Personalized Substitution Ranking for 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

A lexical simplification (LS) system substitutes difficult words in a text with simpler ones to make it easier for the user to understand. In the typical LS pipeline, the Substitution Ranking step determines the best substitution out of a set of candidates. Most current systems do not consider the user's vocabulary proficiency, and always aim for the simplest candidate. This approach may overlook less-simple candidates that the user can understand, and that are semantically closer to the original word. We propose a personalized approach for Substitution Ranking to identify the candidate that is the closest synonym and is non-complex for the user. In experiments on learners of English at different proficiency levels, we show that this approach enhances the semantic faithfulness of the output, at the cost of a relatively small increase in the number of complex words.

1 Introduction

A lexical simplification (LS) system aims to make a text easier to understand for users such as language learners (Petersen and Ostendorf, 2007), children (Belder and Moens, 2010; Kajiwara et al., 2013), those with language disabilities (Devlin and Tait, 1998; Carroll et al., 1999; Rello et al., 2013), as well as those without the background needed for understanding the text in a specialized domain (Zeng et al., 2005; Elhadad, 2006). Given an input text, the system substitutes difficult words with simpler words or phrases, in such a way as to satisfy two requirements:

Semantic faithfulness The output text should preserve the meaning of the input as much as possible;

Word complexity The output text should minimize the number of complex words, i.e., words that the user cannot understand.

These two goals are often in conflict. To reduce word complexity, the system should make substitutions with the simplest words possible. However, when limited to simple words, it is more challenging to find a close synonym for the original word. In general, the larger the vocabulary pool, the better the simplified text can semantically approximate the original.

The trade-off between these two goals is best resolved with respect to the user's vocabulary knowledge. For example, to simplify the word "obnoxious", the word "offensive" would be the best substitution *if the user knows it*; otherwise, a looser synonym but simpler word such as "bad" should be preferred. Human editors follow this principle when composing graded versions of a text: they stick to simple substitutions ("bad") in basic versions, but allow more difficult words ("offensive") in more advanced versions, if these words better reflect the original meaning.

Current LS approaches do not mimic this strategy. The typical system offers the same substitutions, regardless of individual users' language proficiency, because it is trained on simple-complex text pairs that do not specify the target reader. With the notable exception of the Newsela corpus (Xu et al., 2015), annotators for LS corpora are not typically given precise guidelines on the vocabulary proficiency of the intended user, making it difficult to make optimal or consistent trade-off between semantic faithfulness and word complexity.

A possible solution is to use Newsela-style corpora to train systems to automatically generate multiple simplified versions, optimized for readers at different proficiency levels. The sheer amount of annotation required, however, would limit the granularity of the proficiency levels. This paper instead explores an alternative solution that leverages two existing subfields of computational

Proceedings of The 12th International Conference on Natural Language Generation, pages 258–267,
Tokyo, Japan, 28 Oct - 1 Nov, 2019. ©2019 Association for Computational Linguistics

Word	User A	User B	User C
liberal	×	×	√
open	√	√	√
progressive	√	×	√
relaxed	√	√	√

Table 1: Example CWI model predictions on three users' vocabulary knowledge. Complex words are marked with a cross (×), and non-complex ones with a tick (√).

linguistics. We propose a novel, two-step Substitution Ranking algorithm for the LS pipeline, by combining lexical substitution (McCarthy and Navigli, 2009)[1] and personalized complex word identification (CWI) (Ehara et al., 2014). This algorithm first ranks the substitution candidates according to semantic proximity to the target word; it then selects as output the highest-ranked candidate that is non-complex for the user.

We make two main contributions. First, in terms of evaluation methodology, we present the first LS evaluation that explicitly takes into account the trade-off between semantic faithfulness and word complexity for users at different proficiency levels. Second, we show that our proposed Substitution Ranking algorithm leads to substitutions that are more meaning-preserving, with a relatively small increase in word complexity.

The goal of this paper is not to advance the state-of-the-art for lexical substitution or for CWI. Instead, we aim to show that, by combining existing methods in these two tasks, an LS system achieves superior trade-off between semantic faithfulness and word complexity, compared to the conventional approach of optimizing on word simplicity alone.

The rest of the paper is organized as follows. After a presentation of the general LS system architecture in the next section, Section 3 summarizes previous research in Substitution Ranking. Section 4 describes our proposed approach. Section 5 gives details on our datasets. Section 6 defines the evaluation metrics. Section 7 discusses the experimental set-up, and Section 8 presents experimental results. Finally, Section 9 concludes.

2 Background: LS pipeline

The most common LS architecture is a pipeline architecture with three steps: Complex Word Identification, Substitution Generation, and Substitution Ranking (Shardlow, 2014; Paetzold and Specia, 2016c).

2.1 Complex Word Identification (CWI)

CWI classifies each word in an input text as either "complex" (i.e., difficult to understand) or "non-complex" (i.e., not difficult to understand) (Paetzold and Specia, 2016c; Yimam et al., 2017). Complex words become *target words* for substitution in the rest of the pipeline.

Most CWI approaches adopt a generic definition of "complexity", without catering to variation in proficiency level among users. Some studies have begun to build personalized CWI models to predict whether a word can or cannot be understood by an individual user (Ehara et al., 2014; Lee and Yeung, 2018). For example, the top row in Table 1 shows the predictions for the word "liberal" for three users. Given these predictions, the system would attempt to simplify the word "liberal" in the sentence in Table 2 for Users A and B, but would not do so for User C.

2.2 Substitution Generation

For each target word, the Substitution Generation step identifies possible substitution candidates, without assessing the simplicity of the candidates.[2] In our running example in Table 2, the system generates three possible synonyms — "open", "progressive", and "relaxed" — as substitution candidates for "liberal".

2.3 Substitution Ranking

As the final step, Substitution Ranking chooses the best simplification among the generated candidates. The next section review previous research on this step, as well as previous work that we use as the basis for personalizing it.

3 Previous work

Substitution Ranking is "the task of ranking a set of selected substitutions for a target complex word with respect to their simplicity." (Paetzold and Specia, 2015). As this definition suggests, most

[1] In this paper, the abbreviation LS always refers to "Lexical Simplification", and not to "Lexical Substitution."

[2] Some systems take a separate Substitution Selection step to discard candidates that distort the meaning of the text or affect its grammaticality, and retain those that fit the context.

<table>
<tr><td colspan="5"><b>Input sentence</b>: She is a product of a tremendously unorthodox family with <i>liberal</i> views toward sex, marriage, religion and child-rearing.</td></tr>
<tr><th>Substitution Ranking algorithm</th><th>User</th><th colspan="3">Ranked substitution candidates for <i>liberal</i></th></tr>
<tr><td>(a) Simplicity ranking</td><td>n/a</td><td>1. open</td><td>2. relaxed</td><td>3. progressive</td></tr>
<tr><td>(b) Similarity ranking</td><td>n/a</td><td>1. progressive</td><td>2. relaxed</td><td>3. open</td></tr>
<tr><td>(c) Personalized ranking</td><td>User A</td><td>1. progressive</td><td>2. relaxed</td><td>3. open</td></tr>
<tr><td></td><td>User B</td><td>~~1. progressive~~</td><td>1. relaxed</td><td>2. open</td></tr>
</table>

Table 2: Gold ranking of substitution candidates for the target word "liberal" by optimizing on (a) lexical simplicity; (b) semantic similarity to the target word; (c) both user vocabulary knowledge (Table 1) *and* semantic similarity to the target word.

LS systems use simplicity, or word complexity, as the basis for ranking (Section 3.1). In contrast, we will propose an approach that takes into account both word complexity and semantic faithfulness, by building on previous research on ranking by semantic similarity (Section 3.2) and personalized CWI (Section 3.3).

3.1 Simplicity ranking

Various statistical models have been trained to rank substitution candidates by "simplicity", using a wide range of features including the number of syllables, word frequencies, n-gram language model scores, word embeddings, as well as relative frequencies in standard Wikipedia and Simple Wikipedia (Carroll et al., 1999; Ligozat et al., 2012; Horn et al., 2014; Glavaš and Štajner, 2015; Pavlick and Callison-Burch, 2016). Among the three candidates in our running example, "open" would be ranked first, and thus chosen to substitute for "liberal" (Table 2a).

Researchers have recognized that the "one-size-fits-all" approach does not adequately cater to users at different vocabulary proficiency levels, since they do not share the same notion of "simplicity". Some have begun to explore adaptation of the ranking through user feedback (Paetzold and Specia, 2016a; Yimam and Biemann, 2018). For example, the Lexi system asks the user whether the original word or a candidate substitution makes the sentence easier to understand, and then uses pairwise online logistic regression to adapt its simplicity ranking (Bingel et al., 2018).

3.2 Similarity ranking

Another possible criterion for ranking substitution candidates is their semantic similarity, or proximity, to the target word. On this criterion, "progressive" would be ranked first among the three candidates in our running example (Table 2b).

Similarity ranking has been intensively studied for the task of lexical substitution (McCarthy and Navigli, 2009). To find the most appropriate paraphrases, PPDB uses a scoring model that estimates semantic distance between two words with WordNet, and also considers lexical overlap, distributional similarity, as well as cosine similarity of word embeddings (Pavlick et al., 2015). The SALSA system performs similarity ranking through latent semantic analysis, explicit semantic analysis and n-gram scores (Sinha and Mihalcea, 2014). It has been deployed in a reading assistance tool that displays synonyms for difficult words (Azab et al., 2015), but it does not attempt LS.

3.3 Personalized CWI

The first studies on personalized CWI were reported by Ehara et al. (2012; 2014). They proposed a graph-based active learning method, trained on word frequencies from various corpora, to select the most informative words for user annotation. A label propagation algorithm achieved an accuracy of 76.4% for English CWI on a 50-word training set (Ehara et al., 2014).

Lee and Yeung (2018) applied personalized CWI on LS. Their LS algorithm assumes a personal CWI model for each user. There were four possible CWI models, each corresponding to a vocabulary list. Each user is given the model that optimizes the F-measure on his or her small CWI training set. During ranking of the substitution candidates, candidates that are deemed complex for the user are rejected. Experimental results showed that this approach reduced both unnecessary simplification and the word complexity of the output text. Although their approach is similar to our CWI filtering step (Section 4), they did not incorporate similarity ranking or evaluate its effect on semantic faithfulness in the LS output.

4 Proposed Approach

In the Substitution Ranking step, the optimal candidate is often not the simplest one. A more pertinent criterion is whether the candidate is non-complex, i.e., whether it can be understood by the user. This is because a non-complex candidate that is more similar in meaning to the target word should be preferred, even if it is less simple. Our proposed algorithm selects the candidate that is semantically closest to the target word (to maximize semantic faithfulness), with the constraint that the candidate be non-complex for the user (to minimize word complexity). More concretely, it consists of two steps:

Similarity ranking Rank the substitution candidates according to semantic proximity, as a lexical substitution system would do.

CWI filtering Personalize the ranking by removing candidates that the user cannot understand, according to the prediction of the personalized CWI model.

Note that while CWI filtering employs the same CWI model as the one used in the first step in the LS pipeline (Section 2.1), it predicts user knowledge on the substitution candidates, rather than on the target word.

In our running example, for the word “liberal”, the similarity ranking step would yield the ranked list “1. progressive; 2. relaxed; 3. open” (Table 2b), since “progressive” is semantically closest to the target word “liberal”. The CWI filtering step then produces a personalized ranking for each user (Table 2c). Based on the personalized CWI model (Table 1), the ranked list for User A remains unchanged, since all three candidates are known to User A. The word “progressive” is removed from the ranked list for User B, however, since User B does not understand it. The second-best synonym, “relaxed”, is instead offered to User B.

5 Data

Standard LS datasets, such as BenchLS (Paetzold and Specia, 2016b) and the Newsela corpus (Xu et al., 2015), are not suitable for evaluating our proposed ranking approach (Section 4) since they do not offer human judgment on semantic similarity between target words and their substitutions. We instead took an existing paraphrase dataset as our starting point (Section 5.1), and then exploited a language learner dataset (Section 5.2) to construct personalized substitution rankings (Section 5.3).

5.1 Similarity rankings

During the development of PPDB 2.0 (Pavlick et al., 2015), human judgment on similarity was collected for 40,410 phrase pairs. Five human raters assigned a similarity score on a 5-point scale to each pair. For each target word, we computed the similarity ranking of its candidates according to their average score. While other lexical substitution corpora such as the 2007 English Lexical Substitution shared task (McCarthy and Navigli, 2009) and CoInCo (Kremer et al., 2014) could also serve as evaluation data, we chose PPDB because it offers a much larger number of substitution candidates per word, and thus facilitates a clear comparison between the simplicity-based approach and ours. We will refer to this dataset as “PPDB Set”.

5.2 Personalization dataset

We used a personalized CWI dataset annotated by 15 learners of English as a foreign language (Ehara et al., 2010). Each learner rated their knowledge of 12,000 English words on a five-point scale. Following Ehara et al. (2014), we collapsed these five categories into either “complex” (score 1 through 4) or “non-complex” (score 5). For analysis purposes, we define the seven more advanced students as “high-proficiency”, and the remaining students as “low-proficiency”.

A disadvantage of this dataset is that it does not situate the target words in sentences. As a result, it is not possible to evaluate an in-context CWI model in this study. We decided to use this dataset because of its scale and wide representation: no other existing personalized CWI dataset approaches its size, with over 180K annotations by language learners spanning a wide range of proficiency levels.

5.3 Personalized rankings

We personalized the substitution candidate rankings in the PPDB Set (Section 5.1) for each of the 15 users in the personalized CWI dataset (Section 5.2), as follows:

- We excluded target words that are not present in the CWI dataset;
- We excluded target words that are annotated as non-complex for the user, since simplification is not needed;
- We removed substitution candidates from the ranked list if they are annotated as complex for the user, since they are not acceptable LS output for that user. For example, since User B does not understand the word "progressive" (Table 1), it is removed from his list and his optimal substitution becomes "relaxed".

Hence, each target word has 15 different personalized rankings of substitution candidates, derived from the original similarity ranking in the PPDB Set. These personalized rankings serves as the gold answer (Table 2c).

After these processing steps, our final dataset contains an average of 64.5 instances of word simplification for each of the 15 users; each target word has an average of 22.1 substitution candidates.

6 Evaluation metrics

Since the proposed algorithm aims to optimize the trade-off between word complexity and semantic faithfulness, we need to define metrics for both of these qualities.

6.1 Word complexity

We use the standard metric, Precision, to measure the simplicity of the output text.[3] Precision is defined as the ratio of correct simplifications out of all simplifications made by an LS system (Horn et al., 2014; Glavaš and Štajner, 2015). All candidates in the personalized ranking, without regard to its semantic faithfulness, are considered "correct". Thus, the higher the precision, the lower the word complexity of the output, i.e., fewer words are complex for the user.

We will rename Precision as **Precision**$_{all}$ for reasons to be described in the next subsection.

6.2 Semantic faithfulness

It is not straightforward to draw a line on what suffices as a semantically "similar" substitution, or to specify the minimum level of semantic faithfulness. Extending the definition of Precision, we introduce the metric **Precision**$_r$ to express the degree of semantic similarity between the target word and its substitution, where the parameter r specifies the maximum position in the personalized ranking for the candidate to be considered "correct". More specifically, Precision$_r$ is the percentage of substitutions made by the system that are ranked r or above. The larger the value of r, the less strict the metric is on semantic faithfulness.

As an illustration, consider User A in Table 2c. The **Precision**$_1$ metric (i.e., $r = 1$) counts as correct only the first-ranked substitution in the gold ranking, i.e., "progressive". In contrast, the **Precision**$_2$ metric (i.e., $r = 2$) counts both "progressive" and "relaxed" as correct.

In this context, then, the standard definition of Precision (Section 6.1) can be viewed as Precision$_r$ where r is allowed to be any value. For clarity, we will refer to Precision as **Precision**$_{all}$ in the rest of this paper.

7 Experimental set-up

This section describes the experimental design to evaluate the two-step Substitution Ranking approach proposed in Section 4.

7.1 Ranking methods

In the first step, our proposed approach performs similarity ranking on the candidates:

Similarity Ranking (Automatic) We ranked the substitution candidates according to path similarity between the candidates and the target word using WordNet 3.0. The path similarity score denotes how similar two words are based on the shortest path that connects them. WordNet synsets have been found to be more coarse-grained, but consistent with lexical substitute sets (Kremer et al., 2014).

Similarity Ranking (Gold) This is derived from the gold similarity rankings in the PPDB Set (Section 5.1). This ceiling establishes the maximum performance of the proposed approach.

[3] Another standard metric, accuracy, is defined as the ratio of correct simplifications out of all target words that should be simplified (Horn et al., 2014; Glavaš and Štajner, 2015) Since we are concerned only with Substitution Ranking, the system attempts to simplify all complex words. Therefore, accuracy is always the same as precision in our experiments.

Our baseline ranks candidates with respect to simplicity, which is the prevalent approach in current LS systems:

Simplicity Ranking (Gold) We could not use the human judgement on simplicity collected during the construction of Simple PPDB (Pavlick and Callison-Burch, 2016), since it was performed on a set of word pairs that do not overlap with the gold similarity rankings in the PPDB Set. Instead, we derived the gold simplicity ranking from the personalization dataset (Section 5.2) by averaging the 15 subjects' annotations on each word.

Simplicity Ranking (Automatic) We ranked the substitution candidates according to their gold scores in the probabilistic track of the 2018 CWI shared task (Yimam and Biemann, 2018). The score is defined as the proportion of annotators who marked the word as "complex". The use of CWI gold scores, rather than an actual system output, ensured a strong baseline for comparison against our proposed approach.

7.2 CWI filtering methods

In the second step, the proposed approach uses a personalized CWI model to remove complex words from the ranked list. If all candidates are predicted to be complex, the first-ranked candidate is proposed as the substitution.

Gold Personalized CWI (Gold PCWI) This ceiling establishes the maximum performance of the proposed approach, by looking up the actual CWI annotation for each user (Section 5.2). In other words, the system can perfectly predict whether a particular user knows a particular word.

Automatic Personalized CWI (Auto PCWI) This model automatically predicts a word as complex or non-complex, based on graded vocabulary lists. We initially created four word lists with the 6,777 words covered by the New General Service List (NGSL).[4] Further, we split them into 222 groups of 25 words each, by sorting the words within each list according to frequency in the Google Web Trillion Word Corpus (Brants and Franz, 2006). Next, we constructed 222 CWI models corresponding to these groups: each model predicts the words in its group and the preceding ones to be "non-complex", and all other words to be "complex".

We then randomly selected 50 words as the training set.[5] For each user, we computed the precision and recall of each of the 222 CWI models on this training set, and then selected the model with the highest F-score.

Our baselines for CWI filtering are as follows[6]:

No CWI This baseline predicts all words to be non-complex; in other words, the system always returns the top-ranked candidate as output.

Generic CWI This baseline optimizes its prediction on the learner dataset (Section 5.2); in other words, it predicts a word to be complex if and only if a majority of the 15 users annotated it to be so. This baseline establishes the maximum performance of the generic, or user-independent CWI approach.

8 Experiments

We performed two experiments on the Substitution Ranking approach proposed in Section 4.

8.1 Results with gold similarity ranking

In this first experiment, the system used the gold similarity ranking and performed automatic CWI filtering only. This set-up is designed to provide a controlled contrast between similarity ranking and simplicity ranking by excluding noise from automatic semantic analysis. In the remainder of this subsection, please refer to the results in Table 3.

Word complexity. We first consider the Precision$_{all}$ results. When assuming perfect prediction of users' vocabulary knowledge (Filtering="Gold PCWI"), all simplifications were by definition non-complex, resulting in 100% precision for both ranking methods.

With automatic prediction of the users' vocabulary knowledge (Filtering="Auto PCWI"), the use of similarity ranking led to more complex words

[4] The composition of these four lists followed the specifications in Lee and Yeung (2018).

[5] In actual deployment, this set-up means that each user would be required to annotate 50 words.

[6] We did not evaluate the model proposed by Ehara et al. (2014) since we were not able to get access to its system output.

Metric	**Precision$_1$**		**Precision$_2$**		**Precision$_{all}$**	
Ranking → ↓ Filtering	Gold Simplicity (Baseline)	Gold Similarity (Proposed)	Gold Simplicity (Baseline)	Gold Similarity (Proposed)	Gold Simplicity (Baseline)	Gold Similarity (Proposed)
No CWI	17.74%	**30.08%*** (+12.34%)	37.53%	**72.44%*** (+34.91%)	**92.95%***	72.44% (-20.51%)
Generic CWI	27.61%	**81.94%*** (+54.33%)	40.22%	**85.81%*** (45.59%)	**99.41%***	87.00% (-12.41%)
Gold PCWI	28.13%	**100%*** (+71.87%)	40.74%	**100%*** (+59.26%)	100%	100% (same)
Auto PCWI	26.48%	**77.79%*** (+51.31%)	39.43%	**86.38%*** (+46.95%)	**99.00%***	90.70% (-8.30%)
- *low only*	28.12%	**74.97%*** (+46.85%)	40.26%	**84.43%*** (+44.17%)	**98.43%***	86.83% (-11.60%)
- *high only*	24.62%	**81.00%*** (+56.38%)	38.47%	**88.62%*** (+50.15%)	**99.65%***	95.13% (-4.52%)

Table 3: LS performance using gold ranking and various CWI filtering methods on the dataset in Section 5.3. An asterisk means the improvement over the other ranking is significant ($p < 0.05$ by McNemar's test).

(-8.30% Precision$_{all}$) than simplicity ranking. The degradation is also observed for "No CWI" and "Generic CWI". This is because similarity ranking prioritizes semantically closer synonyms, which tend to be more difficult and therefore more likely to produce a complex output in the face of CWI error. In contrast, by always aiming for the simplest words, simplicity ranking reduced the chance of producing complex words in the output.

Semantic faithfulness. We next consider results with the Precision$_r$ metric. With perfect prediction of users' vocabulary knowledge (Filtering="Gold PCWI"), gold similarity ranking by definition chose the best non-complex synonym, hence achieving perfect semantic scores. In contrast, gold simplicity ranking rarely picked the best synonym (28.13% at Precision$_1$), and most of the time did not pick either of the top two synonyms (40.74% at Precision$_2$).

With automatic prediction of vocabulary knowledge (Filtering="Auto PCWI"), even gold similarity ranking was liable to selecting complex candidates. Its performance degraded to 77.79% for Precision$_1$, and 86.38% for Precision$_2$. It still outperformed gold simplicity ranking by large margins, with an absolute improvement of +51.31% for Precision$_1$ and +46.95% for Precision$_2$.

Impact of language proficiency. We now examine how language proficiency affects LS performance, using results obtained with automatic prediction of vocabulary knowledge (Filtering="Auto PCWI"). Overall, the similarity ranking method led to a 51.31% gain in semantic faithfulness (in terms of Precision$_1$), at the expense of a 8.3% increase in word complexity (i.e., a 8.3% loss in terms of Precision$_{all}$).

In terms of semantic faithfulness, high-proficiency users (as defined in Section 5.2) experienced a gain of 56.38% in Precision$_1$ (Filtering="Auto PCWI - high only"), compared to a gain of only 46.85% for low-proficiency users (Filtering="Auto PCWI - low only"). This is because a larger vocabulary pool resulted in more instances where a non-complex and better synonym was available. The trend was similar for Precision$_2$, though the absolute gains were smaller for both groups of users.

In terms of word complexity, our proposed method also led to a smaller increase in unknown words for high-proficiency users than low-proficiency users. The former experienced a loss of 4.52% in Precision$_{all}$ (Filtering="Auto PCWI - high only"), while the latter suffered a loss of 11.60% (Filtering="Auto PCWI - low only"). This is because similarity ranking, being more aggressive in utilizing more difficult words, is less likely to produce a complex word for a higher-proficiency user.

Thus, generally speaking, the higher the users' language proficiency, the more they benefited from the proposed method. For more proficient users, the LS output shows increasing gain in semantic faithfulness and diminishing degradation in word complexity.

8.2 Fully automatic ranking

The second experiment adopted the realistic setting, with the system performing automatic processing for both steps in Substitution Ranking. Overall, results followed the same trends as observed in Section 8.1, but with lower performance in all settings. In the remainder of this subsection, please refer to the results in Table 4.

Word complexity. We first consider the results in Precision$_{all}$. Given perfect prediction of users' vocabulary knowledge (Filtering="Gold PCWI"), all simplifications were non-complex, resulting in 100% precision for both ranking methods.

For the same reasons as laid out in the previ-

Metric	**Precision$_1$**		**Precision$_2$**		**Precision$_{all}$**	
Ranking → ↓ Filtering	Automatic Simplicity (Baseline)	Automatic Similarity (Proposed)	Automatic Simplicity (Baseline)	Automatic Similarity (Proposed)	Automatic Simplicity (Baseline)	Automatic Similarity (Proposed)
No CWI	**22.92%**	22.13% (-0.79%)	**39.22%***	37.87% (-1.35%)	**90.21%***	77.71% (-12.50%)
Generic CWI	25.43%	**27.91%** (+2.48%)	37.26%	**46.15%*** (+8.89%)	**94.41%***	86.75% (-7.66%)
Gold PCWI	28.30%	**34.48%*** (+6.18%)	41.14%	**57.03%*** (+15.89%)	100%	100% (same)
Auto PCWI	25.09%	**34.47%*** (+9.38%)	36.63%	**50.79%*** (+14.16%)	**94.11%***	91.54% (-2.57%)
- *low only*	24.11%	**29.53%*** (+5.42%)	34.96%	**46.41%*** (+11.45%)	**91.78%***	86.96% (-4.82%)
- *high only*	26.21%	**40.10%*** (+13.89%)	38.54%	**55.79%*** (+17.25%)	96.77%	96.77% (same)

Table 4: LS performance using automatic ranking and various CWI filtering methods on the dataset in Section 5.3. An asterisk means the improvement over the other ranking is significant ($p < 0.05$ by McNemar's test).

ous experiment, when using automatic predictions of vocabulary knowledge, similarity ranking produced more complex words than simplicity ranking. The noise in ranking, however, reduced the performance gap between them. Similarity ranking trailed simplicity ranking by only -2.57% (Filtering="Auto PCWI"), compared to -8.30% in the first experiment in Table 3.

Semantic faithfulness. As can be expected, the use of automatic similarity ranking in this experiment led to a substantial drop in semantic faithfulness. With gold vocabulary knowledge prediction (Filtering="Gold PCWI"), the proposed method yielded 34.48% for Precision$_1$ and 57.03% for Precision$_2$. Even though similarity ranking was performed with a simple WordNet-based method, it still significantly outperformed a strong baseline of simplicity ranking; the absolute improvement was +6.18% for Precision$_1$ and +15.89% for Precision$_2$. When using automatic vocabulary knowledge prediction (Filtering="Auto PCWI"), performance degraded for both ranking methods, but similarity ranking continued to significantly outperform simplicity. It secured an absolute improvement of +9.38% for Precision$_1$ and +14.16% for Precision$_2$.

Impact of language proficiency. As seen from the results obtained with automatic prediction of vocabulary knowledge (Filtering="Auto PCWI"), the effect of language proficiency is similar to the trend observed in the first experiment. Overall, the similarity ranking method led to a +9.38% gain in semantic faithfulness (in terms of Precision$_1$), at the cost of a 2.57% increase in word complexity (i.e., a 2.57% loss in Precision$_{all}$).

In terms of semantic faithfulness, high-proficiency users (as defined in Section 5.2) experienced a gain of 13.89% in Precision$_1$ (Filtering="Auto PCWI - high only"), compared to a gain of only 5.42% for low-proficiency users (Filtering="Auto PCWI - low only"). This gap also held for Precision$_2$.

In terms of word complexity, similarity ranking caused no degradation in Precision$_{all}$ for high-proficiency users (Filtering="Auto PCWI - high only"), but a 4.82% loss for low-proficiency users (Filtering="Auto PCWI - low only").

Thus, the overall impact of language proficiency remained the same with automatic ranking. Similar to the first experiment, the higher the users' language proficiency, the more they benefited from the proposed method. More advanced users generally experienced larger gains in semantic faithfulness, with a relatively small trade-off in word complexity. Both the gain and degradation are however smaller in magnitude.

9 Conclusion

We have proposed a novel, two-step Substitution Ranking algorithm that optimizes the trade-off between semantic faithfulness and word complexity for the lexical simplification (LS) task, by selecting the non-complex candidate that is the closest synonym to the target word. Experiments suggest that this algorithm leads to significantly enhanced semantic faithfulness in the LS system output, at the price of a relatively small increase in word complexity. For higher-proficiency users, this approach is especially beneficial because their larger vocabulary makes it more likely that the system can select a high-quality synonym that is more difficult but still known to them.

Acknowledgment

This work was partially 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 the CityU Internal Funds for External Grant Schemes (#9678104).

References

Mahmoud Azab, Chris Hokamp, and Rada Mihalcea. 2015. Using Word Semantics To Assist English as a Second Language Learners. In *Proc. HLT-NAACL: Demonstrations*.

J. De Belder and M. F. Moens. 2010. Text Simplification for Children. In *Proc. SIGIR Workshop on Accessible Search Systems*.

Joachim Bingel, Gustavo H. Paetzold, and Anders Søgaard. 2018. Lexi: A Tool of Adaptive, Personalized Text Simplification. In *Proc. COLING*.

Thorsten Brants and Alex Franz. 2006. The google web 1t 5-gram corpus version 1.1. In *LDC2006T13*.

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, Yusuke Miyao, Hidekazu Oiwa, Issei Sato, and Hiroshi Nakagawa. 2014. Formalizing Word Sampling for Vocabulary Prediction as Graph-based Active Learning. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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. International Conference on Computational Linguistics (COLING)*.

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.

Noemie Elhadad. 2006. Comprehending technical texts: Predicting and defining unfamiliar terms. In *AMIA annual symposium proceedings*, volume 2006, page 239. American Medical Informatics Association.

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.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What Substitutes Tell Us – Analysis of an “All-Words” Lexical Substitution Corpus. In *Proc. EACL*.

John Lee and Chak Yan Yeung. 2018. Personalizing Lexical Simplification. In *Proc. International Conference on Computational Linguistics (COLING)*.

Anne-Laure Ligozat, Anne Garcia-Fernandez, Cyril Grouin, and Delphine Bernhard. 2012. Annlor: A Naive Notation-System for Lexical Outputs Ranking. In *Proc. 6th International Workshop on Semantic Evaluation*.

Diana McCarthy and Roberto Navigli. 2009. The English Lexical Substitution Task. *Language Resources and Evaluation*, 43:139–159.

Gustavo H. Paetzold and Lucia Specia. 2015. LEXenstein: A Framework for Lexical Simplification. In *Proc. ACL-IJCNLP System Demonstrations*.

Gustavo H. Paetzold and Lucia Specia. 2016a. Anita: An Intelligent Text Adaptation Tool. In *Proc. COLING: System Demonstrations*.

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

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

Ellie Pavlick and Chris Callison-Burch. 2016. Simple PPDB: A Paraphrase Database for Simplification. In *Proc. ACL*.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better Paraphrase Ranking, Fine-grained Entailment Relations, Word Embeddings, and Style Classification. In *Proc. ACL*.

Sarah E. Petersen and Mari Ostendorf. 2007. Text Simplification for Language Learners: A Corpus Analysis. In *Proc. SlaTE*.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or Help? Text Simplification Strategies for People with Dyslexia. In *Proc. 10th International Cross-Disciplinary Conference on Web Accessibility*.

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

Ravi Sinha and Rada Mihalcea. 2014. Explorations in Lexical Sample and All-words Lexical Substitution. *Natural Language Engineering*, 20(1):99–129.

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 and Chris Biemann. 2018. Par4Sim — Adaptive Paraphrasing for Text Simplification. In *Proc. COLING*.

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 *ISBMDA 2005, LNBI 3745*, pages 184–192.