

Out in the Open: Finding and Categorising Errors in the Lexical Simplification Pipeline

Matthew Shardlow

School of Computer Science, University of Manchester
m.shardlow@cs.man.ac.uk

Abstract

Lexical simplification is the task of automatically reducing the complexity of a text by identifying difficult words and replacing them with simpler alternatives. Whilst this is a valuable application of natural language generation, rudimentary lexical simplification systems suffer from a high error rate which often results in nonsensical, non-simple text. This paper seeks to characterise and quantify the errors which occur in a typical baseline lexical simplification system. We expose 6 distinct categories of error and propose a classification scheme for these. We also quantify these errors for a moderate size corpus, showing the magnitude of each error type. We find that for 183 identified simplification instances, only 19 (10.38%) result in a valid simplification, with the rest causing errors of varying gravity.

Keywords: lexical simplification, text simplification, error annotation

1. Introduction

Text simplification comes naturally to many people. It is a task that people find easy, yet is innately difficult to model computationally. One popular form of simplification is ‘lexical simplification’ and is concerned solely with a text’s vocabulary. It follows the supposition that the readability and understandability of a text may be improved by replacing words which a reader does not understand with more familiar substitutions. For example, in the following sentence:

“Aristotle profoundly shaped mediaeval scholarship”

The words *profoundly* and *scholarship* may be transformed to *deeply* and *learning* respectively. Thus, improving the comprehensibility of the sentence for a low literacy reader. There are many groups that could benefit from simplification, with existing applications for people with aphasia (Devlin and Tait, 1998; Devlin and Unthank, 2006), deaf people (Inui et al., 2003), people with dyslexia (Bott et al., 2012), lay readers of technical medical language (Elhadad and Sutaria, 2007; Leroy et al., 2013) and low literacy readers (Watanabe et al., 2009). So far only a fraction of the potential uses have been explored and many are yet to come. Language is the medium of the information age and accessibility must be considered. Lexical simplification has the potential to improve accessibility, empowering readers and allowing texts to reach further.

When simplifying a text, certain considerations must be in place. The resulting text should convey the same meaning as its original, it should also be easier to read and understand. It should minimise errors, grammatical or otherwise, that were not present in the original text. Here, we encounter some problems with the automatic simplification of documents. Automatic processes seek to implement general rules which work correctly in the majority of cases. However, in some cases these rules will fail and the result of the simplification operation either will be more difficult to understand, convey a different meaning, or in the worst case become completely unintelligible.

Rudimentary lexical simplification systems suffer from a high error rate as shown later in this paper. This is characterised by a form of simplified text which is often very hard to understand. The structure and contributions of this paper are as follows:

- A basic simplification system modelled on the seminal work in the field (Devlin and Tait, 1998) is presented in Section 3.1.
- The categorisation of errors is shown in Section 3.2. A classification scheme is also proposed.
- The results of the classification for a moderate sized corpus are presented in Section 4. An inter-annotator study is also presented.

Related and future work are presented in Sections 2. and 6. respectively. An extended discussion of the results can be found in Section 5.

2. Background

Manual text simplification has existed for a long time (Blum and Levenston, 1978), however its automation is a recent endeavour. The first such system was developed to aid writers of technical aircraft manuals (Hoard et al., 1992), where a strict controlled natural language was necessary. Later, Chandrasekar and Srinivas (1997) developed a sentence simplifier capable of learning specialised simplification rules from a corpus.

The first notable lexical simplification system is the work of Devlin and Tait (1998) (which we have closely emulated). The system was developed as part of a wider simplification project called PSET (Carroll et al., 1998), which also incorporated syntactic simplification. The main target group of PSET was people with aphasia. PSET later developed into the HAPPI project (Devlin and Unthank, 2006), although no further advances were made to the lexical simplifier. The PSET project has influenced lexical simplification systems to the present date. Errors have historically been a factor of lexical simplification systems. PSET was found to produce

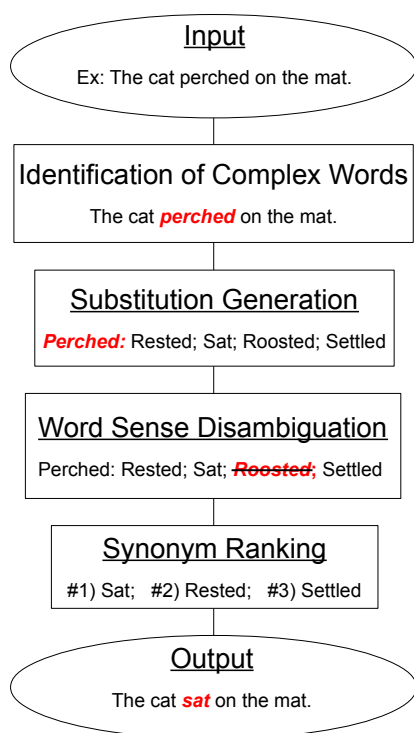


Figure 1: The lexical simplification pipeline. In the worked example the word ‘perched’ is transformed to sat. ‘Roosted’ is eliminated during the word sense disambiguation step as this does not apply properly in the context of ‘cat’.

strange sounding text (Pearce, 2001) and to change a text’s meaning (Lal and Ruger, 2002).

A typical lexical simplification system follows the pipeline shown in Figure 1. The worked example shows how a sample input sentence may be processed by the lexical simplification pipeline. This is heavily influenced by Devlin’s simplifier, the main addition being the word sense disambiguation step. Although other simplification systems may not have explicitly followed the pipeline shown in Figure 1, most efforts can be shown to fall under this framework.

The tasks of complex word identification and synonym ranking are highly similar as they both need a definition of lexical complexity. Complex word identification has been performed for technical medical language (Elhadad and Sutaria, 2007), with the aim of discovering which words are problematic for patients, however most lexical simplification systems only perform rudimentary identification. The two most prevalent techniques are either simplifying everything (Devlin and Tait, 1998; Thomas and Anderson, 2012) and seeing which operations are successful, or implementing a threshold over some lexical complexity value (Biran et al., 2011; Zeng et al., 2005) (such as word frequency). Ranking has seen renewed interest since the recent SemEval-2012 task 1 on lexical simplification (Specia et al., 2012), which asked participants to rank sets of synonyms according to their complexity in a given context. The winning system (Jauhar and Specia, 2012) incorporated the Google-Web1T frequencies (Brants and Franz,

2006) with a number of lexical and syntactic features using a support vector machine for classification. More recent work using the same data from the SemEval-2012 task 1 has focussed on the composition of the corpus from which the word frequencies are derived (Kauchak, 2013). It was possible to show that using frequencies from a simple corpus was better for synonym ranking than frequencies from a more complicated corpus.

Substitution generation has received very little attention for lexical simplification. English language systems typically use WordNet, whereas non-English language systems use some other comprehensive thesaurus. The Spanish language Simplext project (Saggion et al., 2011) recently produced a paper comparing thesaurus modules for their lexical simplification system (Saggion et al., 2013). The lack of substitution generation research is contrasted by a large amount of work towards reducing errors due to word sense disambiguation. Notable attempts have employed language models (De Belder et al., 2010), distributional semantics (Biran et al., 2011; Bott et al., 2012) and WordNet based classification (Thomas and Anderson, 2012).

3. Experimental Design

To investigate the types of errors prevalent in the lexical simplification pipeline, we built a simplification system similar to that described by Devlin and Tait (1998). As no standard system for lexical simplification exists, we chose not to apply any of the optimisations which have been proposed for Devlin’s simplifier. A corpus was created from the introductory lines of 115 news articles across varying topics. The original PSET project was designed for use with news text and we found the vocabulary in our corpus to be a suitable mix of linguistic difficulty. The system simplified each sentence in the corpus. To aid in the annotation process, the system printed a verbose transcript of each simplification operation. An annotation workflow, shown in Figure 2 was used to reduce subjectivity during the annotation. The simplifications were recorded and cross validated to ensure that categories were consistent. To further ensure consistency and reproducibility, we make all the raw data available via the LRE map.

3.1. Simplification System

The simplification system used in this research follows the structure shown in Figure 1. Below, we detail the design decisions taken for each step in the pipeline.

Complex Word Identification A threshold determined whether a word would require simplification. Every word with a Kučera-Francis frequency (Kučera and Francis, 1967) below five was considered for simplification. We chose to omit capitalised words as these generally denote named entities.

Substitution Generation WordNet was used for substitution generation. All wordforms from all the synsets associated with the target were conflated to give a list of candidates for substitution.

Word Sense Disambiguation No word sense disambiguation was applied, reflecting the majority of previous work.

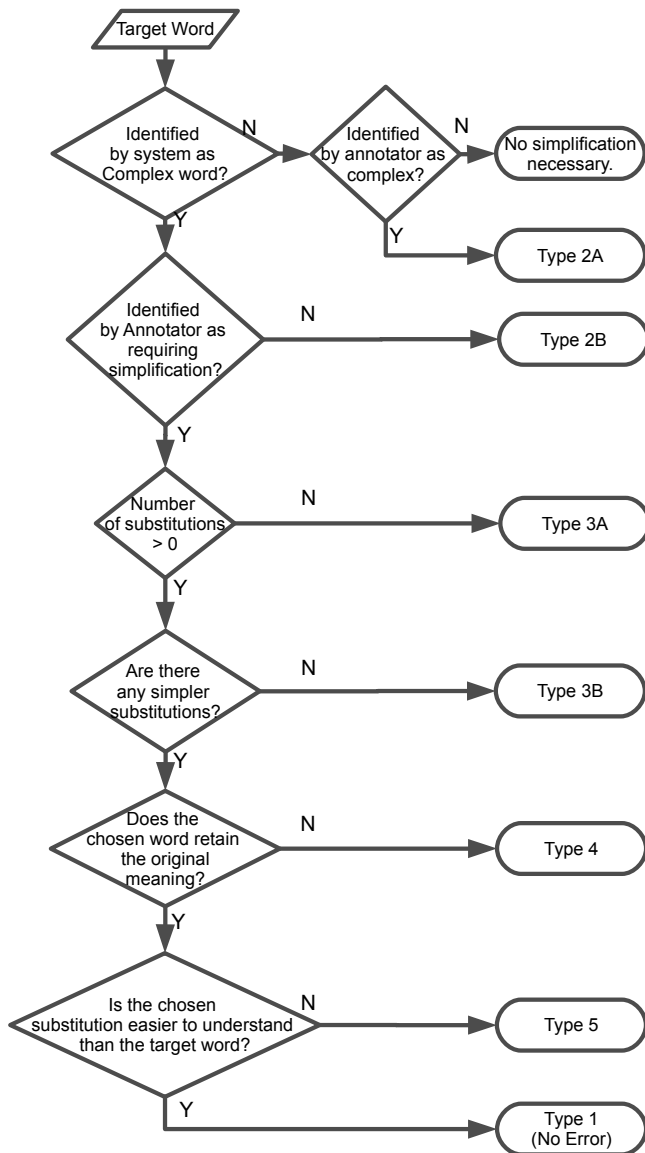


Figure 2: The annotation process used to determine the kinds of errors occurring during simplification operations.

Synonym Ranking We used the Kučera-Francis frequencies to order the candidate substitutions. The most frequent, and hence simplest, substitution was selected.

3.2. Annotation Workflow

The annotation workflow is shown in Figure 2. We choose to report the first error that occurs in each instance. This may mask further errors, as resolving an error early on in the pipeline does not guarantee safe passage through the later sections.

The error types mirror the stages of the lexical simplification pipeline, the first category being reserved for the state of no error. Upon analysis, it became clear that both complex word identification and substitution generation (types 2 and 3 respectively) could each be split into two further categories. These were labelled alphabetically (2A, 2B, 3A and 3B) to avoid confusion. The categories are described below:

Type 1: *No error.* The system successfully simplified this

word.

Type 2A: A *complex word* which was misidentified as a simple word.

Type 2B: A *simple word* which was misidentified as a complex word.

Type 3A: *No substitutions* available for the target word.

Type 3B: *No simplifying substitutions* available for the target word.

Type 4: *Word sense disambiguation error.* The meaning of the sentence has changed significantly.

Type 5: *Ranking Error.* A replacement which does not simplify the sentence has been selected.

4. Results

The annotation was undertaken in one sitting by the author. The workflow (Figure 2) was used to decide the category of each word. This required some interpretation of complex/simple by the annotator. As there is no standard automatic measure of lexical complexity, human judgement must be relied upon. To investigate the reliability of human judgement we also performed an inter-annotator agreement study, details are given in Section 4.1.

The results of the error annotation are shown in Table 1 and Figure 3. There are surprisingly few successful simplifications by the system. Notably, type 2 errors are the most frequent, indicating that complex word identification is often difficult. Type 2B errors are more frequent than 2A, indicating many false positives. Type 3 is the next most frequent error type, indicating that many words had either no substitutions, or none which would be useful in simplification.

Figure 3 shows that each of the stages of the lexical simplification pipeline exhibits fewer errors than the previous stage. To investigate whether each stage was having a masking effect, we analysed the performance of each stage individually. Figure 4 shows the error rate at each stage as the proportion of simplification operations which were evaluated at that stage. For example, at the first stage (complex word identification), there are 183 simplification operations. 119 of these resulted in error, giving this stage an error rate of 65.03%. At the next stage (substitution generation) there are 64¹ operations to take into account, of which 27 result in error, giving an error rate of 42.19%. The results of this indicate that each stage does indeed achieve a lower error rate than the previous stage. We can also observe that the difference in error rate between all the error types is much smaller than in Figure 3.

4.1. Inter-Annotator Agreement

To assess the reliability of using human judgement, three annotators were given the task of assigning error categories to a ten sentence sample. The sentences were also taken from the introductory lines of news text. There was no crossover with the main corpus. The annotators were given

¹183 - 119 = 64

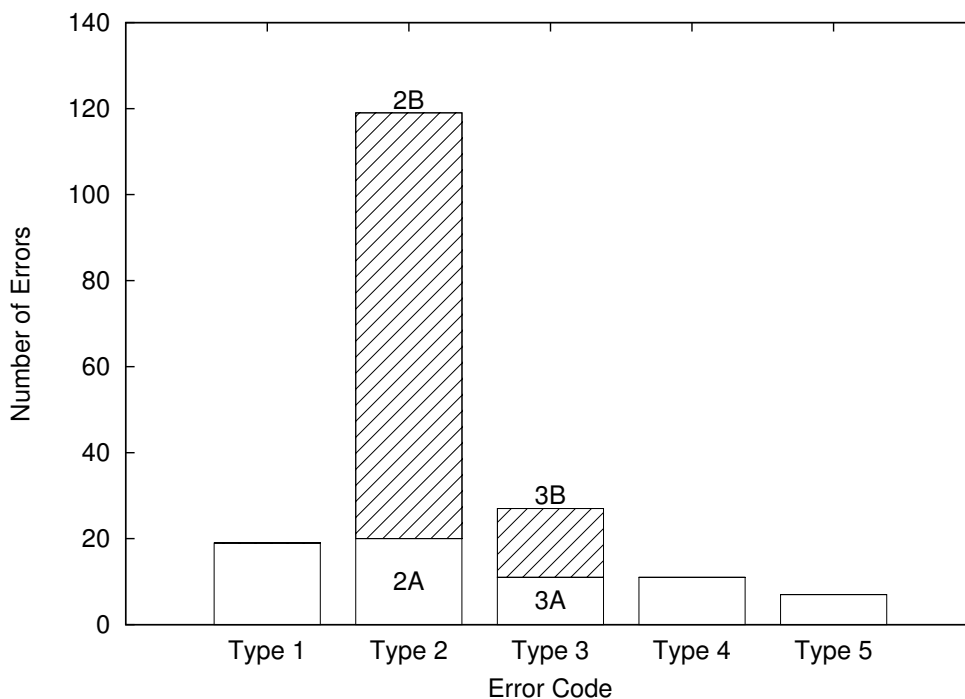


Figure 3: The distribution of errors between categories. N.B. The most common error is type 2, which is the first stage of the pipeline. Errors at later stages may be partially masked.

Description	Error Code	Amount
No error	Type 1	19
Word not identified as complex	Type 2A	20
Word incorrectly identified as complex	Type 2B	99
No substitutions found	Type 3A	11
No simpler substitutions found	Type 3B	16
Substitution changes word sense	Type 4	11
Substitution does not simplify	Type 5	7

Table 1: The raw error data, showing the number of errors assigned to each type. In total, 183 simplification operations were identified. 164 of these resulted in some form of error.

the transcript of the simplification system and a separate document to record their annotation. They were briefed on the task and shown examples of the annotations they would be making. Inter-annotator agreement was calculated using Fleiss’ kappa (Fleiss, 1971), which is an extension of Cohen’s kappa (Cohen, 1968) to more than two annotators. The kappa agreement of the annotators was 0.3556 ($p < 0.0001$). This indicates a moderate agreement between the annotators with the chance of this result occurring by random being insignificant. Although 0.3556 may seem low (1 is a total agreement, 0 is no agreement) this is not necessarily the case. Kappa agreement is strongly affected by the number of categories from which an annotator may choose. The annotators in this scheme had seven very distinct categories (one for each error type) to choose from and this will have contributed to the reduced kappa. Although the agreement could be improved by reducing the number of choices available to the annotators, this would reduce the descriptive power of the error categorisations.

5. Discussion

The low success rate exhibited by the system indicates that this form of simplification requires many modifications to produce coherent text. We deliberately refrained from implementing the obvious improvements to our system in order to determine the severity of the different error types in a baseline system. This section will discuss the raw results as well as suggesting mitigations for the different error types.

The high rate of type 2B errors implies that too many words were falsely considered for simplification, although this may have been altered by adjusting the threshold used in complex word identification. A reduction in 2B errors would likely have led to an increase in 2A errors. The basic filtering limited our system to only assessing uncapitalised words. However, some named entities were not picked up and stronger named entity recognition would have reduced the error rate. Some errors were also due to hyphenated multi-word expressions. These were not found in the Kučera-Francis frequencies or in WordNet and so were assigned zero frequency and no substitutions, despite being

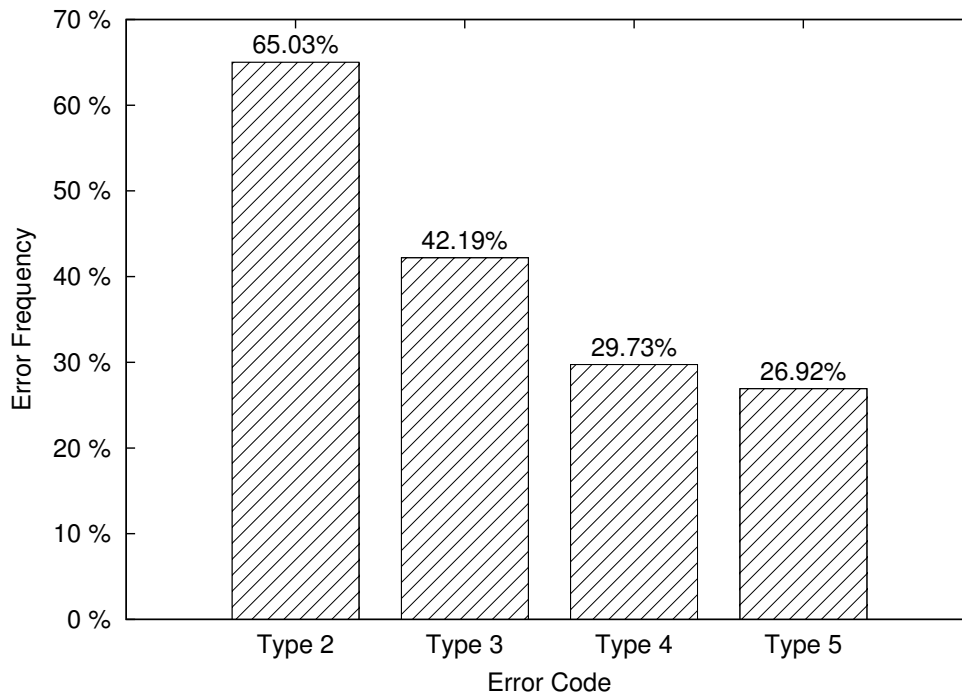
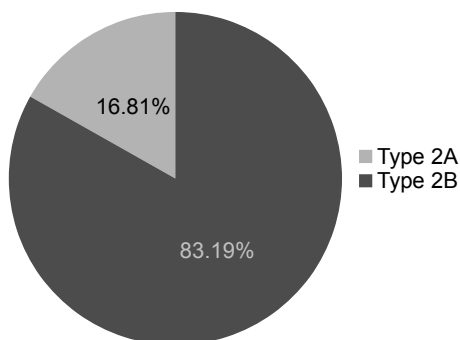
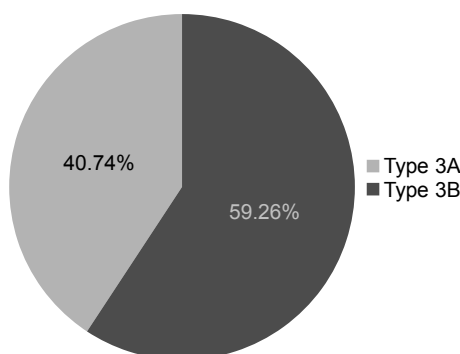


Figure 4: The percentage of errors occurring at each stage of the pipeline. A lower proportion of errors occur at each stage.



(a) Type 2A vs. Type 2B



(b) Type 3A vs. Type 3B

Figure 5: The sub distributions for type 2 and 3 errors.

easily understandable. Using a more comprehensive dictionary or some compound word splitting may have aided in this task.

Overall, a stronger notion of lexical complexity is required. The Kučera-Francis frequencies are an old resource and newer resources now exist which take their counts from

much larger corpora (Brysbaert and New, 2009; Brants and Franz, 2006). These may provide a more accurate notion of lexical complexity. Recently, work has used larger resources (Jauhar and Specia, 2012). To identify complex words, lexical simplification systems typically use either a thresholding approach, similar to that taken here (Zeng-Treitler et al., 2008; Elhadad, 2006), or a machine learning approach (Zeng et al., 2005). Very little work has taken place to recognise multi-word expressions for lexical simplification. This may be most manageable when simplification is required for specific domains.

The next most frequent error is type 3, those caused by a failure at the substitution generation stage. It may be intuitive to believe that the main failure would be that complex (and hence infrequent) words are not found in WordNet and hence have no valid substitutions (Error 3A). However, the results in Figure 5b indicate that Error 3B (the case of substitutions being generated, but none being sufficient to simplify the original word) is more frequent. This shows that as well as improving the scope of resources, it is also important to directly generate simpler synonyms for complex words. This has been previously explored with limited success (Yatskar et al., 2010; Biran et al., 2011).

Type 4 errors were caused by a word of the wrong sense being selected. This happens as one wordform may map to several senses. For example, ‘run’ may be used in the sentence: ‘The event will *run* for 3 days’; where ‘continue’ or ‘pass’ may be valid synonyms. ‘run’ may also be used in the sentence: ‘I took the car for a test *run*’, where ‘drive’ would be a valid synonym. If the simplification system does not know which sense of run is correct, then it must select all the possible synonyms for run, potentially resulting in error. It has been previously suggested that words requiring simplification will be infrequent and hence monosemous (Carroll et al., 1998), however this has proven empirically

not to be the case. This invokes the need for word sense disambiguation, a well established field in natural language processing. Some recent publications have sought to apply word sense disambiguation algorithms to the lexical simplification pipeline (Thomas and Anderson, 2012; De Belder et al., 2010), with limited success. Distributional semantics has also been employed as a disambiguation tool (Biran et al., 2011; Bott et al., 2012). The apparent disadvantage of applying out-of-the-box disambiguation tools is the reliance on the underlying resources. For example, if a tool is based on WordNet, then only senses from WordNet may be assigned, which may not be sufficient.

Ranking errors (type 5) occur when a word which is more difficult to understand is chosen over a word which is easier to understand. This is related to the first stage of complex word identification (and hence type 2 errors), underlying both of these is the idea of lexical complexity, which seeks to assign a difficulty value to each word. This permits a partial ordering over words and hence allows us to say which words are more complex than others. In reality, lexical complexity changes with context and word sense and so is very difficult to determine. The strategy of this paper has been to use word frequency as a measure of lexical complexity, assuming that more frequent words will be easier to understand, as they are encountered more often. However, many other factors can also affect lexical complexity (Specia et al., 2012).

It is surprising to note that type 2 and 3 errors far outweigh 4 and 5. Much of the research to date in lexical simplification has focussed on addressing issues of word sense ambiguity (Thomas and Anderson, 2012; Bott et al., 2012; De Belder et al., 2010) and lexical ranking (Kauchak, 2013; Specia et al., 2012). We show here, however, that a more prominent cause of error in the pipeline arises from the identification of complex words and the generation of substitutions. This will be an important aspect of future research into lexical simplification. As we only identified the first error at each simplification, there may be a masking effect on the later stages, preventing erroneous cases from reaching the later modules in the pipeline. However, Figure 4 shows that each module has a successively lower error rate. Furthermore, the largest error type (2B) would not be passed through to the rest of the pipeline as this category indicates those words which were incorrectly identified as requiring simplification. If these words had not been selected for simplification, they would remain uncategorised as they do not need simplification.

Finally, it was noted during the analysis that different errors had different effects on the resultant text. Some errors have a greater effect on the readability and understandability of the final text. The effects for each error are itemised below.

Type 2A: Because a complex word has not been identified for simplification, it will remain in the final text, potentially reducing user comprehension.

Type 2B: A word which did not require simplification may be altered in a way that obscures the original meaning of the text.

Type 3: As no substitution can be made, the selected word

cannot be simplified and so a complex word remains in the final text. This is the same for 3A and 3B

Type 4: The meaning of the sentence is altered. Although readability and understandability may improve, the text no longer conveys the original information. This may also result in a nonsensical translation.

Type 5: Although the system could have simplified the word, a substitution which made the text more difficult to understand was selected instead. The final text is more difficult to read.

6. Future Work

Future work should concentrate on the mitigation of errors in the lexical simplification pipeline. Although we have exposed these errors, novel techniques will be required to prevent them from occurring in the future. This work should help to focus and motivate future research into the lexical simplification pipeline. Several optimisations for the lexical simplification pipeline exist (see Section 2.) and an error analysis of these improvements is left to future work.

This form of error analysis is time intensive and is not intended to replace current evaluation methods. It is used here to expose the exact types of errors which occur in the pipeline. Future analyses may make use of manual and automatic evaluation techniques for simplification.

7. Conclusion

This paper has brought to light certain dangers associated with lexical simplification. It has shown that errors occur at each stage of the pipeline, diminishing the usefulness of a system. This research shows that any successfully deployed lexical simplification system must take into account the errors arising from each of the components of its processing pipeline.

The method of analysis proposed here could be further applied to evaluate a system's specific contributions to particular functions within the pipeline. For example, it would be possible to determine the effect of a word sense disambiguation technique by examining its effect on the error distribution. We have deliberately sought not to mitigate the errors in this experiment, but rather to expose them, thus creating a robustly analysed baseline simplification system which will form the basis for future evaluation.

8. Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council, grant number EP/I028099/1. Thanks go to the annotators, reviewers and many others who have supported this work.

9. References

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 496–501, Stroudsburg, PA, USA. Association for Computational Linguistics.

- Shoshana Blum and E. A. Levenston. 1978. Universals of lexical simplification. *Language Learning*, 28(2):399–415.
- Stefan Bott, Luz Rello, Biljana Drndarevic, and Horacio Saggion. 2012. Can Spanish be simpler? LexSiS: Lexical simplification for Spanish. In *Proceedings of COLING 2012*, pages 357–374, Mumbai, India, December. The COLING 2012 Organizing Committee.
- Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram corpus version 1.1. *Linguistic Data Consortium*.
- Marc Brysbaert and Boris New. 2009. Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, 41(4):977–990.
- John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10.
- Raman Chandrasekar and Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10(3):183 – 190.
- Jacob Cohen. 1968. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.
- Jan De Belder, Koen Deschacht, and Marie-Francine Moens. 2010. Lexical simplification. In *1st International Conference on Interdisciplinary Research on Technology, Education and Communication*.
- 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.
- Siobhan Devlin and Gary Unthank. 2006. Helping aphasic people process online information. In *Proceedings of the 8th international ACM SIGACCESS conference on computers and accessibility*, Assets '06, pages 225–226, New York, NY, USA. ACM.
- Noemie Elhadad and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 49–56. Association for Computational Linguistics.
- Noemie Elhadad. 2006. Comprehending technical texts: Predicting and defining unfamiliar terms. In *AMIA Annual Symposium proceedings*, page 239243. American Medical Informatics Association.
- Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76:378–382, November.
- James E. Hoard, Richard Wojcik, and Katherina Holzhauser. 1992. An automated grammar and style checker for writers of simplified english. In *Computers and Writing*, pages 278–296. Springer Netherlands.
- Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 9–16, Sapporo, Japan, July. Association for Computational Linguistics.
- Sujay Kumar Jauhar and Lucia Specia. 2012. UOW-SHEF: SimpLex – lexical simplicity ranking based on contextual and psycholinguistic features. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 477–481, Montréal, Canada, 7-8 June. Association for Computational Linguistics.
- David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria, August. Association for Computational Linguistics.
- H. Kučera and W. N. Francis. 1967. *Computational analysis of present-day American English*. Brown University Press, Providence, RI.
- Partha Lal and Stefan Rieger. 2002. Extract-based summarization with simplification. In *Proceedings of the Workshop on Text Summarization*, pages 90–96, July.
- Gondy Leroy, James E. Endicott, David Kauchak, Obay Mouradi, and Melissa Just. 2013. User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention. *Journal of Medical Internet Research*, 15(7):e144.
- Darren Pearce. 2001. Synonymy in collocation extraction. In *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, pages 41–46.
- Horacio Saggion, Elena Gómez Martínez, Esteban Etayo, Alberto Anula, and Lorena Bourg. 2011. Text simplification in simplext. making text more accessible. *Procesamiento del lenguaje natural*, 47:341–342.
- Horacio Saggion, Stefan Bott, and Luz Rello. 2013. Comparing resources for Spanish lexical simplification. In Adrian-Horia Dediu, Carlos Martín-Vide, Ruslan Mitkov, and Bianca Truthe, editors, *Statistical Language and Speech Processing*, volume 7978 of *Lecture Notes in Computer Science*, pages 236–247. Springer, Berlin Heidelberg.
- Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. SemEval-2012 task 1: English lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, Montréal, Canada, 7-8 June. Association for Computational Linguistics.
- S. Rebecca Thomas and Sven Anderson. 2012. WordNet-based lexical simplification of a document. In Jeremy Jancsary, editor, *Proceedings of KONVENS 2012*, pages 80–88. ÖGAI, September.
- William Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and San-

- dra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication, SIGDOC '09*, pages 29–36, New York, NY, USA. ACM.
- Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. 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, HLT '10*, pages 365–368, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Qing Zeng, Eunjung Kim, Jon Crowell, and Tony Tse. 2005. A text corpora-based estimation of the familiarity of health terminology. In José Luís Oliveira, Víctor Maojo, Fernando Martín-Sánchez, and António Sousa Pereira, editors, *Biological and Medical Data Analysis*, volume 3745 of *Lecture Notes in Computer Science*, pages 184–192. Springer Berlin Heidelberg.
- Qing Zeng-Treitler, Sergey Goryachev, Tony Tse, Alla Keselman, and Aziz Boxwala. 2008. Estimating consumer familiarity with health terminology: a context-based approach. *Journal of the American Medical Informatics Association*, 15:349–356.