Putting it Simply: a Context-Aware Approach to Lexical Simplification

Or Biran
Computer Science
Columbia University
New York, NY 10027
ob2008@columbia.edu

Samuel Brody
Communication & Information
Rutgers University
New Brunswick, NJ 08901
sdbrody@gmail.com

Noémie Elhadad
Biomedical Informatics
Columbia University
New York, NY 10032
noemie@dbmi.columbia.edu

Abstract

We present a method for lexical simplification. Simplification rules are learned from a comparable corpus, and the rules are applied in a context-aware fashion to input sentences. Our method is unsupervised. Furthermore, it does not require any alignment or correspondence among the complex and simple corpora. We evaluate the simplification according to three criteria: preservation of grammaticality, preservation of meaning, and degree of simplification. Results show that our method outperforms an established simplification baseline for both meaning preservation and simplification, while maintaining a high level of grammaticality.

1 Introduction

The task of simplification consists of editing an input text into a version that is less complex linguistically or more readable. Automated sentence simplification has been investigated mostly as a preprocessing step with the goal of improving NLP tasks, such as parsing (Chandrasekar et al., 1996; Siddharthan, 2004; Jonnalagadda et al., 2009), semantic role labeling (Vickrey and Koller, 2008) and summarization (Blake et al., 2007). Automated simplification can also be considered as a way to help end users access relevant information, which would be too complex to understand if left unedited. As such, it was proposed as a tool for adults with aphasia (Carroll et al., 1998; Devlin and Unthank, 2006), hearing-impaired people (Daelemans et al., 2004), readers with low-literacy skills (Williams and Reiter, 2005), individuals with intellectual disabilities (Huenerfauth et al., 2009), as well as health consumers looking for medical information (Elhadad and Sutaria, 2007; Deléger and Zweigenbaum, 2009).

INPUT: In 1900, Omaha was the center of a national uproar over the kidnapping of Edward Cudahy, Jr., the son of a local meatpacking **magnate**.
CANDIDATE RULES:
{magnate → king} {magnate → businessman}
OUTPUT: In 1900, Omaha was the center of a national uproar over the kidnapping of Edward Cudahy, Jr., the son of a local meatpacking **businessman**.

Figure 1: Input sentence, candidate simplification rules, and output sentence.

Simplification can take place at different levels of a text – its overall document structure, the syntax of its sentences, and the individual phrases or words in a sentence. In this paper, we present a sentence simplification approach, which focuses on lexical simplification.[1] The key contributions of our work are (i) an unsupervised method for learning pairs of complex and simpler synonyms; and (ii) a context-aware method for substituting one for the other.

Figure 1 shows an example input sentence. The word *magnate* is determined as a candidate for simplification. Two learned rules are available to the simplification system (substitute *magnate* with *king* or with *businessman*). In the context of this sentence, the second rule is selected, resulting in the simpler output sentence.

Our method contributes to research on lexical simplification (both learning of rules and actual sentence simplification), a topic little investigated thus far. From a technical perspective, the task of lexical simplification bears similarity with that of para-

[1] Our resulting system is available for download at http://www.cs.columbia.edu/ ob2008/

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 496–501,
Portland, Oregon, June 19-24, 2011.

phrase identification (Androutsopoulos and Malakasiotis, 2010) and the SemEval-2007 English Lexical Substitution Task (McCarthy and Navigli, 2007). However, these do not consider issues of readability and linguistic complexity. Our methods leverage a large comparable collection of texts: English Wikipedia[2] and Simple English Wikipedia[3]. Napoles and Dredze (2010) examined Wikipedia Simple articles looking for features that characterize a simple text, with the hope of informing research in automatic simplification methods. Yatskar et al. (2010) learn lexical simplification rules from the edit histories of Wikipedia Simple articles. Our method differs from theirs, as we rely on the two corpora as a whole, and do not require any aligned or designated simple/complex sentences when learning simplification rules.[4]

2 Data

We rely on two collections – English Wikipedia (EW) and Simple English Wikipedia (SEW). SEW is a Wikipedia project providing articles in Simple English, a version of English which uses fewer words and easier grammar, and which aims to be easier to read for children, people who are learning English and people with learning difficulties. Due to the labor involved in simplifying Wikipedia articles, only about 2% of the EW articles have been simplified.

Our method does not assume any specific alignment or correspondance between individual EW and SEW articles. Rather, we leverage SEW only as an example of an in-domain simple corpus, in order to extract word frequency estimates. Furthermore, we do not make use of any special properties of Wikipedia (e.g., edit histories). In practice, this means that our method is suitable for other cases where there exists a simplified corpus in the same domain.

The corpora are a snapshot as of April 23, 2010. EW contains 3,266,245 articles, and SEW contains 60,100 articles. The articles were preprocessed as follows: all comments, HTML tags, and Wiki links were removed. Text contained in tables and figures was excluded, leaving only the main body text of the article. Further preprocessing was carried out with the Stanford NLP Package[5] to tokenize the text, transform all words to lower case, and identify sentence boundaries.

3 Method

Our sentence simplification system consists of two main stages: rule extraction and simplification. In the first stage, simplification rules are extracted from the corpora. Each rule consists of an ordered word pair $\{\textit{original} \rightarrow \textit{simplified}\}$ along with a score indicating the similarity between the words. In the second stage, the system decides whether to apply a rule (i.e., transform the original word into the simplified one), based on the contextual information.

3.1 Stage 1: Learning Simplification Rules

3.1.1 Obtaining Word Pairs

All content words in the English Wikipedia Corpus (excluding stop words, numbers, and punctuation) were considered as candidates for simplification. For each candidate word w, we constructed a context vector CV_w, containing co-occurrence information within a 10-token window. Each dimension i in the vector corresponds to a single word w_i in the vocabulary, and a single dimension was added to represent any number token. The value in each dimension $CV_w[i]$ of the vector was the number of occurrences of the corresponding word w_i within a ten-token window surrounding an instance of the candidate word w. Values below a cutoff (2 in our experiments) were discarded to reduce noise and increase performance.

Next, we consider candidates for substitution. From all possible word pairs (the Cartesian product of all words in the corpus vocabulary), we first remove pairs of morphological variants. For this purpose, we use MorphAdorner[6] for lemmatization, removing words which share a common lemma. We also prune pairs where one word is a prefix of the other and the suffix is in {s, es, ed, ly, er, ing}. This handles some cases which are not covered by MorphAdorner. We use WordNet (Fellbaum, 1998) as a primary semantic filter. From all remaining word pairs, we select those in which the second word, in

[2] http://en.wikipedia.org

[3] http://simple.wikipedia.org

[4] Aligning sentences in monolingual comparable corpora has been investigated (Barzilay and Elhadad, 2003; Nelken and Shieber, 2006), but is not a focus for this work.

[5] http://nlp.stanford.edu/software/index.shtml

[6] http://morphadorner.northwestern.edu

its first sense (as listed in WordNet)[7] is a synonym or hypernym of the first.

Finally, we compute the cosine similarity scores for the remaining pairs using their context vectors.

3.1.2 Ensuring Simplification

From among our remaining candidate word pairs, we want to identify those that represent a complex word which can be replaced by a simpler one. Our definition of the complexity of a word is based on two measures: the *corpus complexity* and the *lexical complexity*. Specifically, we define the *corpus complexity* of a word as

$$C_w = \frac{f_{w,English}}{f_{w,Simple}}$$

where $f_{w,c}$ is the frequency of word w in corpus c, and the *lexical complexity* as $L_w = |w|$, the length of the word. The final complexity χ_w for the word is given by the product of the two.

$$\chi_w = C_w \times L_w$$

After calculating the complexity of all words participating in the word pairs, we discard the pairs for which the first word's complexity is lower than that of the second. The remaining pairs constitute the final list of substitution candidates.

3.1.3 Ensuring Grammaticality

To ensure that our simplification substitutions maintain the grammaticality of the original sentence, we generate grammatically consistent rules from the substitution candidate list. For each candidate pair *(original, simplified)*, we generate all consistent forms $(f_i(original), f_i(substitute))$ of the two words using MorphAdorner. For verbs, we create the forms for all possible combinations of tenses and persons, and for nouns we create forms for both singular and plural.

For example, the word pair *(stride, walk)* will generate the form pairs *(stride, walk), (striding, walking), (strode, walked)* and *(strides, walks)*. Significantly, the word pair *(stride, walked)* will generate exactly the same list of form pairs, eliminating the original ungrammatical pair.

Finally, each pair *(f_i(original), f_i(substitute))* becomes a rule $\{f_i(original) \rightarrow f_i(substitute)\}$, with weight *Similarity(original, substitute)*.

3.2 Stage 2: Sentence Simplification

Given an input sentence and the set of rules learned in the first stage, this stage determines which words in the sentence should be simplified, and applies the corresponding rules. The rules are not applied blindly, however; the context of the input sentence influences the simplification in two ways:

Word-Sentence Similarity First, we want to ensure that the more complex word, which we are attempting to simplify, was not used precisely because of its complexity - to emphasize a nuance or for its specific shade of meaning. For example, suppose we have a rule $\{Han \rightarrow Chinese\}$. We would want to apply it to a sentence such as *"In 1368 Han rebels drove out the Mongols"*, but to avoid applying it to a sentence like *"The history of the Han ethnic group is closely tied to that of China"*. The existence of related words like *ethnic* and *China* are clues that the latter sentence is in a specific, rather than general, context and therefore a more general and simpler hypernym is unsuitable. To identify such cases, we calculate the similarity between the target word (the candidate for replacement) and the input sentence as a whole. If this similarity is too high, it might be better not to simplify the original word.

Context Similarity The second factor has to do with ambiguity. We wish to detect and avoid cases where a word appears in the sentence with a different sense than the one originally considered when creating the simplification rule. For this purpose, we examine the similarity between the rule as a whole (including both the original and the substitute words, and their associated context vectors) and the context of the input sentence. If the similarity is high, it is likely the original word in the sentence and the rule are about the same sense.

3.2.1 Simplification Procedure

Both factors described above require sufficient context in the input sentence. Therefore, our system does not attempt to simplify sentences with less than seven content words.

[7] Senses in WordNet are listed in order of frequency. Rather than attempting explicit disambiguation and adding complexity to the model, we rely on the first sense heuristic, which is know to be very strong, along with contextual information, as described in Section 3.2.

Type	Gram.	Mean.	Simp.
Baseline	70.23(+13.10)%	55.95%	46.43%
System	77.91(+8.14)%	62.79%	75.58%

Table 1: Average scores in three categories: grammaticality (Gram.), meaning preservation (Mean.) and simplification (Simp.). For grammaticality, we show percent of examples judged as *good*, with *ok* percent in parentheses.

For all other sentences, each content word is examined in order, ignoring words inside quotation marks or parentheses. For each word w, the set of relevant simplification rules $\{w \rightarrow x\}$ is retrieved. For each rule $\{w \rightarrow x\}$, unless the replacement word x already appears in the sentence, our system does the following:

- Build the vector of sentence context $SCV_{s,w}$ in a similar manner to that described in Section 3.1, using the words in a 10-token window surrounding w in the input sentence.
- Calculate the cosine similarity of CV_w and $SCV_{s,w}$. If this value is larger than a manually specified threshold (0.1 in our experiments), *do not* use this rule.
- Create a common context vector $CCV_{w,x}$ for the rule $\{w \rightarrow x\}$. The vector contains all features common to both words, with the feature values that are the minimum between them. In other words, $CCV_{w,x}[i] = min(CV_w[i], CV_x[i])$. We calculate the cosine similarity of the common context vector and the sentence context vector:

$$ContextSim = cosine(CCV_{w,x}, SCV_{s,w})$$

If the context similarity is larger than a threshold (0.01), we *use* this rule to simplify.

If multiple rules apply for the same word, we use the one with the highest context similarity.

4 Experimental Setup

Baseline We employ the method of Devlin and Unthank (2006) which replaces a word with its most frequent synonym (presumed to be the simplest) as our baseline. To provide a fairer comparison to our system, we add the restriction that the synonyms should not share a prefix of four or more letters (a baseline version of lemmatization) and use MorphAdorner to produce a form that agrees with that of the original word.

Type	Freq.	Gram.	Mean.	Simp.
Base	High	63.33(+20)%	46.67%	50%
Sys.	High	76.67(+6.66)%	63.33%	73.33%
Base	Med	75(+7.14)%	67.86%	42.86%
Sys.	Med	72.41(+17.25)%	75.86%	82.76%
Base	Low	73.08(+11.54)%	53.85%	46.15%
Sys.	Low	85.19(+0)%	48.15%	70.37%

Table 2: Average scores by frequency band

Evaluation Dataset We sampled simplification examples for manual evaluation with the following criteria. Among all sentences in English Wikipedia, we first extracted those where our system chose to simplify exactly one word, to provide a straightforward example for the human judges. Of these, we chose the sentences where the baseline could also be used to simplify the target word (i.e., the word had a more frequent synonym), and the baseline replacement was different from the system choice. We included only a single example (simplified sentence) for each rule.

The evaluation dataset contained 65 sentences. Each was simplified by our system and the baseline, resulting in 130 simplification examples (consisting of an *original* and a *simplified* sentence).

Frequency Bands Although we included only a single example of each rule, some rules could be applied much more frequently than others, as the words and associated contexts were common in the dataset. Since this factor strongly influences the utility of the system, we examined the performance along different frequency bands. We split the evaluation dataset into three frequency bands of roughly equal size, resulting in 46 *high*, 44 *med* and 40 *low*.

Judgment Guidelines We divided the simplification examples among three annotators [8] and ensured that no annotator saw both the system and baseline examples for the same sentence. Each simplification example was rated on three scales: **Grammaticality** - *bad*, *ok*, or *good*; **Meaning** - did the transformation preserve the original meaning of the sentence; and **Simplification** - did the transformation result in

[8]The annotators were native English speakers and were not the authors of this paper. A small portion of the sentence pairs were duplicated among annotators to calculate pairwise inter-annotator agreement. Agreement was moderate in all categories (Cohen's Kappa $= .350 - .455$ for Simplicity, $.475 - .530$ for Meaning and $.415 - .425$ for Grammaticality).

a simpler sentence.

5 Results and Discussion

Table 1 shows the overall results for the experiment. Our method is quantitatively better than the baseline at both grammaticality and meaning preservation, although the difference is not statistically significant. For our main goal of simplification, our method significantly ($p < 0.001$) outperforms the baseline, which represents the established simplification strategy of substituting a word with its most frequent WordNet synonym. The results demonstrate the value of correctly representing and addressing content when attempting automatic simplification.

Table 2 contains the results for each of the frequency bands. Grammaticality is not strongly influenced by frequency, and remains between 80-85% for both the baseline and our system (considering the *ok* judgment as positive). This is not surprising, since the method for ensuring grammaticality is largely independent of context, and relies mostly on a morphological engine. Simplification varies somewhat with frequency, with the best results for the medium frequency band. In all bands, our system is significantly better than the baseline. The most noticeable effect is for preservation of meaning. Here, the performance of the system (and the baseline) is the best for the medium frequency group. However, the performance drops significantly for the low frequency band. This is most likely due to sparsity of data. Since there are few examples from which to learn, the system is unable to effectively distinguish between different contexts and meanings of the word being simplified, and applies the simplification rule incorrectly.

These results indicate our system can be effectively used for simplification of words that occur frequently in the domain. In many scenarios, these are precisely the cases where simplification is most desirable. For rare words, it may be advisable to maintain the more complex form, to ensure that the meaning is preserved.

Future Work Because the method does not place any restrictions on the complex and simple corpora, we plan to validate it on different domains and expect it to be easily portable. We also plan to extend our method to larger spans of texts, beyond individual words.

References

Androutsopoulos, Ion and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research* 38:135–187.

Barzilay, Regina and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proc. EMNLP*. pages 25–32.

Blake, Catherine, Julia Kampov, Andreas Orphanides, David West, and Cory Lown. 2007. Query expansion, lexical simplification, and sentence selection strategies for multi-document summarization. In *Proc. DUC*.

Carroll, John, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplication of english newspaper text to assist aphasic readers. In *Proc. AAAI Workshop on Integrating Artificial Intelligence and Assistive Technology*.

Chandrasekar, R., Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *Proc. COLING*.

Daelemans, Walter, Anja Hthker, and Erik Tjong Kim Sang. 2004. Automatic sentence simplification for subtitling in Dutch and English. In *Proc. LREC*. pages 1045–1048.

Deléger, Louise and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proc. Workshop on Building and Using Comparable Corpora*. pages 2–10.

Devlin, Siobhan and Gary Unthank. 2006. Helping aphasic people process online information. In *Proc. ASSETS*. pages 225–226.

Elhadad, Noemie and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Proc. ACL BioNLP Workshop*. pages 49–56.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Huenerfauth, Matt, Lijun Feng, and Noémie Elhadad. 2009. Comparing evaluation techniques

for text readability software for adults with intellectual disabilities. In *Proc. ASSETS*. pages 3–10.

Jonnalagadda, Siddhartha, Luis Tari, Jörg Hakenberg, Chitta Baral, and Graciela Gonzalez. 2009. Towards effective sentence simplification for automatic processing of biomedical text. In *Proc. NAACL-HLT*. pages 177–180.

McCarthy, Diana and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proc. SemEval*. pages 48–53.

Napoles, Courtney and Mark Dredze. 2010. Learning simple wikipedia: a cogitation in ascertaining abecedarian language. In *Proc. of the NAACL-HLT Workshop on Computational Linguistics and Writing*. pages 42–50.

Nelken, Rani and Stuart Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proc. EACL*. pages 161–166.

Siddharthan, Advaith. 2004. Syntactic simplification and text cohesion. Technical Report UCAM-CL-TR-597, University of Cambridge, Computer Laboratory.

Vickrey, David and Daphne Koller. 2008. Applying sentence simplification to the CoNLL-2008 shared task. In *Proc. CoNLL*. pages 268–272.

Williams, Sandra and Ehud Reiter. 2005. Generating readable texts for readers with low basic skills. In *Proc. ENLG*. pages 127–132.

Yatskar, Mark, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Proc. NAACL-HLT*. pages 365–368.