chinese CWI. In the second step, a personal CWI classifier is trained for each learner. Using a 50-word training set, Ehara et al. (2014) achieve 76.44% accuracy in English CWI with Local and Global Consistency, a label propagation algorithm. Lee and Yeung (2018) reported 78.0% accuracy for Chinese CWI with an SVM classifier.

An alternative approach is to build only a fixed number of CWI models. After soliciting annotation of vocabulary knowledge on a small number of sample words from the user, the system predicts the most suitable model. These models can correspond to graded vocabulary lists, such as the New General Service Lists (http://www.newgeneralservicelist.org), or they may potentially be trained on graded text corpora, such as the Newsela corpus (Xu et al., 2015). This approach thus offers more coarse-grained personalization, akin to graded readers, and not every user necessarily fits neatly into one of the predetermined levels.

2.2 Substitution ranking

Given a set of candidates from the Substitution Selection step, the Substitution Ranking step chooses the simplest candidate. Most current approaches impose the same notion of simplicity on all users. Recent systems have applied machine learning approaches, such as the SVM (Horn et al., 2014) and neural models (Paetzold and Specia, 2017), on a range of features including word frequencies in large corpora and human rankings in LS datasets. This step can potentially be enhanced with CWI to filter out candidates that are complex words.

3 Data

In this section, we first describe our dataset of language learners (Section 3.1), and then explain how we used it to create personalized versions of an existing, user-independent dataset of lexical simplification (Section 3.2).

3.1 User dataset

Our user dataset was annotated by 15 learners of English as a foreign language who were native speakers of Japanese (Ehara et al., 2010). Each learner rated their knowledge of 12,000 English words on a five-point scale: (1) Never seen the word before; (2) Probably seen the word before; (3) Absolutely seen the word before but do not know its meaning, or tried to learn the word before but forgot its meaning; (4) Probably know, or able to guess, the words meaning; and (5) Absolutely know the words meaning. Following Ehara et al. (2014), we collapsed these five categories into either "complex" (score 1 through 4) or "non-complex" (score 5). Table 2 shows some example annotations.

These 15 learners covered a wide range of proficiency levels. The least proficient learner rated only 17.97% of the words as "non-complex", while the most proficient one rated 94.26% of the words as "non-complex". To help analyze the effect of personalized LS at different proficiency levels, we define two subsets of learners based on vocabulary proficiency:

- **Low Proficiency.** The four least proficient learners, all of whom knew less than 41% of the words, constitute the "Low" proficiency subset.
- **High Proficiency.** The four most proficient learners, all of whom knew more than 75% of the words, constitute the "High" proficiency subset.

3.2 Personalized lexical simplification dataset

The BenchLS dataset contains 929 instances of target words and their gold simple words, annotated by English speakers from the U.S. (Paetzold and Specia, 2016a). For the 15 users (Section 3.1), we created 15 personalized versions of BenchLS with the following steps:

Word	User A	User B	User C	User D
avert				
avoid				√
bypass	√	√	√	√
deter				√
dodge				√
elude				
escape	√	√		√
evade				√
limit	√	√	√	√
miss	√	√		√
prevent		√		√
stop	√	√	√	√

Table 2: Annotations on 12 example words by four users in the user dataset (Section 3.1). Non-complex words are indicated with a checkmark (√); all other words are complex.

User A	User B	User C	User D
1. stop 2. {miss, escape} 3. {limit, bypass}	1. prevent 2. stop 3. {miss, escape} 4. {limit, bypass}	1. stop 2. {limit, bypass}	*null*

Table 3: Personalized gold ranked list of substitutions for the target word "avoid" in Table 1, based on annotations in the user dataset shown in Table 2.

- When the target word is non-complex for the user, we set the gold answer to *null*. Since the user already understands the word, in the interest of meaning preservation, the system should not simplify it. Consider the target word "avoid" in Table 1. Since it is non-complex for User D (Table 2), the gold answer for this target word should be *null* for User D (Table 3).

- When the target word is complex for the user, the system should attempt simplification on it. We retrieve the gold ranked list of substitutions in BenchLS, and remove all complex words from the list, since they would not be helpful for the user. If the list becomes empty, we exclude this instance from our evaluation. Consider again the target word and its gold list of substitutions in Table 1. When editing this list for Users A, B, and C, we keep only those words that are non-complex for them, according to their annotations in Table 2. Notably, the first-ranked substitution is no longer "prevent" for Users A and C, since they do not know this word. The resulting personal gold lists are shown in Table 3.

After filtering, our evaluation dataset contained 883 instances.

4 Approach

We propose a lexical simplification (LS) algorithm that aims to turn complex words in a text into non-complex ones for the user, while keeping intact the non-complex words in the text. This algorithm applies a personalized complex word identification (CWI) model in two steps in the LS pipeline:

- **CWI for detection**: Most current approaches deploy a user-independent CWI model as the first step in their pipeline to detect words that should be simplified (Section 2.1). In contrast, we train a personalized CWI model for this purpose, such that the choice of target words can vary from one user to another. For example, given the input sentence in Table 1, the system is expected to simplify "avoid" for Users A, B, and C, but not for User D (Table 3).

Level	# Words	Content of vocabulary List
1	1,000	First 1,000 words in the New General Service List (NGSL)
2	2,000	First 2,000 words in the NGSL
3	2,800	All words in the NGSL
4	6,777	All words in the NGSL, the TOEIC Service List (TSL), the New Academic Word List (NAWL), and the Business Service List (BSL)

Table 4: Vocabulary lists corresponding to the four CWI models used in the Graded Vocabulary List approach (Section 4).

- **CWI for ranking**: The Substitution Ranking step of the pipeline ranks the substitution candidates according to their simplicity (Section 2.2). In addition, we apply the personalized CWI model to filter out candidates that are complex for the user. For example, given the list of candidate substitutions in Table 1, the system is expected to reject the word "prevent" for Users A and C, since they do not know it. If the model predicts all candidates to be complex, it still returns the first-ranked candidate as the suggested substitution.

Our experiments apply various configurations of the following three models[1] to perform CWI for detection and CWI for ranking, respectively:

- **Baseline** (`nil`): When used as CWI for detection, this baseline always predicts a word to be complex regardless of the user, so the system always attempts simplification. When used as CWI for ranking, it always predicts a word to be non-complex, so the system never removes any word from the user-independent list of substitutions.

- **Oracle** (`gold`): The oracle performs perfect CWI on each user, according to his/her annotation in the user dataset (Section 3.1). When the oracle is used as CWI for detection, the system attempts simplification if and only if the word is complex. When it is used as CWI for ranking, the system returns the highest-ranked substitution that is non-complex for the user.

- **Graded Vocabulary List** (`auto`): This model automatically predicts a word as complex or non-complex, based on graded vocabulary lists. As shown in Table 4, we define four vocabulary proficiency levels, based on 6,777 words covered by a number of vocabulary lists. We then construct four CWI models corresponding to these four levels; each model predicts all words in its vocabulary list to be "non-complex", and all other words to be "complex".

  Next, we select n out of the 6,777 words in the dataset as the training set, with the n words divided evenly among the four levels. For each user, based on his/her annotation in the user dataset (Section 3.1), we calculate the precision and recall of each of the four CWI models. We then assign the user to the model with the highest F-score. In our evaluation, we set $n = 40$, meaning that each user would have to annotate 40 words as "complex" or "non-complex" in order to personalize the LS system.[2]

5 Evaluation metrics

We report two metrics used in previous LS research (Horn et al., 2014; Glavaš and Štajner, 2015):

- *Precision* is the ratio of correct simplifications out of all simplifications made by the system.

- *Accuracy* is the ratio of correct simplifications out of all target words that should be simplified, i.e., in our context, out of all complex target words.

[1] We did not evaluate the model proposed by Ehara et al. (2014) since we were not able to get access to its system output.
[2] Users who do not know any of the 40 words are assigned to the level-1 model.

CWI for detection	CWI for ranking	Precision	Accuracy	Readability
nil	nil	21.39%	89.95%	91.47%
		43.14% (low only)	76.31% (low only)	80.02% (low only)
		4.25% (high only)	100% (high only)	99.01% (high only)
auto	nil	31.97%	76.19%	89.40%
nil	auto	23.36%	**94.19%**	**94.57%**
auto	auto	**34.81%**	80.36%	91.67%
gold	nil	89.95%	89.95%	95.55%
nil	gold	26.31%	100%	100%
gold	gold	100%	100%	100%

Table 5: Performance on the personalized LS dataset (Section 3.2), based on gold substitution lists in BenchLS (Paetzold and Specia, 2016b) and on three methods for CWI for detection and CWI for ranking: nil predicts all target words to be complex, and all candidate substitutions to be non-complex; gold returns the annotation in the user dataset (Section 3.1); auto is the automatic CWI model based on graded vocabulary lists. Low and high refer to proficiency level (Section 3.1).

Correctness of a simplification is based on the personalized LS dataset (Section 3.2) rather than the BenchLS dataset (Paetzold and Specia, 2016a). For instance, in the sentence in Table 1, it is correct to substitute "avoid" with "prevent" for User B, but incorrect to do so for User A and C. Further, it is deemed incorrect to make any substitution for User D (Table 3).

Note that precision penalizes simplifications of non-complex words, even if the substitution is also a non-complex word in the gold list in BenchLS. For some users, this penalty may be reasonable since few substitutions fully preserve the meaning and intent of the original text. For others, unnecessary simplifications may be perfectly acceptable, given the overriding goal of minimizing the number of unknown words in the output text.

To represent the latter perspective, we also report the *readability* metric, which computes the proportion of words in the output text that can be understood by the user and do not distort the original meaning. More precisely, a word in the output text is *readable* if it satisfies two conditions: (1) it is non-complex for the user; and (2) it is either included in the original gold substitution list in BenchLS, or it is unsimplified. Since this metric does not consider whether the original word is complex or non-complex, it allows unnecessary simplification as long as it is appropriate.

6 Experiments

We conducted two experiments to evaluate the effect of adding personalization to a lexical simplification (LS) system. In both experiments, the baseline is a user-independent ranked list of substitutions. We manipulate the list with various combinations of CWI for detection and/or CWI for ranking (see Section 4), and then measure any gain in LS performance. All results are averaged among the 15 users in the dataset (Section 3.1).

6.1 Experiment 1: Personalization with gold substitutions

To better isolate the performance gain as a result of personalization, the first experiment used the gold substitution lists in BenchLS. This design ensures that the performance gain would not be influenced by the extent and nature of the particular substitution errors made by the LS system chosen as baseline.

Table 5 reports performance on the personalized LS dataset (Section 3.2). Because of the use of gold substitutions in BenchLS, the absolute level of performance is overestimated. We will focus on the *difference* between the baseline and the personalized systems, and will verify if the difference holds in realistic conditions in the second experiment.

Precision. The oracle (detect=gold, rank=gold), by definition, achieved the perfect score in all metrics. In contrast, the user-independent approach (detect=nil, rank=nil), even with perfect substi-

CWI for detection	CWI for ranking	Precision	Accuracy	Readability
nil	nil	8.09% (low only) 14.87% (high only) 1.96%	39.17% (low only) 27.06% (high only) 51.25%	37.94% (low only) 27.53% (high only) 45.94%
auto	nil	12.03%	32.59%	61.27%
nil	auto	12.37%	**50.91%**	53.45%
auto	auto	**18.01%**	42.25%	**69.39%**
gold	nil	39.28%	39.17%	83.38%
nil	gold	14.57%	52.52%	56.01%
gold	gold	58.57%	52.52%	87.44%

Table 6: Performance on the personalized LS dataset (Section 3.2), based on output from a user-independent LS system (Paetzold and Specia, 2017), and on three methods for CWI for detection and CWI for ranking: nil predicts all target words to be complex, and all candidate substitutions to be non-complex; gold returns the annotation in the user dataset (Section 3.1); auto is the automatic CWI model based on graded vocabulary lists. Low and high refer to proficiency level (Section 3.1).

tutions, hit a ceiling at 21.39% precision. One source of error for precision was the simplification of non-complex words in the input text, since the system always attempted simplification. Naturally, this was especially problematic for high-proficiency users, as reflected in the lower precision (4.25%), but had less impact on low-proficiency users (43.14% precision). Personalized CWI reduced these unnecessary simplifications, raising precision to as high as 89.95% with oracle CWI for detection (detect=gold, rank=nil). The automatic CWI approach (detect=auto, rank=nil), based on vocabulary lists, also succeeded in reducing them and attained 31.97% precision, a 10% absolute improvement over the baseline.

Accuracy. Another source of error for the user-independent approach was the fact that some gold substitutions were complex; in other words, while these substitutions were considered simpler than the target words, they were still too difficult for the user. This phenomenon resulted in the 89.95% accuracy rate for the user-independent approach (detect=nil, rank=nil). As expected, the phenomenon was magnified among low-proficiency users, as shown by the lower accuracy (76.31%), but it barely affected the high-proficiency users (100% accuracy). Personalized CWI helped steer the system to choose non-complex words as output. Oracle CWI for ranking (detect=nil, rank=gold), by definition, achieved 100% accuracy. The automatic CWI approach (detect=nil, rank=auto) yielded smaller improvement but, at 94.19% accuracy, still outperformed the baseline by over 4% absolute.

Readability. With respect to the readability measure, which accepts unnecessary simplifications, personalized approaches still produced better output than the user-independent baseline. Among configurations that did not involve gold, the highest readability score (94.57%) was achieved by the system that always attempted simplification but used automatic CWI for ranking (detect=nil, rank=auto); this represented a 3% improvement over the baseline. It also achieved the highest accuracy (94.19%). Using automatic CWI for both detection and ranking (detect=auto, rank=auto) produced the best precision (34.81%), an absolute improvement of over 13% over the baseline. However, its accuracy and readability were suboptimal because, by making fewer simplifications, it left more complex words in the input text unsimplified.

6.2 Experiment 2: Personalization with automatically generated substitutions

Results from the first experiment, which assumed perfect substitutions, showed consistent performance gains as a result of personalization. The second experiment investigated whether these gains would hold under more realistic conditions. Instead of gold substitutions from BenchLS, we used the output of a state-of-the-art, user-independent LS system (Paetzold and Specia, 2017). Table 6 shows the performance on the personalized LS dataset (Section 3.2). In this setting, the oracle (detect=gold, rank=gold)

attained only 58.57% precision and 52.52% accuracy. As will be discussed below, the gap between the baseline and the personalized systems persisted.

Precision. The user-independent approach (detect=`nil`, rank=`nil`) achieved only 8.09% precision. A major source of error for precision, as observed in the first experiment, was the simplification of non-complex words in the input text. Oracle CWI for detection (detect=`gold`, rank=`nil`) raised the precision to 39.28%. Automatic CWI for detection (detect=`auto`, rank=`nil`) yielded 12.03% precision, improving the baseline by almost 4% absolute.

Accuracy. The ability of personalization to reduce the other major source of error — selection of complex words as substitutions — was also observed in this experiment. While the accuracy of the user-independent approach (detect=`nil`, rank=`nil`) was only 39.17%, automatic CWI for ranking (detect=`nil`, rank=`auto`) improved it by over 11% to reach 50.91%. This level of performance was very close to the upper bound of 52.52% accuracy suggested by oracle CWI for ranking (detect=`nil`, rank=`gold`).

Readability. In terms of readability, after excluding configurations that involve `gold`, the best readability score (69.39%) was achieved by using automatic CWI for both detection and ranking (detect=`auto`, rank=`auto`). This represented an absolute improvement of over 31% in comparison to the user-independent baseline. Unlike in the first experiment, the system's conservativeness in making simplifications worked in its favor because of the presence of substitution errors. This configuration also yielded the highest precision (18.01%), outperforming the baseline by almost 10%. However, the highest accuracy (50.91%), similar to the first experiment, was obtained by using automatic CWI for ranking only (detect=`nil`, rank=`auto`).

7 Conclusion

Most current approaches to lexical simplification (LS) are user-independent. This paper proposed the use of personalized models of complex word identification (CWI) to tailor LS systems to the vocabulary proficiency of the user. We presented the first study on the effect of personalized CWI in two steps of the LS pipeline: to detect which words require simplification, and to reject substitution candidates that are still too difficult for the user. We measured both the upper bounds of performance gains with an oracle CWI model, as well as the actual gains of a simple automatic CWI model that required only a 40-word training set per user, based on graded vocabulary lists.

Experimental results with oracle CWI demonstrated much room for improving LS systems through personalization. Further, systems that used the automatic CWI model consistently outperformed the user-independent baseline, by reducing both the number of unnecessary simplifications and the number of complex words in the output. The performance gains persisted regardless of whether the substitutions were gold or automatically generated, yielding improvement in precision and accuracy ranging from 4% to 13%. As CWI research produces higher-performing models, future LS systems can expect to derive even greater benefits from personalization.

Acknowledgements

This work was supported by the Innovation and Technology Fund (Ref: ITS/132/15) of the Innovation and Technology Commission, the Government of the Hong Kong Special Administrative Region; and by CityU Internal Funds for ITF Projects (no. 9678104). We thank Dr. Lis Pereira for assisting with the experiments.

References

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying Text for Language-Impaired Readers. In *Proc. 9th EACL.*

Siobhan Devlin and John Tait. 1998. The Use of a Psycholinguistic Database in the Simplification of Text for Aphasic Readers. *Linguistic Databases*, pages 161–173.

Yo Ehara, Nobuyuki Shimizu, Takashi Ninomiya, and Hiroshi Nakagawa. 2010. Personalized Reading Support for Second-Language Web Documents by Collective Intelligence. In *Proc. 15th International Conference on Intelligent User Interfaces*, pages 51–60.

Yo Ehara, Issei Sato, Hidekazu Oiwa, and Hiroshi Nakagawa. 2012. Mining words in the minds of second language learners: learner-specific word difficulty. In *Proc. COLING*.

Yo Ehara, Yusuke Miyao, Hidekazu Oiwa, Issei Sato, and Hiroshi Nakagawa. 2014. Formalizing Word Sampling for Vocabulary Prediction as Graph-based Active Learning. In *Proc. EMNLP*.

Goran Glavaš and Sanja Štajner. 2015. Simplifying Lexical Simplification: Do We Need Simplified Corpora? In *Proc. ACL*.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a Lexical Simplifier Using Wikipedia. In *Proc. ACL*.

Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting Proper Lexical Paraphrase for Children. In *Proc. 25th Conference on Computational Linguistics and Speech Processing (ROCLING)*, pages 59–73.

Batia Laufer and Paul Nation. 1999. A Vocabulary-size Test of Controlled Productive Ability. *Language Testing*, 16(1):33–51.

John Lee and Chak Yan Yeung. 2018. Automatic Prediction of Vocabulary Knowledge for Learners of Chinese as a Foreign Language. In *Proc. International Conference on Natural Language and Speech Processing (ICNLSP)*.

Gustavo H. Paetzold and Lucia Specia. 2016a. Benchmarking Lexical Simplification Systems. In *Proc. LREC*.

Gustavo H. Paetzold and Lucia Specia. 2016b. SemEval 2016 Task 11: Complex Word Identification. In *Proc. 10th International Workshop on Semantic Evaluation (SemEval-2016)*.

Gustavo H. Paetzold and Lucia Specia. 2016c. Unsupervised Lexical Simplification for Non-native Speakers. In *Proc. AAAI*.

Gustavo H. Paetzold and Lucia Specia. 2017. Lexical Simplification with Neural Ranking. In *Proc. EACL*.

Matthew Shardlow. 2014. Out in the Open: Finding and Categorising Errors in the Lexical Simplification Pipeline. In *Proc. LREC*.

Wei Xu, Chris Callison-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.

Seid Muhie Yimam, Sanja Stajner, Martin Riedl, and Chris Biemann. 2017. Multilingual and Cross-Lingual Complex Word Identification. In *Proc. Recent Advances in Natural Language Processing (RANLP)*.

Qing Zeng, Eunjung Kim, Jon Crowell, and Tony Tse. 2005. A Text Corpora-based Estimation of the Familiarity of Health Terminology. In J.L. Oliveira et al., editor, *ISBMDA 2005, LNBI 3745*, pages 184–192.