SimpLe: Lexical Simplification using Word Sense Disambiguation

Nikolay YAKOVETS [a,1], Ameeta AGRAWAL [a]
[a] *Department of Computer Science and Engineering, York University, Canada*

Abstract. Sentence simplification aims to reduce the reading complexity of a sentence by incorporating more accessible vocabulary and sentence structure. In this chapter we examine the process of lexical substitution and particularly the role that word sense disambiguation plays in this task. Most previous work substitutes difficult words using a predefined dictionary. We present the challenges faced during lexical substitution and how it can be improved by disambiguating the word within its context. We provide empirical results which show that our method creates simplifications that significantly reduce the reading difficulty of the input text while maintaining its grammaticality and preserving its meaning.

Keywords. lexical simplification, sentence simplification, word sense disambiguation.

1. Introduction

Sentence simplification is a task that reduces the reading complexity of text while maintaining its grammaticality and preserving its meaning. Given an input sentence, the aim is to output a sentence which is easier to read with a simpler vocabulary structure. An example is shown in Table 1. The input sentence consists of several words where initially each word is a potential candidate for substitution. If a simpler and more frequently synonym is identified, then the candidate word is replaced with the target synonym.

Sentence simplification is usually used to preprocess text for Natural Language Processing tasks such as parsing [1–3] and summarization [4]. Recently, it has been used to simplify complex information into easily understandable and accessible text [5]. Similar to work presented in Chapter 5 of this book, sentence simplification has been proposed as an aide for people with disabilities. In particular, it can help people with aphasia [6, 7] and readers with low literacy skills [8].

From a technical perspective, the task of simplification is related to, but different from paraphrase extraction [9]. We must not only have access to paraphrases but also be able to combine them to generate new, simpler sentences by

[1] Corresponding Author: Nikolay Yakovets, Department of Computer Science and Engineering, York University, CSE 1003, 4700 Keele St, M3J1P3, Toronto Canada; E-mail: hush@cse.yorku.ca.

INPUT:	It is a virtue hitherto nameless to us, and which we will venture to call 'humanism'
OUTPUT:	It is a virtue yet unknown to us, and which we will guess to call 'humanism'

Table 1. Sample input and output sentences

addressing issues of readability and linguistic complexity. The task is also distinct from sentence compression as it aims to render a sentence more accessible while preserving its meaning. On contrary, compression unavoidably leads to some information loss as it creates shorter sentences without necessarily reducing complexity. In fact, sentence simplification may result in longer rather than shorter output.

In general, text can be simplified at various levels of granularity - overall document, syntax of the sentences, individual phrases or words in a sentence. In this chapter, we present a sentence simplification approach using lexical substitution. We use an unsupervised method for replacing complex words with simpler synonyms by employing word sense disambiguating techniques to preserve the original meaning of the sentence.

2. Related Work

Due to its potential various applications, the task of sentence simplification has recently started to garner a lot of research attention. Most previous approaches simplify text at lexical level by substituting difficult words by more common WordNet synonyms or paraphrases found in a predefined dictionary [10, 11].

More recently, a variety of linguistic resources such as WordNet and crowd-sourced corpora such as English Wikipedia (EW) and Simple English Wikipedia (SEW) have received some attention as useful resources for text simplification. SEW serves as a large repository of simplified language. It uses fewer words and simpler grammar than the ordinary English Wikipedia and is aimed at non-native English speakers, children, translators and people with learning disabilities or low reading proficiency. Due to the labour involved in simplifying Wikipedia articles, only about 2% of the EW articles have been simplified.

[12] have explored data-driven methods to learn lexical simplification rules based on the edits identified in the revision histories of EW and SEW. However, they only provide a list of the top phrasal simplifications and do not utilize them in an end-to-end simplification system.

[13] also leverage the large comparable collection of texts from EW and SEW. However, unlike [12], they rely on the two corpora as a whole and do not require any specific alignment or correspondence between individual EW and SEW articles. Our method differs from [13] as we employ word sense disambiguation to find the most appropriate substitution word using WordNet. This may result in a synonym, which is not necessarily the first sense in WordNet as opposed to relying solely on the first sense heuristic technique.

Zhu et al. proposed the first statistical text simplification model in their paper [14] published in 2010. Their tree transformation was based on techniques from statistical machine translation (SMT) [15–17]. It integrally covered four rewrite operations, namely substitution, reordering, splitting, and deletion. They used Wikipedia-Simple Wikipedia as a complex-simple parallel dataset to learn the parameters of their model by iteratively applying an expectation maximization (EM) algorithm. The training process was sped up by using a method based on monolingual word mapping. Finally, they used a greedy strategy based on the highest outside probability to generate the simplified sentences.

In 2011, Woodsend et al. proposed both lexical and syntactical simplification approaches [18] based on quasi-synchronous grammar (QG) [19], a formalism that can naturally capture structural mismatches and complex rewrite operations. Woodsend et al. argue that their model finds globally optimal simplifications without resorting to heuristics or approximations during the decoding process. Their work joins others in using EW-SEW to extract data appropriate for model training. They evaluated their model both automatically using FKGL, BLEU and TERp scores and manually by human judgments against gold standard sentences. They found their model to produce the highest human rated simplifications among others. They also reported that while Zhu et al.'s model achieved the best FKGL automatic score, it was the least grammatical model by human judgment.

Some researchers treated text simplification as English-to-English translation problem. In 2011, Coster et al. proposed a parallel corpora extraction technique for EW-SEW [20] and a translation model for text simplification [21]. The authors use a modified version of statistical machine translation system Moses [22] to perform the simplification. They modify Moses to model phrasal deletion that commonly occurs in text simplification. Coster et al. did not compare their model to other state-of-the-art simplification systems. Instead, they chose to evaluate their model against two other text compression systems. They perform the evaluation using BLEU, word-F1 and SSA scores, but fail to provide text readability scores such as FKGL. Finally, they report that their model ranks highest amongst the systems compared according the metrics they used.

3. Sentence Simplification Model

Our sentence simplification model takes a text as an input and processes it sentence-by-sentence to create a text that is simpler to read. This process consists of two primary phases: Word Sense Disambiguation (WSD), implemented using Perl and Lexical Simplification (LS), implemented using Java. The system overview is presented in Figure 1.

3.1. Disambiguation

WSD is the process of identifying which sense of a word (i.e. meaning) is used in a sentence when the word has multiple meanings (polysemy). We utilize SenseRelate (AllWords version) Perl toolkit that uses measures of semantic similarity and relatedness to assign a meaning to every content word in a text [23]. After initial

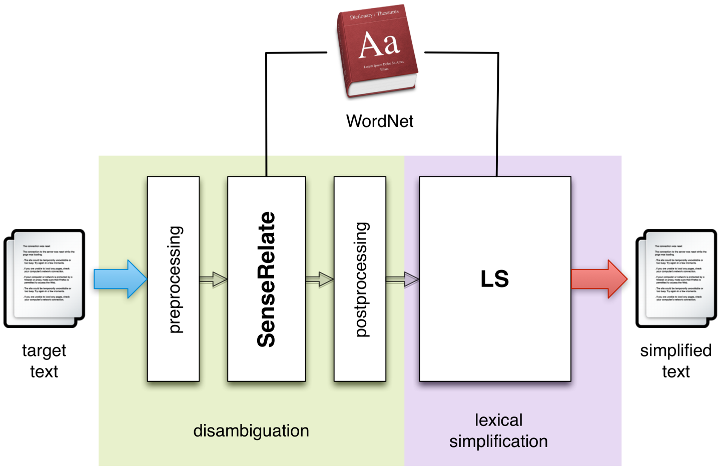

Figure 1. System Architecture

preprocessing of the source text (removal of any non-alphanumeric text, excluding HTML tags, tables and figures and splitting text into sentences), it is used as an input to SenseRelate disambiguator. The output from SenseRelate consists of several files containing for of each disambiguated word, its base form, its part-of-speech and its sense as found in WordNet. WordNet is a large lexical database where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms called synsets. These synsets are interlinked by means of semantic and lexical relations. Finally, the output from SenseRelate is merged into a single file, which is used as an input for Lexical Simplification phase.

3.2. Lexical Simplification

The second stage, the LS, is the process of simplifying sentences at the lexical level after having identified potential substitutions for each source word. It is encapsulated by JavaFX desktop application, which takes as input the output of the previous WSD phase and produces simplified sentences. To perform the correct sentence simplification the goal of our system is to ensure that each replacement word: 1) has the same meaning as was intended in the original sentence; 2) is grammatically correct; and 3) is simpler than the candidate word it replaced. We discuss how SimpLe achieves these goals in the following subsections.

3.2.1. Preserved Meaning

We rely on Word Sense Disambiguation to ensure that the replacement word has the same meaning as intended in the original sentence. For each candidate word, the Disambiguation phase gives us its base form, its part-of-speech and its sense in WordNet. We use this meta-data to extract all synonyms of the candidate word from WordNet in the correct sense and part-of-speech. This way we ensure that the possible replacement words preserve the meaning of the original candidate.

3.2.2. Correct Grammaticality

The replacement synonyms are obtained from WordNet in their respective base forms. In our work, we make sure that the replacement synonym appears in the same form as the candidate appeared in the original sentence. For example, consider a candidate word "espouses". Based on WordNet usage counts and word lengths we choose synonym "to marry" as a replacement. We build a collection of all possible form pairs: (to espouse, to marry), (espouses, marries), (espoused, married), etc. From this collection, we choose the replacement so that it matches the form of the candidate.

3.2.3. Ensuring Simplification

Once we obtain the list of replacement synonyms, we need to find one that is simpler than the original candidate word. In our work, we calculate the complexity of a word using its length and WordNet usage count. Specifically, we consider the word to be simpler than other words if it has the highest usage count and is shorter than other words. In this manner we identify the simplest candidate replacement, if it exists.

4. Experiments and Evaluation

In this section we present our experimental setup for assessing the performance of the simplification model described above. To evaluate the simplicity of the resulting simplified sentences, we ran some preliminary experiments to gauge the readability of the output text.

The test corpus comprises of 2000 original sentences which we automatically extracted from 10 English Wikipedia articles on various topics such as linguistics, humanity, technology and so on. We evaluated our model, which takes in an original sentence and outputs a simplified sentence and compared our system against two other systems - Spencer[2] and Biran et al.[3]

Spencer is a simple baseline that uses solely lexical simplifications. They assembled a list of simple words and simplifications using a combination of dictionaries and manual effort. They provide a list of 17,900 simple words - words that do not need further simplification - and a list of 2000 transformation pairs.

Biran et al. also perform lexical simplification but they start by extracting simplification rules from EW and SEW. Each rule consists of an ordered word pair (original → simplified) along with a score indicating the similarity between the words. Based on the contextual information, the system then decides whether to apply the rule.

Another obvious idea that we tried was to treat sentence simplification as an English-to-English translation problem and use an off-the-shelf system like Moses[4] for the task. But Moses performed poorly as it generated output identical to the source in most cases. We also thought of extending this idea to translate

[2] http://www.spencerwaterbed.com/soft/simple/

[3] http://www.cs.columbia.edu/ orb/

[4] http://www.statmt.org/moses/

Source^1:	By extension academia has come to mean the cultural accumulation of knowledge, its development and transmission across generations.
Biran:	By extension academia has come to mean the cultural accumulation of knowledge, its development and transmission across generations.
Spencer:	By extension academia has come to mean the cultural group knowledge, its development and message across generations.
SimpLe:	By extension academia has come to mean the cultural collection of knowledge, its growth and transmission across generations.
Source^2:	Secular humanism is a secular ideology which espouses reason, ethics and justice, specifically rejecting supernatural and religious dogma as a basis of morality.
Biran:	Secular humanism is a secular ideology which espouses reason, ethics and justice, specifically rejecting supernatural and religious dogma as a basis of morality.
Spencer:	Secular humanism is a secular ideology which espouses reason, ethics and justice, specifically rejecting supernatural and religious dogma as a basis of morality.
SimpLe:	Secular humanism is a layman ideology which marries reason, ethics and judge, specifically rejecting supernatural and religious dogma as a basis of morality.

Table 2. Comparison of Simplifications Produced

from an original English sentence into another language and back to English to see if the sentence is in any way simplified in the process due to dissimilar or limited vocabulary between the two languages. But two main problems with this approach arose: the lack of a good open source inter-lingual translation system and identifying which language pairs would result in meaningful simplification. However, this idea may have potential if explored at length.

Some example simplifications produced by SimpLe system as well as Spencer and Biran et al. systems are shown in Table 2. One thing which is evident is that SimpLe is able to simplify lexically not only nouns but also verb phrases in the correct tense as shown by simplified sentence 2.

Intuitively, the use of metrics for measuring the readability of the output text seems reasonable. We start with reporting our results using the well-known Flesch-Kincaid Grade Level index (FKGL) and the Flesch Reading Ease score (FRE). These methods were designed to indicate comprehension difficulty when reading a passage of contemporary academic English. Although they use the same core measures of word length and sentence length, they have different weighting factors. The aim is to get a higher score on the FRE test and a lower score on the

	FRE	FKGL	GFI	C-LI	ARI	SMOG
Original	17.1	14.9	16.7	17.1	14.5	15.3
Biran	18.1	14.8	16.5	16.9	14.3	15.1
Spencer	21.0	14.4	16.2	16.4	14.0	15.0
SimpLe	24.8	13.8	15.8	15.7	13.3	14.5

Table 3. Evaluation Results

FKGL test. The U.S. Department of Defense uses the FRE test as the standard test of readability for its documents and forms[5].

We also present comparison using four other readability scores, namely the Gunning fog index (GFI), Coleman-Liau index (C-LI), Automated Readability Index (ARI) and SMOG index. GFI estimates the years of formal education needed to understand the text on a first reading. The C-LI and ARI also approximate the U.S. grade level thought necessary to comprehend the text. Unlike most of the other indices however, these two indices rely on characters instead of syllables per word. The SMOG index is another widely used readability metric, particularly for checking health messages.

The results of our automatic evaluation are summarized in Table 3. The columns report the various readability scores of the source sentence (Original), the simplified sentence produced by Biran et al, by Spencer and finally by our SimpLe system. The goal is to get a high Flesch Reading Ease score as it signifies easier readability. For example, a childrens fairy tale book usually scores around 90, whereas legalese can range around 5. On the other hand, for FKGL, GFI, C-LI, ARI and SMOG, the goal is to get as low score as possible as that approximates the number of years of formal education needed to understand the sentence.

As can be seen, the original source sentence has the lowest FRE score and the highest score for all the other indices, which means it has the highest reading level. This is closely followed by Biran et al.'s system, which means that they have small simplifications done. Next on the ease of readability is Spencer system, which has significant improvement even though it works with a very limited fixed size dictionary. Lastly, the simplified output of our system SimpLe produces the lowest reading level and significantly outperforms the other two systems. It can be noticed that the results are consistent over all the readability metrics tested. These scores indicate that even simple rewriting using lexical substitution can considerably improve the readability of a sentence.

5. Conclusions and Future Work

This chapter examined the task of sentence simplification with focus on lexical substitution. Though several approaches have been proposed, to the best of our knowledge, none of them employed word sense disambiguation techniques

[5] http://law.onecle.com/florida/insurance/627.4145.html

when choosing the appropriate substitutions. We first disambiguate each candidate word and then use WordNet to find the most relevant synonym, which is simpler than the original candidate word.

We measured the ease of readability using several readability metrics and found significant improvement in our results as compared to other recently proposed approaches. This indicates that our system can be effectively used for simplification of words.

As an extension to our work, in the future we would like to get help from human evaluators to test the output of our system. Some future research directions include splitting of long-winded sentences into simpler ones possibly using chunking techniques and also restructuring the sentences to better reflect grammatical accuracy. We also plan to extend our method of lexical substitution to larger span of texts, beyond individual words. Another direction in which further research can be carried out is in the task of monolingual sentence alignment.

References

[1] R. Chandrasekar, C. Doran, and B. Srinivas. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 1041–1044. Association for Computational Linguistics, 1996.

[2] L. Feng. Text simplification: A survey. Technical report, CUNY, 2008.

[3] S. Jonnalagadda, L. Tari, J. Hakenberg, C. Baral, and G. Gonzalez. Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 177–180. Association for Computational Linguistics, 2009.

[4] C. Blake, J. Kampov, A. Orphanides, D. West, and C. Lown. Query expansion, lexical simplification and sentence selection strategies for multi-document summarization. In *Document understanding conference*, 2007.

[5] S.F. Martins. The right to understand, November 2011.

[6] J. Carroll, G. Minnen, Y. Canning, S. Devlin, and J. Tait. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Citeseer, 1998.

[7] S. Devlin and G. Unthank. Helping aphasic people process online information. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, pages 225–226. ACM, 2006.

[8] S. Williams and E. Reiter. Generating readable texts for readers with low basic skills. In *Proceedings of ENLG*, volume 5, page 140, 2005.

[9] R. Barzilay and K.R. Adviser-Mckeown. *Information fusion for multidocument summarization: paraphrasing and generation.* PhD thesis, 2003.

[10] K. Inui, A. Fujita, T. Takahashi, R. Iida, and T. Iwakura. Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 9–16. Association for Computational Linguistics, 2003.

[11] N. Kaji, D. Kawahara, S. Kurohash, and S. Sato. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 215–222. Association for Computational Linguistics, 2002.

[12] M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 365–368. Association for Computational Linguistics, 2010.

[13] Or Biran, Samuel Brody, and Noémie Elhadad. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the As-*

sociation for Computational Linguistics: Human Language Technologies: short papers - Volume 2, HLT '11, pages 496–501, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[14] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1353–1361, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[15] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[16] Kenji Yamada and Kevin Knight. A decoder for syntax-based statistical mt. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 303–310, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[17] Jonathan Graehl, Kevin Knight, and Jonathan May. Training tree transducers. *Comput. Linguist.*, 34(3):391–427, September 2008.

[18] Kristian Woodsend and Mirella Lapata. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 409–420, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[19] Dipanjan Das and Noah A. Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 468–476, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[20] William Coster and David Kauchak. Simple english wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 665–669, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[21] William Coster and David Kauchak. Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 1–9, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[22] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[23] T. Pedersen and V. Kolhatkar. Wordnet:: Senserelate:: Allwords: a broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics, companion volume: Demonstration session*, pages 17–20. Association for Computational Linguistics, 2009.