LEXICAL SIMPLIFICATION: OPTIMISING THE PIPELINE

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

2015

By
Matthew Shardlow
School of Computer Science

Contents

List of Tables

List of Figures

Abstract

Introduction: This thesis was submitted by Matthew Shardlow to the University of Manchester for the degree of Doctor of Philosophy (PhD) in the year 2015. Lexical simplification is the practice of automatically increasing the readability and understandability of a text by identifying problematic vocabulary and substituting easy to understand synonyms. This work describes the research undertaken during the course of a 4-year PhD. We have focused on the pipeline of operations which string together to produce lexical simplifications. We have identified key areas for research and allowed our results to influence the direction of our research. We have suggested new methods and ideas where appropriate.

Objectives: We seek to further the field of lexical simplification as an assistive technology. Although the concept of fully-automated error-free lexical simplification is some way off, we seek to bring this dream closer to reality. Technology is ubiquitous in our information-based society. Ever-increasingly we consume news, correspondence and literature through an electronic device. E-reading gives us the opportunity to intervene when a text is too difficult. Simplification can act as an augmentative communication tool for those who find a text is above their reading level. Texts which would otherwise go unread would become accessible via simplification.

Contributions: This PhD has focused on the lexical simplification pipeline. We have identified common sources of errors as well as the detrimental effects of these errors. We have looked at techniques to mitigate the errors at each stage of the pipeline. We have created the CW Corpus, a resource for evaluating the task of identifying complex words. We have also compared machine learning strategies for identifying complex words. We propose a new preprocessing step which yields a significant increase in identification performance. We have also tackled the related fields of word sense disambiguation and substitution generation. We evaluate the current state of the field and make recommendations for best practice in lexical simplification. Finally, we focus our attention on evaluating the effect of lexical simplification on the reading ability of people with aphasia. We find that in our small-scale preliminary study, lexical simplification has a negative effect, causing reading time to increase. We evaluate this result and use it to motivate further work into lexical simplification for people with aphasia.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

There are many people without whom this thesis would not have been possible. The journey has been long and the road has not always been straight. The following people have made me laugh when I have been down, set me on the right track when I have been lost and helped me to carry on when I have felt like stopping.

My first thanks go to my incredible wife, Fran. Without her love and support, I would not be who I am today. She has stood by me through the stress and the joy that accompanies a PhD and I am thankful that she stands by me today. I am also grateful to Fran for her excellent proof reading services. She spent many hours reading and offering an extra perspective on this thesis. I must also thank my parents for encouraging me to pursue university and later academia. Their example of hard work and dedication has been a driving factor in my motivation.

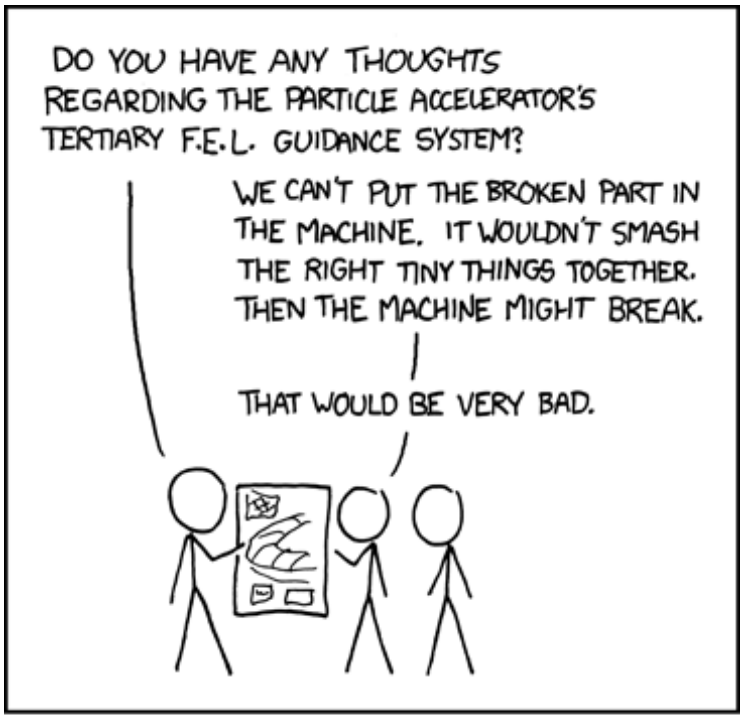
Thanks go to Jock McNaught and Simon Harper for their efforts in supervising my PhD. I have often arrived at one office or the other with lots of confusion over research directions or data interpretation. After time, discussions and much thought we have always found a way through. I am also thankful for the professional mentorship offered by both my supervisors and their efforts to develop me as a researcher.

Thank you to those who initially encouraged me to apply for a PhD. Steve Pettifer and Ernie Hill both independently suggested that I should think about postgraduate research. I would never have imagined that I was capable of a PhD and I am thankful that they encouraged me to consider the option.

My thank yous extend to the organisers of the centre for doctoral training (CDT), of which I was a part of the first cohort. Thank you to Jon Shapiro and Barry Cheetham for organising the programme and providing an extra layer of pastoral support to us all. Thanks too, to all of the other people involved with making the CDT scheme successful. It has been an excellent programme and something I am proud to have participated in. I am glad to have been a part of the CDT community, which has deeply enriched my PhD experience.

Thank you to all those who have helped out during my research. Whether that has been talking over an idea, participating in an experiment, suggesting a paper to read, proof reading my work or anything else, thank you for your help. Thank you too to all of the anonymous reviewers who have seen my work and returned constructive criticism. It has all been taken on board and used to further develop my research. In particular, thanks for the helpful feedback and encouragement to my PhD reviewers: Sophia Ananiadou and Simonetta Montemagni. Thanks also go to Matthew Lambon Ralph and Paul Conroy who helped out immensely with the user study.

Above all, I am thankful to God for all that He has done for me through Jesus Christ. My faith has been a driving force throughout my life and I am thankful for God's provision. The bible teaches that God sent his son, Jesus Christ, to die for us at the cross and that through Jesus we may be forgiven. "God made him who had no sin to be sin for us, so that in him we might become the righteousness of God" — 2 Corinthians 5:21.



I SPENT ALL NIGHT READING SIMPLE.WIKIPEDIA.ORG, AND NOW I CAN'T STOP TALKING LIKE THIS.

http://www.xkcd.com/547

Chapter 1

Introduction

Jargon. Technical terminology. Those words we use that everybody knows — or so we think. What happens when a reader finds an unfamiliar word? How can technology help struggling readers? These are the broad questions that have guided the research in this thesis. We have focused on lexical simplification (LS), the task of making difficult words easier to understand.

Information is universally disseminated as text, whether it is a website, an instruction manual, subtitles or a novel — text is everywhere. Yet, some people struggle to engage with this information. For people with a low reading age, it can be frustrating or even impossible to interact with text based information. For lay people required to read a technical or legal document, the vocabulary can be unassailable. For someone recovering from a brain injury affecting language processing functions, reading the newspaper they used to read every day may be impossible. Text simplification (TS) can help.

The need for simplified English in particular is evidenced by the popularity of the Simple English Wikipedia project, an alternative to the main English Wikipedia, which provides simplified versions of Wikipedia articles. The encyclopædia contains over $100,000$ articles which have been hand written in simple English (`https://simple.wikipedia.org`). The size of Simple Wikipedia indicates the requirement for simple English on a wide scale. However, the process of hand crafting these articles is time consuming. Automation in simplification would address this need. Simple Wikipedia is only available for the English language and there are currently no plans to develop simple Wikipedias for other languages.

In our work, we have focused on the simplification of difficult words. But why do we use difficult words in the first place? The problem could be solved if everybody wrote in a concise, easy to read style. Surely, this would be easier for the writer as well as the reader. Pinker offers the following opinion:

> *"Every human pastime — music, cooking, sports, arts, theoretical physics — develops an argot to spare its enthusiasts from having to say or type a long-winded description every time they refer to a familiar concept in each other's company. The problem is that as we become proficient at our job or hobby we come to use these as catchwords so often that they flow out of our fingers automatically, and we forget that our readers may not be members of the clubhouse in which we learned them.*
>
> *Obviously writers cannot avoid abbreviations and technical terms altogether. Shorthand terms are unobjectionable, indeed indispensable when*

> *a term has become entrenched in the community one is writing for. Biologists needn't define transcription factor or spell out mRNA every time they refer to those things, and many technical terms become so common and are so useful that they eventually cross over into everyday parlance, like cloning, gene and DNA. But the curse of knowledge ensures that most writers will overestimate how standard a term has become and how wide the community is that has learned it."* (Pinker, 2014).

We each have a different vocabulary. This is influenced by the newspapers we read, the company we keep and the classes we have taken. As Pinker so elegantly explains, specialist terms quickly become entrenched in our vocabulary. We forget whether our audience might or might not know what a word means. We do this not only with specialist vocabulary, but also with rare words. We do not realise that our vocabulary does not match that of our readers. This is the root cause of difficult language, and why we need LS.

The contributions to LS made as a part of this PhD are as follows. Each contribution reflects the author's sole work.

- An analysis of the errors produced by the processing pipeline standardly used in LS. We discovered that many errors occur at the earlier stages of the LS pipeline. Chapter 3 gives further details.
- The CW Corpus. A parallel corpus of sentences each with one lexical change, which is a simplification, is developed in Chapter 4.
- Research into the nature of Lexical Complexity (LC). We show that LC is best defined relatively in Chapter 5.
- Research into the adaptability of LC. We show that LC can be adapted for genre, but not for user. We also show that multi-word expressions can be incorporated into a LC scheme by focusing on relevant words and difficult features. These findings are shown in Chapter 5.
- In Chapter 6 we make recommendations to the field regarding how to build resources which are both comprehensive and produce meaningful substitutions.
- Finally, we present the results of a study investigating how helpful automated LS may be for people with aphasia. We show a negative result, indicating that people

Table 1.1: A list of acronyms used throughout this thesis.

Acronym	Expansion
LS	Lexical Simplification
TS	Text Simplification
LC	Lexical Complexity
MWE	Multi-word Expression
CW	Complex Word
WSD	Word Sense Disambiguation
RQ	Research Question
RH	Research Hypothesis
STD	Standard Deviation

with aphasia will take longer to read documents which have been simplified using automated LS. See Chapter 7 for details.

At this point, it is important to define some key terms that will appear in later chapters. Table 1.1 gives expansions of some common acronyms that will appear.

Text simplification is the overall process of improving the understandability of a piece of text. Simplification is hard to describe in purely heuristic terms. For example, it may seem intuitive that short sentences are simpler than long sentences. However, it may be the case that a short sentence uses complex grammatical structures or uncommon words, whereas a longer sentence could be more explanatory. The definition of simple is subjective, as different users have differing needs, and what one person considers simple may be difficult for somebody else to understand.

Lexical simplification is a type of TS. This research has focused on LS, namely, the process of identifying and mitigating complex vocabulary. Easy to understand alternatives are inserted in the place of the original words. Retention of meaning is an important factor here as the author's original intent must be preserved.

Lexical complexity is a hard notion to define. There are some cases when we can say that one word is much easier to understand (and hence more simple) than another. For example, 'accept' and 'acquiesce'. Here, the former is clearly easier to understand than the latter, yet their meanings are highly similar. There are also some cases where it can be difficult to distinguish simplicity between two words. For example, 'canine' and 'feline'. Here, it is very difficult to rank one from this pair as more understandable than the other. They are semantically unrelated, but similar in terms of length and familiarity. TS depends on several factors such as word familiarity, word length, morphology, context and ambiguity. Some combination of these factors is typically

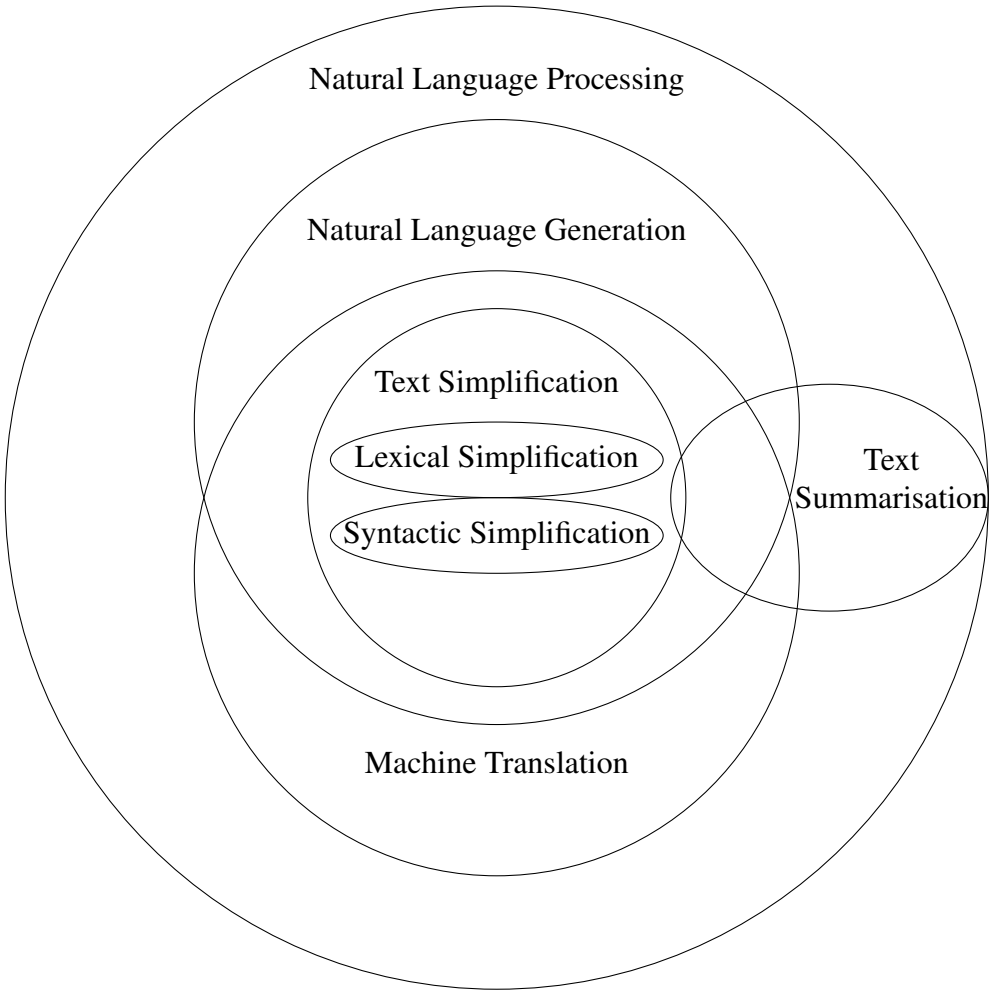


Figure 1.1: A diagram showing the place of LS in the field of natural language processing.

used to give a notion of TS.

In Figure 1.1, we show the place of LS in the wider research field. LS is a subset of TS, as is the related sub-discipline of syntactic simplification. TS can be framed as an automated machine translation problem. It can also be considered as a natural language generation problem. TS is highly related to the field of text summarisation, which draws on similar areas. Natural language generation and machine translation are both part of the field of natural language processing.

We have published the results from this thesis as they have arisen. Table 1.2 presents a list of the published work which occurs in this thesis. Often this work is replicated with very little change. We intend to submit remaining results for publication as permitted by time and resources.

Table 1.2: The papers published during the course of this PhD.

Chapter	Year	Title	Venue
2	2014	A Survey of Automated Text Simplification	International Journal of Advanced Computer Science and Applications (IJACSA)
3	2014	Out in the Open: Finding and Categorising Errors in the Lexical Simplification Pipeline	Language and Resources Evaluation Conference (LREC)
4	2013	The CW Corpus: A New Resource for Evaluating the Identification of Complex Words	Proceedings of the Second Workshop on Predicting and Improving Text Readability (PITR13)
4	2013	A Comparison of Techniques to Automatically Identify Complex Words	51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop

We have performed several different types of experiment. Appropriate experimental procedures are described alongside the research in this thesis. Some of our work took an exploratory approach, documenting the existence and form of a specific effect or phenomenon. This approach was taken in Chapter 3 where we systematically categorised errors in the LS pipeline. Other areas sought to develop and understand resources to the benefit of future research. We have seen this approach from different perspectives in Chapters 4 and 6. In Chapter 4 we sought to build a new resource which could be used to evaluate the identification of complex words (CWs), whereas in Chapter 6 we evaluated the existing resources and made recommendations based on our findings. We have also performed empirical research throughout this thesis, as seen in Chapters 5 and 7. In both of these chapters, we have explicitly stated our research questions and hypotheses. We have designed experiments to test these hypotheses and used null hypothesis significance testing to reject or accept our hypotheses. The research in this thesis is varied, but it is all joined together through the theme of LS. We have systematically investigated the LS pipeline and made improvements where possible.

Chapter 2

Literature Review

In this chapter, we will review LS from the following five perspectives: how is it used, who uses it, how is it done, how is it tested, and what resources exist? Other forms of simplification, such as syntactic and machine translation, are not the focus of this review, although they are mentioned where relevant. A more comprehensive overview of the wider literature can be found both in our survey (Shardlow, 2014) and that of Siddharthan (2014). Literature which is only relevant to a single experiment or chapter will be discussed at an appropriate point.

Written in 2008, the first survey of TS (Feng, 2008) came just before an influx of papers on the topic. It is useful as a snapshot of the early research in the field. The rapid growth in simplification research is evidenced by the number of systems evaluated by later surveys. Whereas Feng identified 8 systems, both recent surveys identified many more. These two recent surveys were published at around the same time. They cover much of the same ground, but our survey gives more of a precedence to work on LS, whereas Siddharthan's treats syntactic simplification in greater depth. The difference reflects our backgrounds and approaches to the task.

The work in this chapter is adapted from our survey paper. We have renovated the content to focus solely on LS. We have also updated the references with recently published work as well as reworded some sections to better explain the field in the context of our work.

2.1 Applications — How Is It Used?

LS is applied in many ways. It is often used in tandem with syntactic simplification, in which case the whole process is generally referred to as text simplification. The application domain of a system should be taken into consideration during its design, as different approaches suit different applications.

2.1.1 Assistive Technology

The majority of the LS literature is targeted towards assistive technologies. Text is often difficult to understand and may be poorly written. Assistive technologies enable users to interact with their environment, and LS can be used to help wade through the deluge of incomprehensible text encountered in daily life. LC acts as a barrier to understanding text and so finding easier words will help a reader to comprehend a text. Automated simplification will allow the target audience to process a text quickly and

extract more information from it.

Many different groups benefit from LS as an assistive technology showing that simplification is useful at many levels. For example, a person with learning difficulties may find it difficult to understand the English in a newspaper text, and require simplification to a very basic level. However, a person with an average reading age may understand this text perfectly, yet require a textbook entry on particle physics to be simplified into non-technical language. These are two cases where simplification is necessary, but both the input and results will be different.

An automated simplification system sitting between a user and a document would certainly be useful. However, there is a problem with this method. The statistical techniques used to apply LS to a document often result in errors occurring in the final text. These must be tightly controlled, as an error might hinder a user rather than assisting them.

Several large projects have developed TS systems for specific user groups. In the PSET project (Carroll et al., 1998), newspaper texts were simplified for stroke victims who were suffering from aphasia. Lexical and syntactic simplification were employed to remove complexities of the original text and replace them with easy to follow structures. LS was carried out (Devlin and Tait, 1998; Devlin, 1999), however no context was taken into account, which would have led to erroneous substitutions. The low quality of the substitutions may have affected the overall understandability of the text (Pearce, 2001; Lal and Rüger, 2002). The PSET project later developed into the HAPPI project (Devlin and Unthank, 2006) which had a similar focus, but was intended for use over the Internet.

The PorSimples project (Aluísio and Gasperin, 2010) developed an assistive technology for users with low literacy in Brazilian Portuguese. The need was driven by high levels of illiteracy in Brazil, where 65% of the population has difficulty with reading and comprehension. Three systems were developed:

- The first system is Simplifica, an interactive simplifying text editor (Scarton et al., 2010). A user may compute the readability index of their document and then perform automatic syntactic simplification to improve it.

- The second and third systems are FACILITA and Educational FACILITA (Watanabe et al., 2009, 2010). These are both Web-based automated TS systems, which operate within a Web browser and allow a user to select text and see it simplified.

The Spanish language Simplex project developed an LS module named LexSiS.

Simplex targets users with reading difficulties such as dyslexia and their work is tailored to suit this area. The project commenced with a corpus study of professionally simplified texts (Drndarević and Saggion, 2012), in which several LS strategies were identified. Several modules were developed including work on the simplification of numerical expressions (Bautista et al., 2013). Further work in this project looked at the development of a syntactic and LS system, especially with regard to the choice of resources (Drndarević et al., 2013; Saggion et al., 2013; Bott and Saggion, 2014).

The Flexible Interactive Reading Support Tool (FIRST) project (Barbu et al., 2013, 2015) has developed a reading aid system for people with Autism Spectrum Disorder (ASD) known as Open Book. Alongside syntactic simplification (Aranzabe et al., 2012), the project also provides lexical simplification (Barbu et al., 2015). Images and other elaborations are presented to help a user with difficult sections in the text.

LS is further applied as an assistive technology in the work of Leroy et al. (2013a), where educational medical literature is simplified with an interactive tool for lay readers. This ongoing project has sought to improve the health of medical patients by enabling medical professionals to write in an appropriate style.

As well as the larger projects mentioned above, there are also many smaller scale efforts to apply LS as an assistive technology. These efforts vary widely in their target user groups and approaches to simplification (Elhadad, 2006; Kandula et al., 2010; Yatskar et al., 2010; Woodsend and Lapata, 2011; Eom et al., 2012; Nunes et al., 2013; Angrosh et al., 2014; Vu et al., 2014; Azab et al., 2015; Collantes et al., 2015; Rennes and Jönsson, 2015).

2.1.2 Writing Aid

Another weapon in the simplification arsenal is to target the authors of documents. Every sentence is written by a person, and if that person can be enabled and encouraged to write in a simple style then further simplification is not necessary. Several researchers have built systems which help a writer to assess the readability of a text and simplify where necessary. Although some of these systems have focused solely on syntactic simplification (Adriaens, 1995; Max, 2006; Saggion et al., 2011), others have incorporated both lexical and syntactic simplifications (Scarton et al., 2010; Hervás et al., 2014), or solely focused on LS (Hoard et al., 1992; Leroy et al., 2013a).

This method empowers an author to create documents at an appropriate level for their target audience. The author can correct any mistakes made by the system and ensure that the simplified text keeps to the original meaning. The scope of this method is

limited to documents which are being prepared for specific groups who require simple language.

2.1.3 Computational Aid

Finally, simplification may be used as a preprocessing step to improve the performance of other tools in a pipeline. Syntactic simplification is typically used where long sentences create problems for parsers (Chandrasekar and Srinivas, 1997) and other tools (Beigman Klebanov et al., 2004; Siddharthan et al., 2004; Jonnalagadda and Gonzalez, 2009; Peng et al., 2012; Blake et al., 2007; Vickrey et al., 2008). LS can be applied in the same way as syntactic simplification here to reduce the complexity of an input and improve the performance of a tool. To our knowledge, only one system takes advantage of LS in this way (Chen et al., 2012). The system uses LS as a preprocessing step before a machine translation system converts the text from English to Chinese. The text is then restored to its original complexity in Chinese. The statistical machine translation system is able to create a more accurate translation from the simpler and hence more frequent language. Interestingly, no attempt is made to preserve meaning in the simplification process, as the changed words are tracked and restored later.

2.2 User Groups — Who Uses It?

Several distinct user groups have emerged as the focus of simplification projects. In this section we will look at the needs of each group and efforts to develop LS systems for them. Different groups have different needs and a simplification system developed for one group may not be useful for another. When developing a simplification system, the researcher should pay attention to his target audience and the kinds of words they are likely to find easy and difficult to understand. It is common for an LS system to be proposed which does not pay attention to the intended user group (Yatskar et al., 2010; Biran et al., 2011; Woodsend and Lapata, 2011; Thomas and Anderson, 2012; Nunes et al., 2013; Angrosh et al., 2014; Horn et al., 2014). Care must be taken when comparing and evaluating these systems as it is unclear who will benefit. Generic approaches can be beneficial if they propose novel methods for simplification which can later be adapted for specific user groups.

2.2.1 Second Language Learners

TS is a task routinely carried out by teachers of foreign languages. When teaching a topic which is of some cultural significance or relevance to current affairs, the teacher may wish to use news articles written for an audience fluent in the target language, immersing the students in the culture. Reading the same material as fluent speakers allows the student to better understand the mindset as well as learning idioms, writing patterns and grammar structures. However, the issue is that the student (especially those less advanced) will often not understand a large portion of the words in the text. LS may be used to replace difficult to understand words with easier alternatives (Blum and Levenston, 1978). The teacher controls this process to maintain a level of difficulty which is appropriate for the class.

This type of simplification is done manually. A teacher identifies the complex features of a text and replaces them by hand. There are five important strategies (Blum and Levenston, 1978): superordination, approximation, synonymy, transfer and circumlocution.

- Superordination is the technique of finding a common category to describe several items. The category may be simpler as the common category is better understood than the items it represents.

- Approximation is found when a concept such as a cultural idiom is rephrased to more generic terms.

- Synonymy is the practice of replacing rare words with more frequent synonyms. This technique is commonly found in LS. It should be noted that some words, whilst considered synonyms, may contain small semantic differences. This is generally not a problem — as the concept is approximated in simplification anyway.

- Transfer is more specific to simplification for learners of a foreign language. It involves replacing difficult words with those which are closer to the speaker's mother tongue. Highly specialised simplifications may result, which would be poor in general terms, but highly useful for their target audience.

- Circumlocution (sometimes referred to as explanation generation) is the process of explaining a difficult term for a user. Explanations may be inserted as a free flowing part of the text or may be presented alongside as a glossary.

As well as these techniques, there are many methods of syntactically simplifying a piece of text. Long sentences may be split into shorter components, complex structures may be reordered and parenthetical expressions may be removed. These are also some of the intuitive simplifications carried out when manually simplifying text.

These techniques can be highly subjective and require a working knowledge of the language which is being simplified. The techniques allows a broad range of simplifications to be carried out and produce useful simplified text. However, there are no automated means and so it is a process with a high workload and a low throughput (Petersen and Ostendorf, 2007).

There is some debate amongst academics as to the effectiveness of simplified texts as a learning aid (Crossley et al., 2012; Young, 1999). The main advantage for the foreign language learner is that they are presented with a piece of text which is tailored to their understanding and capabilities. They will be able to better interact with the text and will gain confidence in comprehension skills. However, there are two potential disadvantages. Firstly, if the text is tailored to their ability, then they risk neither encountering nor learning anything novel. If a language learner never encounters new vocabulary, then they cannot be expected to improve. Secondly, the simplified text may not be representative of typical language in the source text. For example, when tailoring an English text for second language learners whose mother tongue is French, it is likely that words similar to those found in French will be retained. Other words will be modified to be more like the language patterns of French. The language learner is presented with a poor example of English, which may even be detrimental to learning the language.

Whilst there are many efforts to define and classify the manual process of LS (Blum and Levenston, 1978; Crossley et al., 2012; Young, 1999), less research exists to automate simplification for language learners. Petersen and Ostendorf (2007) create a corpus of complex and simple articles for use in LS for second language learners. LS for second language learners is just one example of the great need for simplified text. Recent systems (Eom et al., 2012; Azab et al., 2015) build reading aids for language learners. They incorporate word sense disambiguation (WSD) and display definitions upon request. Much work goes into simplifying foreign language texts for learners and automated processes will greatly help those involved.

2.2.2 Cognitive Impairment

Cognitive Impairment affects a reader's ability to interact with the world around them. Text which is easy to understand for an unimpaired reader can be troublesome for those suffering from a language processing disability. Some disabilities are congenital, such as dyslexia. Others present as a result of a brain injury or stroke, such as aphasia. In both cases, the user will benefit from texts which are easier to understand.

The Simplex project (Bott et al., 2012; Bott and Saggion, 2014; Saggion et al., 2015) targeted simplification towards Spanish users with cognitive impairments of all types. They found that specific types of simplifications were often employed for users with cognitive impairments (Drndarević and Saggion, 2012). These were further developed into a TS service (Bautista et al., 2013; Drndarević et al., 2013; Saggion et al., 2013; Bott and Saggion, 2014). In the PSET project (Carroll et al., 1998), LS was developed for users with aphasia (Devlin and Tait, 1998). People with aphasia struggle to remember uncommon words and so the PSET LS strategy replaced infrequent words with more frequent alternatives.

In her thesis, Rello (2014) proposes the DysWebxia programme, an assistive technology for improving reading comprehension in people with dyslexia. Many strategies are proposed including proper selection of font and other visual factors. Rello also proposes several methods of TS which are beneficial to people with dyslexia. These are incorporated into her system.

2.2.3 Lay Readers Of Medical Text

From an outsider's perspective, the medical world has a particularly acute case of jargon overload. Latin and Greek terms are routinely used where a more common term would suffice. Patients are typically not medical experts and must be assisted to interact with the difficult text they may be presented with. Several researchers have looked at building reading aids for medical text which incorporate simplification. Strategies include finding paraphrases for technical terms (Elhadad and Sutaria, 2007; Deléger and Zweigenbaum, 2009; Grabar et al., 2014), generating appropriate explanations (Elhadad, 2006; Kandula et al., 2010; Alfano et al., 2015) and looking up easier synonyms in a medical dictionary (Abrahamsson et al., 2014; Grigonyte et al., 2014). Health informatics is a well funded and well resourced area which makes it a rich area for the implementation of simplification systems.

2.2.4 Adults With Low Literacy

People with low literacy may struggle to interact with information which is presented to them in their daily lives (Aluísio and Gasperin, 2010). Whilst much information is intended to be freely accessible, documents such as works of fiction or news articles may be written in an inaccessible style. When this is the case, people of low literacy are limited in their choice of literature — discouraging them from reading at all. The purpose of LS is two-fold. Firstly, it enables users to interact with new sources of information, giving the user choice over what they wish to discover and read. Secondly it is educational, encouraging a user to read more and thus develop their literacy skills.

2.2.5 Children

As a child develops, their level of literacy grows. They cannot be expected to read texts at a level higher than their current reading grade, although sometimes these may be relevant to their study. Also, some children may develop more slowly than others and require texts that are relevant to their learning to be simplified. Dingli and Cachia (2014) develop an e-reader capable of automatically providing simplifications as required by a struggling child. Other efforts seek to simplify stories (Vu et al., 2014) or everyday information from the Web or newspapers (De Belder et al., 2010; De Belder and Moens, 2010; Kajiwara et al., 2013).

2.3 Approaches — How Is It Done?

This section will explain the different methods taken to create an LS system. Although each system uses a slightly different methodology, we have identified three broad categories as outlined below. We will also argue that every system can be expressed as a version of the LS pipeline and hence that our work on this framework is beneficial to work on other approaches.

2.3.1 Pipeline

As we have previously mentioned, LS is the task of identifying and replacing CWs with simpler substitutes. No attempt to simplify the grammar of a text is made, but instead we focus on simplifying complex aspects of the vocabulary. There are typically 4 steps to LS as shown in Figure 2.1. Firstly, the complex terms in a document must be

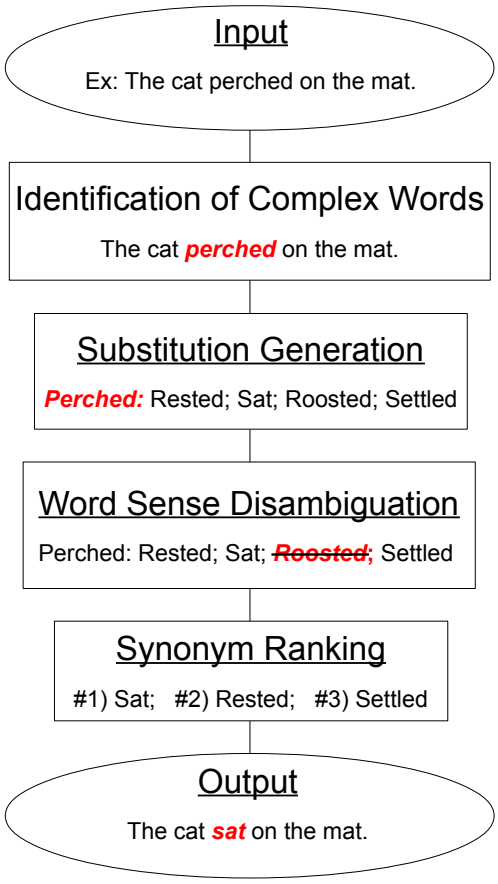


Figure 2.1: The LS pipeline. Many simplifications will be made in a document concurrently. In the worked example the word 'perched' is transformed to 'sat'. 'Roosted' is eliminated during the WSD step as it does not apply properly in the context of 'cat'.

identified. Secondly, a list of substitutions must be generated for each complex term. Thirdly, the substitutions should be refined to retain those which make sense in the original sentence. Finally, the remaining substitutions must be ranked in order of their simplicity. The most simple synonym is used as a replacement for the original word. Systems have made differing variations on this theme with many earlier approaches performing no disambiguation.

In the first notable work in automated LS (Devlin and Tait, 1998), the authors rank synonyms from the semantic thesaurus WordNet (Fellbaum, 1998) using Kučera-Francis frequency (Kučera and Francis, 1967) to identify the most common synonym. This work has influenced LS systems since (Aluísio and Gasperin, 2010; De Belder et al., 2010; Bott et al., 2012; Leroy et al., 2013a; Nunes et al., 2013; Grigonyte et al., 2014; Paetzold, 2015), providing a framework with many avenues for exploration.

One of the major stumbling blocks with primitive lexical substitution systems is a loss of meaning due to word sense ambiguity. A word can have multiple meanings and it is difficult to distinguish which is intended. Different meanings will have different relevant substitutions and so replacing a word with a candidate substitution from the wrong word sense can have disastrous results for the cohesion of the resultant sentence. Early systems (Devlin and Tait, 1998) did not take ambiguity into account at the expense of their accuracy. The natural language processing technique of WSD is useful here. WSD is a set of techniques which use the surrounding context in a sentence to determine the most likely word sense. Once the correct word sense is determined, the potential synonyms should be limited to only those which will maintain coherence within the sentence.

WSD has been applied to LS in a number of different ways. These usually involve taking a standard lexical substitution system and applying a WSD technique at some point. One such system is the latent words language model (LWLM) (Deschacht et al., 2012), which is applied to LS during the substitution generation step. The LWLM is used to generate a set of words which are semantically related to the original word. These are then compared against the substitutions returned by WordNet to remove any antonyms found by the LWLM. WordNet is useful for WSD as it gathers words according to their semantic similarities into a group called a "synset". WordNet is also used in Thomas and Anderson (2012) where a tree of simplification relationships is developed based on WordNet hypernym relations. This tree is used to reduce the size of the vocabulary in a document. WSD is carried out to place content words into their correct WordNet synset. Simplification may then be carried out by looking at the relevant node in the tree. WSD is also carried out by the use of context vectors (Biran et al., 2011; Bott et al., 2012). In this method, a large amount of information is collected on the surrounding context of each word and is used to build a vector of the likely co-occurring words. Vector similarity measures are then used to decide which word is the most likely candidate for substitution in any given context. These methods show the diversity of WSD as applied to LS. Baeza-Yates et al. (2015) use n-grams as a model for the context around a word. They disambiguate by evaluating the 5-gram frequency where the third word reflects the target word and is substituted for the potential substitutions. The 5-gram with the highest frequency is selected.

Other work has attempted to improve LS by improving the frequency metrics which are used. Frequent words have been shown to increase a text's readability (Rello et al., 2013b). Simple Wikipedia has been shown to be more useful than English Wikipedia

as a method for frequency counting (Kauchak, 2013). N-Grams have shown some use in providing more context to the frequency counts, with higher order n-grams giving improved counts (Ligozat et al., 2013). However, the most effective method has so far proven to be the usage of a very large initial data-set (Ligozat et al., 2013; Kauchak, 2013), namely the Google Web 1T Corpus (Brants and Franz, 2006). Frequency modelling is particularly difficult in compounding languages (Abrahamsson et al., 2014), where long CWs can be created from well known lexemes. This research is interesting for English and other languages where a similar phenomenon occurs with multiword expressions (MWEs). Advances in frequency modelling for compounding languages could help to determine the complexity of MWEs, and vice versa.

2.3.2 Paraphrasing

Another approach to LS is to find simplifying paraphrases. A paraphrase is a pair of semantically equivalent phrases, where each phrase may contain one or more words. In this method, the paraphrases are identified manually (Hoard et al., 1992) or learnt from parallel corpora (Elhadad and Sutaria, 2007; Deléger and Zweigenbaum, 2009; Yatskar et al., 2010; Grabar et al., 2014) and stored in a dictionary ready for deployment. When the more difficult phrase from the pair is detected in a source text, then the simpler phrase is substituted into the sentence.

It is essential to ensure the correctness of the pair of phrases in the dictionary. If the phrases are not semantically equivalent then errors may be introduced into later text. If the set of paraphrases is correct, then it may offer a lower error rate than the alternative of looking up synonyms in a dictionary. The system can only make simplifications which are found in its dictionary. This limitation may result in fewer simplifications overall and less adaptability to unseen vocabulary.

Paraphrases were used by Hoard et al. (1992) to produce a tool for the writers of Boeing's aircraft manuals which helped them keep in accordance with the ASD-STE100 standard for simplified English (see `http://www.asd-ste100.org/`). The paraphrases are found in the technical specification of the language. When a writer uses a word which is not allowed, the system identifies the error and suggests a permitted term.

Recently, this method has been adapted to take the context of the paraphrases into account (Angrosh et al., 2014; Siddharthan and Mandya, 2014). The paraphrases are expressed as rules which take syntactic factors into account. The rules contain a set of appropriate contexts for application, which are learnt from the data. If this set of

contexts is particularly extensive then the manager of the rule set might decide to allow the rule to be applied in all situations, similar to Woodsend and Lapata (2011).

It may seem at first that paraphrasing does not bear much resemblance to the pipeline method from Section 2.3.1. However, the key elements of the pipeline must still be addressed. Difficult words are identified, in this case those for which a paraphrase can be found. Next, substitutions must be generated by looking up the correct paraphrase in the dictionary. Once a substitution has been found, the system must decide if it is appropriate for the context, giving rise to disambiguation. Finally, the simplest paraphrase must be selected. Again, this information is encoded in the dictionary and the notion of a 'simplifying paraphrase' must be defined before paraphrases can be created or learnt. These steps are typically defined within the structure of the paraphrase rules themselves, rather than by external resources. Paraphrasing may be more lightweight than the pipeline approach. Conversely, paraphrasing may be less flexible, as specific rules must be learnt for each word rather than generic rules for all words.

2.3.3 Lexical Elaboration

A third approach to simplification is to keep difficult vocabulary in place and provide the reader with short explanations of each complex term. These explanations can be provided as part of the text (Kandula et al., 2010; Drndarević and Saggion, 2012), in pop-up boxes (Watanabe et al., 2010; Alfano et al., 2015; Rello et al., 2015) or separately from the text (Elhadad, 2006). The explanations are found in dictionaries and are often targeted at domain specific language which one would not expect a reader to understand. For example, in a medical text one would try to identify technical medical terminology and provide the reader with concise explanations of each term. The original text is preserved and presented to the reader. The author may have chosen these words to convey specific technical qualities, which would have been lost in replacement. Because the original terms are explained, the reader will learn these terms over time. In some systems, the definition is presented to a user on demand (Elhadad, 2006; Watanabe et al., 2010; Alfano et al., 2015) allowing them to control the level of simplification.

In some cases, it may be distracting for a user to incorporate extra text into a sentence. Several systems which use lexical elaboration also incorporate syntactic simplification to reduce sentence length (Kandula et al., 2010; Watanabe et al., 2010). In one study for people with dyslexia (Rello et al., 2013a), it was found that providing

explanations of terms was more beneficial to the users than simplifying the text.

Kandula et al. (2010) perform lexical elaboration for health information. Short explanations of key concepts are generated. These explanations help a user to identify the key words by associating them with more common words which the user is likely to understand. The explanations take the form of short sentences which describe pre-identified relationships such as "humerus *a part of arm*" or "Polynephritis *a type of infection*". In another form of elaboration, Alfano et al. (2015) process medical texts for readers and place infobuttons next to potentially troublesome terms. The lay reader can click on these buttons if they do not understand a particular term.

Once again, there is a clear set of stages that must be undertaken in this form of LS which closely mirror those in the pipeline. Instead of generating substitutions, we generate explanations. In the case of ambiguous terms, the correct explanation must be selected. Finally, some care should be taken to make sure that the explanation will actually make the term easier to understand as an explanation using technical terms may further confuse the reader. To decide whether an explanation will be easier to understand than the original term we also need a notion of sentence complexity.

2.4 Evaluation — How Is It Tested?

The concept of simplicity is naturally subjective. What we find to be simple will be affected by many factors such as: prior knowledge (understanding of concepts or idioms), previous assumptions (such as a prior explanation) and level of literacy (words or grammatical structures may be misunderstood). This subjectivity means that it is very important to choose a good evaluation measure which will give an accurate representation of a system's performance.

It may seem counter intuitive then that a wide variety of different evaluation methods is used. In fact, most new systems apply their own evaluation method or at least apply a similar evaluation method but in a different way. This variety is a problem for two reasons. Firstly, it becomes very difficult to compare work. Some papers provide evaluations of other work alongside their own, although comparison is often done with historic work which is used as a simple baseline against which improvement may be shown. Comparing contemporary techniques is almost impossible when they use differing evaluation methods. Secondly, there is very little evaluation of the evaluation methods themselves. One measure may give a truer picture of the simplification of a piece of text or some measures may not accurately reflect the simplification. These

issues remain unaddressed.

The easiest solution would be to create one evaluation system which could be used for all TS, similar to the BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) measures in machine translation. However, a unified system may not be a possibility. As previously noted, there is much varied work in TS. The reason for so many different evaluation methods is because these implementations are fundamentally conceptually different. For example, some systems only perform LS, whereas some systems only perform syntactic simplification, and yet others still perform both. Evaluation methods which work for one type may not be useful for all. Trying to find a single evaluation method to work with all these possibilities is a challenging task.

Evaluation methods fall into two main categories: automatic and manual. Automatic measures are generally very quick but do not necessarily reflect a user's perspective. They typically look at heuristic features of a text such as sentence length. Automatic techniques are quick and cheap and can be used with little cost to the researcher. Manual techniques are generally more reliable, as they involve asking several people their opinion on the produced simplification. Whilst this judgment is still subjective, simplification is a subjective process and requires some human input to decide what is truly simple. Subjectivity may lead to variation in results as different people will have individual simplification needs. Some evaluation techniques are listed below. Direct user evaluation and independent judges are both manual techniques. The rest are automatic.

2.4.1 Direct user evaluation

An LS system is usually developed for a specific group. These systems can be evaluated by assessing the reading progress of that group with and without simplification. Original and simplified texts are presented to a user (Carroll et al., 1998; Kandula et al., 2010; Eom et al., 2012; Leroy et al., 2013a; Dingli and Cachia, 2014). It is essential to select the right people for the test. For example, if the simplification is done with the intention of aiding dyslexic users, then it should be tested primarily on this group. Direct user evaluation is not easy to perform and many factors must be controlled. For example, if a simplified text is presented alongside its original text then the the user will learn concepts from the first text which will affect their performance on the second. Presenting unrelated simple-complex pairs may also be problematic as one document may be conceptually simpler than the other. This issue can be resolved by presenting users with several complex and simplified documents. Kandula et al. (2010) perform

Cloze tests (Taylor, 1953) using 4 reviewers and giving them 4 simple and 4 complex documents each.

2.4.2 Independent Judges

The use of independent judges is the most common method of manual evaluation (Hoard et al., 1992; Siddharthan, 2006; Elhadad and Sutaria, 2007; De Belder and Moens, 2010; Bott et al., 2012; Bautista et al., 2013; Nunes et al., 2013; Angrosh et al., 2014; Grabar et al., 2014; Grigonyte et al., 2014; Horn et al., 2014). It is sometimes used alongside automated readability measures (Keskisärkkä, 2012; Drndarević et al., 2013; Abrahamsson et al., 2014) and machine translation measures (Woodsend and Lapata, 2011; Vu et al., 2014). Several independent reviewers are asked to judge the quality of a system's output on several accounts. Typically these include measures of grammaticality, meaning preservation, simplification and cohesiveness. These are often measured on some kind of heuristic scale. For example, Siddharthan (2006) measures meaning preservation on a scale of 0–3, where 0 indicates difference and 3 indicates preservation. 1 and 2 represent intermediary levels, however the evaluation will be subjective as judges may vary. Biran et al. (2011) note a moderate inter-annotator agreement for a similar evaluation method. This method does not involve the users and so it is difficult to say whether a high score in any of these areas would translate into real benefits for the intended audience. Typically, several reviewers of similar linguistic training and fluency are chosen. Results may also be crowdsourced via the Internet (Lasecki et al., 2015), although care should be taken to evaluate and ensure the quality of the annotations (Sabou et al., 2014).

2.4.3 Automated Readability Measures

Several methods exist to computationally determine a sentence's readability. These methods use the surface features of a text such as the number of polysyllabic words, words per sentence, syllables per word or total number of words. Two such methods are the Flesch-Kincaid Grade Level (Kincaid et al., 1975) and SMOG (McLaughlin, 1969) (Simple Measure Of Gobbledygook). These both calculate the US grade level that a text is aimed at. A problem with these and other automated measures is that they may not accurately represent the simplification needs of a user (that is, the specific type of simplification which will help that user to better understand the text in question), and that they may be easily fooled. For example, Flesch-Kincaid takes sentence length

into account. A text with very short sentences will have a higher readability score than a text with long sentences. However the short text may use a much more complex vocabulary making it more difficult to read and understand.

Automated readability is always used in conjunction with another manual method of evaluation (Kandula et al., 2010; Woodsend and Lapata, 2011; Keskisärkkä, 2012; Drndarević et al., 2013; Abrahamsson et al., 2014; Vu et al., 2014). It generally correlates well with these methods, however it cannot be relied on solely. The work of Štajner and Saggion (2013a) has attempted to adapt automated readability measures for TS with some success.

2.4.4 Machine Translation Measures

LS can be thought of as a machine translation task. Difficult English is translated into simple English. The field of machine translation has standard measures of evaluating translation tasks which have been used in the evaluation of simplification (Woodsend and Lapata, 2011; Vu et al., 2014). Two such measures are BLEU (Papineni et al., 2002) and NIST (Doddington, 2002). BLEU compares one candidate translation with several reference translations. N-grams are then compared between the reference and candidate translations to give a measure of precision. The higher the score, the more accurate the translation. NIST is an adaptation of BLEU which also uses n-gram co-occurrence. It differs in that it weights the n-grams according to how informative they are, based on frequency of occurrence in the text. Whilst these measures are similar to the performance of a human judge, it should be noted that they were developed as:

> "An automated understudy to skilled human judges which substitutes for them when there is need for quick or frequent evaluations" (Papineni et al., 2002).

They are not intended as a final measure but to be used in conjunction with manual evaluation, when an alternative is not possible. These methods are useful for LS evaluation given a large corpus of reference translations.

2.4.5 Corpus Based Evaluation

There are several corpora that can be used to evaluate an LS system. Paraphrasing systems might compare their paraphrases to a standard reference list (Deléger and Zweigenbaum, 2009; Yatskar et al., 2010; Kajiwara et al., 2013). Care should be taken

as a reference list offers little guarantee of completeness and may flag correct entries as false if they are not included on the list. Systems which test specific modules and aspects of LS may use corpora designed to isolate and highlight these phenomena (Saggion et al., 2013; Ligozat et al., 2013; Thomas and Anderson, 2012; Inui et al., 2003). This evaluation may be useful to show a specific aspect of the system, however it can be difficult to know how well the system will compare to others in the literature.

A recent important development in the field of LS is the lexical substitution task from SemEval 2012 (Specia et al., 2012). Participants designed a system to rank words in terms of their simplicity. The words were given as valid replacements for a single annotated word in a sentence. Many such sentences were provided and systems were able to train and test on sample data before being deployed for the final testing data. The corpus was developed by crowdsourcing through Amazon's Mechanical Turk (De Belder and Moens, 2012) (see `www.mturk.com`). Annotators were asked to rank the substitutions in order of their simplicity. These rankings were then combined to form one final ranking.

The lexical substitution task isolates the synonym ranking problem within LS where the aim is to find the easiest synonym. Systems do not have to focus on other distractions, such as identifying CWs or synonym generation, but can focus solely on ranking. Several systems were developed to produce these rankings and the techniques used considered a variety of methods such as: language models for word context (Ligozat et al., 2012; Sinha, 2012; Jauhar and Specia, 2012), decompositional semantics (Amoia and Romanelli, 2012) and machine learning techniques (Jauhar and Specia, 2012; Johannsen et al., 2012). A comprehensive overview and comparison of these is given in the task description (Specia et al., 2012). The SemEval task has served LS in two separate ways: firstly, it has promoted the field and specifically the area of LS. Secondly, it has provided an evaluation of different methods for synonym ranking.

The task provided participants with both a gold standard corpus and a standardised evaluation measure. The corpus is highly specialised for the task of ranking synonyms in order of their simplicity. It contains a set of sentences each with a single annotated word. Several possible substitutions are given for the word and the gold standard data provides an optimum ranking of these in terms of their simplicity. A candidate system can be evaluated automatically by comparing the system's ranking with the gold standard. Many trials can be run in a short space of time, which lends well to machine learning approaches. A drawback is that it is highly specialised to one specific part of the LS problem and is less useful for other tasks within the wider

Table 2.1: A list of some key tools involved in LS.

Tool	Input	Output
Tokeniser	A sequence of characters	A sequence of tokens as defined for the application (e.g. words, numbers, punctuation and other symbols).
Lemmatiser	A wordform	The normalised wordform without any prefixes or suffixes.
Part-Of-Speech Tagger	A sequence of words	A Part-Of-Speech Tag (Noun / Verb / etc.) associated with each word.
Named Entity Recogniser	A sequence of words	Tags indicating which words may form named entities in the text.
Word Sense Disambiguation	A word and its context	The sense of the word.
LC Measure	A word	The difficulty of the word.
Thesaurus	Either a word or word sense	Synonyms which can replace the given word.

sphere of simplification. It should also be noted that the evaluation is still an automatic measure and so is best used alongside direct user evaluation.

2.5 What Resources Exist?

In this final section we will take a look at what tools and corpora are necessary for LS. We will see what exists currently and what needs to be developed to further the field.

2.5.1 Tools

Every natural language processing subfield uses a series of tools to analyse a text. LS is no different and relies on several key processes. For reproducibility, the names of the exact resources should always be included in the literature along with version numbers and specific parameters which may affect performance. Without these, it can be very difficult to faithfully reproduce results. Several important tools are shown in Table 2.1.

Tokenizer: Before we can simplify any words, we must work out what those words are. A tokenizer does more than just group letters between spaces. It must also handle punctuation, contractions and possibly hyphenation.

Lemmatiser: A lexical resource is usually categorised by the lemma of each word. The lemma is the root form of the word. So, 'walks' would become 'walk', 'drawers' would become 'drawer' and 'went' would become 'go'. By conflating information for the different forms of each word, we can get more information about the root form of the word. More information about each word will reduce the amount of error in the resource.

Part-Of-Speech Tagger: Many other resources such as the lemmatiser, named entity recognition and thesaurus require a part-of-speech tag associated with each word. Modern part-of-speech taggers use probabilistic models to assign the correct sequence of tags to a sentence.

Named Entity Recognition: Named entities should not be simplified in a text, as they are used by the author to indicate a concept which is key to the discourse. Simplification of these concepts is likely to cause a change in the meaning of the text. Named entity recognition differs depending upon domain. It is important to use a resource which is suitable to the text in question to ensure the highest possible accuracy.

Word Sense Disambiguation: Some words have several meanings. We must attempt to select the correct meaning before attempting to simplify the word. This is a difficult problem with many people working to solve it. First, we need a lexical database such as WordNet or BabelNet (Navigli and Ponzetto, 2012b). Naïve methods of disambiguation, such as always choosing the most common sense, can be hard to beat (Navigli, 2009). However, methods which employ graph similarity have performed well in the past (Pedersen and Kolhatkar, 2009; Navigli and Ponzetto, 2012a).

LC Measure: In order to identify difficult words and to rank substitutions according to their complexity, it is important to be able to measure the difficulty of a word. Length and frequency are often used and combined to give a measure of LC. Other factors can also be incorporated.

Thesaurus: We need a method of identifying candidate synonyms for a given word. The thesaurus must be incorporated with WSD to get synonyms that will retain the original meaning in the original context.

2.5.2 Corpora

Most natural language processing algorithms require a corpus. A corpus is a body of text, usually in the form of sentences and paragraphs collected from some source which is typical of the language that will be encountered by the system. The corpus is used when training and testing algorithms. It may be created and annotated by automatic or manual methods to provide extra information to an algorithm.

Whilst many different corpora have been used for TS, there is no single gold standard. Work using different corpora can be difficult to compare. Many recent systems have begun to use Simple Wikipedia as a corpus (Napoles and Dredze, 2010) and so some standardisation is emerging, however different corpora are separately developed and techniques remain difficult to compare.

One technique for corpus creation is to use annotators to suggest simplifications for complex sentences (Chandrasekar and Srinivas, 1997; Inui et al., 2003). Several experts are presented with sample texts and asked to simplify these according to some common guidelines. Annotators who know the specific needs of the target audience may be used to improve the simplification. The use of annotators is expensive as it requires skilled people to be employed. It will also be difficult to create a large corpus, due to the annotators' speed of work. Inter-annotator agreement may be low due to the subjective nature of simplification (De Belder and Moens, 2012). One sentence may be simplified in several different ways and so several annotators may disagree on the best simplification, even though all successfully simplify the text.

Given the high expense and low output of manually simplifying complex text, many researchers have automatically gathered corpora of pre-existing simplified texts. These consist of original 'complex' texts which are paired up to their simplified versions. News articles are sometimes accompanied with simplified versions (Petersen and Ostendorf, 2007; De Belder et al., 2010; Bott and Saggion, 2011). If no simplified versions exist, then sentences may be manually simplified. News texts are generally written in a modern style and so provide a good representation of modern language. There are many news articles and more are produced each day. These articles are highly distinct as they will usually concern a different topic. Even those concerning the same topic will still have significant differences (otherwise they would not be published as news). This distinction means that a large amount of text can be harvested over time and a large corpus may be built.

'The Encyclopædia Britannica' and its simplified counterpart 'The Britannica Elementary' are another example of a set of articles with simplified counterparts (Barzilay and Elhadad, 2003; Nelken and Shieber, 2006). Training data is created by manually aligning some sentences from the corpus. Alignment is done by clustering topics of paragraphs to find semantically related paragraphs and then performing a more strict matching between these. A simplified corpus is produced, which may be useful for LS. The encyclopædia covers large amounts of information in great depth and so may give a broad overview of complex-simplified texts.

Recently, Simple Wikipedia has become widely used as a reference text of simplified English (Napoles and Dredze, 2010). Simple Wikipedia articles usually have a parallel article on the English Wikipedia site and are often created by directly simplifying English Wikipedia articles. The edit histories of these articles may be obtained and processed to discover common simplification operations which are used. Parallel aligned corpora of sentences may also be obtained by processing aligned complex and simple articles to find semantically related sentences. This approach is taken to produce the PWKP dataset (Zhu et al., 2010)which contains around 108,000 sentence pairs. Simple Wikipedia contains around 83,000 articles compared to English Wikipedia's 3,960,000. As it grows, more parallel aligned text will be available, making it possible to create larger corpora. Xu et al. (2015) criticise the use of Simple Wikipedia in simplification research. They argue that the simplifications are inadequate for TS purposes and that the text style is not suitable for most user groups. They propose a new dataset which is composed of manually simplified news articles.

Corpus construction for LS is an important area of research. Whilst there has been much work into aligning Simple Wikipedia with English Wikipedia, there are also other corpora which do not use Wikipedia (Klerke and Søgaard, 2012). The size of Simple Wikipedia has only become large enough to form parallel corpora in the last few years (Yatskar et al., 2010; Zhu et al., 2010; Hwang et al., 2015). Furthermore, Simple Wikipedia only exists in the English language, and so the creation of monolingual parallel corpora in other languages must use other techniques (Brunato et al., 2015). Creating parallel aligned corpora is also an active field of research for text summarisation and paraphrase generation, which are both very similar tasks to TS. Whilst techniques may be borrowed from these fields, the corpora produced may not be viable for LS purposes.

2.6 Research Influences

The research in this thesis is heavily influenced by the work from others that has come before it. We have extensively reviewed the literature as shown above and we have made decisions for our own research directions based on what we have found in the literature. Research is a community endeavour and our work must be taken in the context of other researchers' work, as reviewed in this chapter.

The LS pipeline, as described earlier in Section 2.3.1, has been hugely influential to our research. We have used this model to build and evaluate LS systems. We have formalised and developed the pipeline and it has influenced all of the research presented in this thesis. In Chapter 3, we implement the pipeline and study the types of errors it produces. In Chapters 4–6, we look at ways of improving the modules of the pipeline to reduce the total number of errors. In Chapter 7, we use the pipeline as part of a human assisted simplification system to produce simplified text for people with aphasia.

We also found that there was no consistent evaluation of LS in the literature. This is still true at the time of writing. We have attempted to provide robust measures of evaluation at each stage of our research, in order to address the lack of evaluation in the field. The creation of the CW corpus in Chapter 4 meets the specific need of a corpus for evaluating the identification of CWs.

Finally, we observed that there were many different user groups for whom simplification was applied. We chose to adopt a general strategy with our research, focusing on the pipeline and researching methods that would be helpful for all groups. In Chapter 7, we adapt our research to meet the needs of people with aphasia. We felt that working with people with aphasia was appropriate as very little research has been done towards simplification for people with aphasia since the work of Carroll et al. (1998).

Chapter 3

Errors in the Pipeline

Early on in our research, we attempted to reimplement the LS pipeline proposed by Devlin and Tait (1998). In accordance with others (Pearce, 2001; Lal and Rüger, 2002), we found that the system frequently led to errors. Moreover it was apparent that although some genuine simplifications were occurring, the majority of operations resulted in nonsense. We made modifications to the pipeline and looked at mitigation strategies, but we found that this high error rate prevailed.

To progress in this work, we need to understand the types of errors that can occur. No previous studies categorise the errors in the LS pipeline, although several people have commented on them. Lal and Rüger (2002) report that a lack of WSD leads to errors in their system. Pearce (2001) uses bigram frequencies to generate substitutions which will fit in the final sentence. The 2012 SemEval task on LS (Specia et al., 2012) encouraged competitors to improve the synonym ranking task. In this chapter, we will present the results of an experiment to classify the errors in the LS pipeline. We will see that rudimentary LS systems suffer from a high error rate. The resultant text sounds strange and may lose its original meaning. The rest of the chapter is structured as follows:

- A basic simplification system modelled on the seminal work in the field (Devlin and Tait, 1998) is presented in Section 3.1.1.
- The categorisation of errors is shown in Section 3.1.2. A classification scheme is also proposed.
- The results of the classification for a moderate sized corpus are presented in Section 3.2. An inter-annotator study is also presented.
- An extended discussion of the results can be found in Section 3.3.

3.1 Experimental Design

To investigate the types of errors present in the LS pipeline, we built a simplification system similar to that described by Devlin and Tait (1998). As no standard system for LS exists, we chose not to apply any optimisations which have been proposed in later work. 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. We inspected the vocabulary in our corpus and found that it was 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 3.1, was used to reduce subjectivity during the annotation. The simplifications were recorded and cross checked to ensure that categories were consistent. To further ensure consistency and reproducibility, all the raw data was made available via the LRE map. The LRE map is a large open library of language resources maintained by the ELRA.

3.1.1 Simplification System

The simplification system used in this research follows the pipeline shown earlier in Figure 2.1. Below, we detail the design decisions taken for each step in the pipeline.

CW Identification A threshold determined whether a word would require simplification. Every word with a Kučera-Francis frequency less than or equal to four was considered for simplification. We chose to omit capitalised words as they generally denote named entities.

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

Word Sense Disambiguation No WSD was applied, reflecting previous work.

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

3.1.2 Annotation Workflow

The annotation workflow is shown in Figure 3.1. We report the first error that occurs in each instance. If the errors at earlier stages were resolved, then an error might still occur at a later stage. We will investigate the extent of this effect in the following section.

The error types parallel the stages of the pipeline. The first category is reserved for the state of no error. The errors associated with CW identification and substitution generation (types 2 and 3 respectively) were split into two further categories. These subcategories were labelled alphabetically (2A, 2B, 3A and 3B) to avoid confusion. The categories are described below:

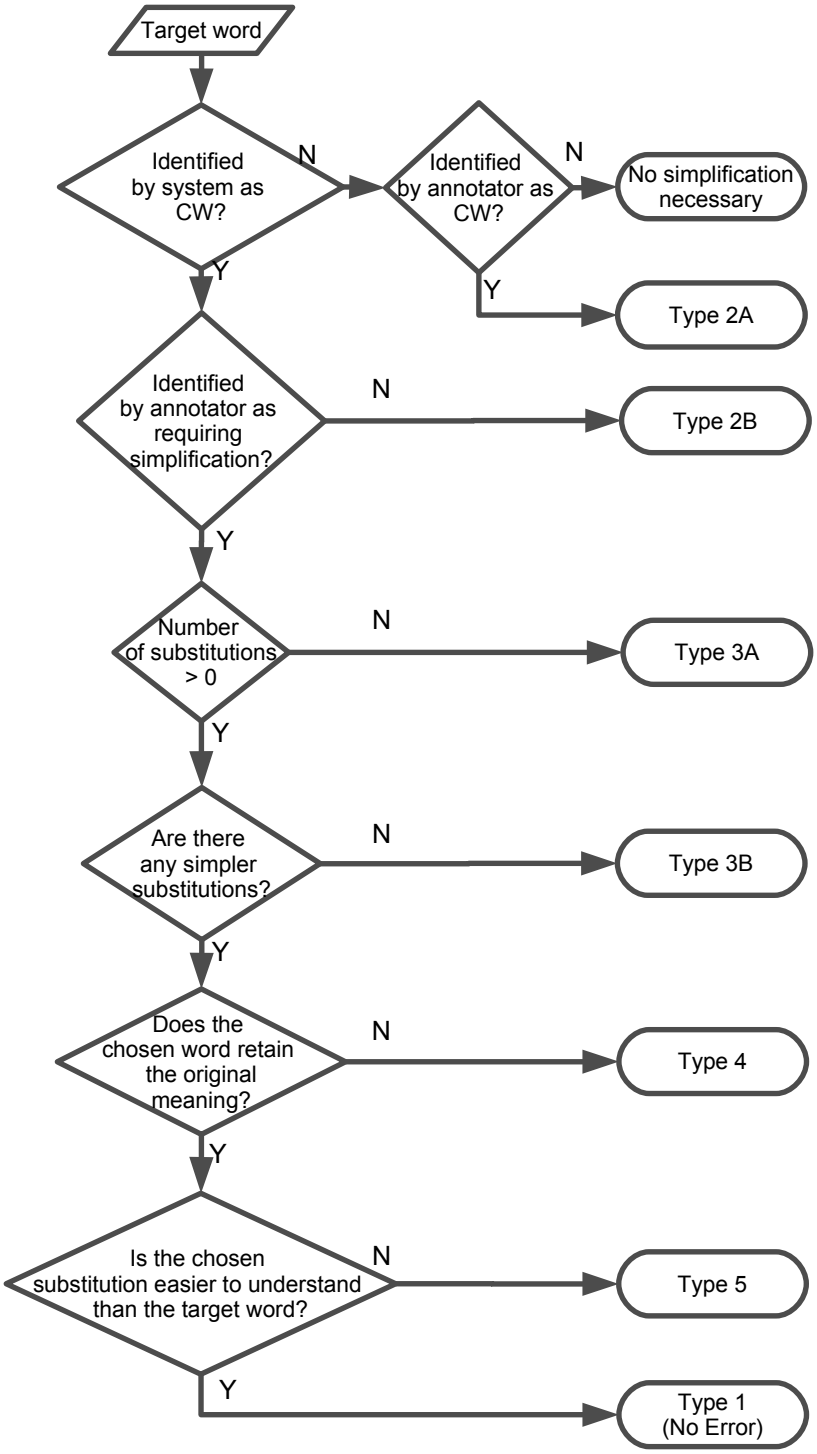


Figure 3.1: The annotation process used to determine the kinds of errors occurring during simplification operations.

Type 1: *No error* The system successfully simplified the 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 word.

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

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

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

3.2 Results

The annotation was undertaken in one sitting by the author. The workflow (Figure 3.1) was used to decide the category of each word. Some interpretation of complex/simple was necessary. As there is no standard automatic measure of LC, 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 3.2.1.

The results of the error annotation are shown in Table 3.1 and Figure 3.2. There are surprisingly few successful simplifications by the system. Notably, type 2 errors are the most frequent, indicating that CW 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 3.2 shows that each of the stages of the LS pipeline exhibits fewer errors than the previous stage. It is possible that errors early on in the pipeline masked future potential errors. To investigate, we analysed the performance of each stage individually. Figure 3.3 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, there are 183 simplification operations. 119 operations resulted in error, giving this stage an error rate of 65.03%. At the next stage (substitution generation) there are 64 ($183 - 119 = 64$) operations to take into account, of which 27 result in error, giving an error rate of 42.19%. The results indicate that each stage does indeed achieve a lower

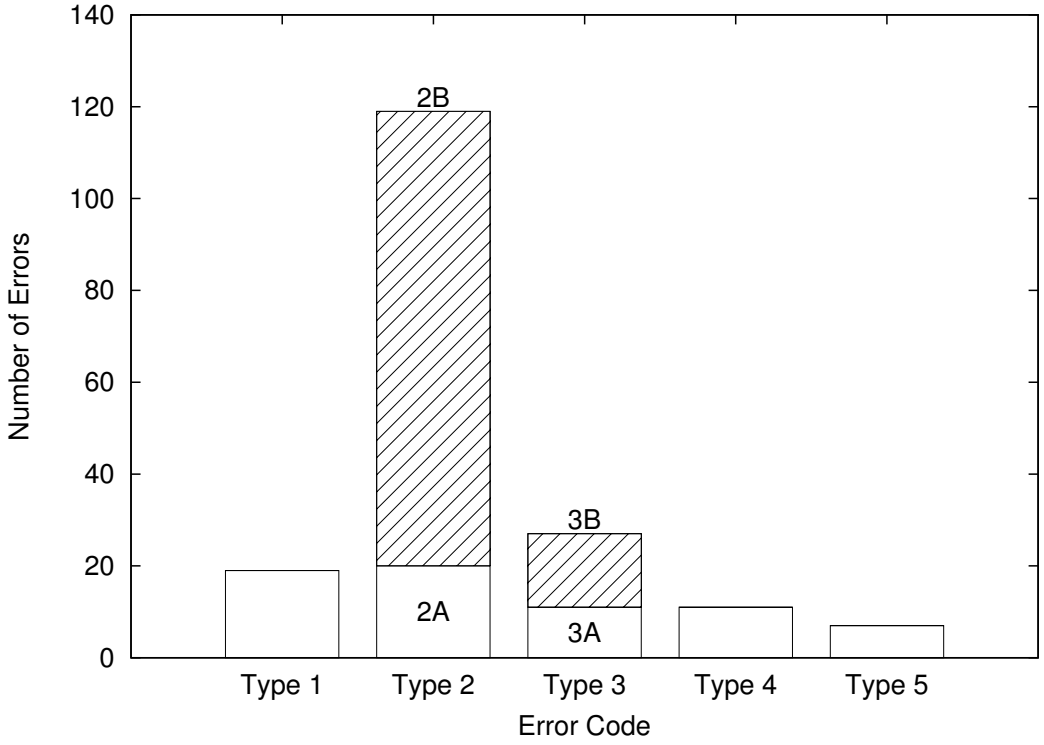


Figure 3.2: The distribution of errors between categories. N.B. The most common error is type 2, which corresponds to the first stage of the pipeline. Errors at later stages may be partially masked.

Table 3.1: The raw error data, showing the number of errors assigned to each type. In total, 183 simplification operations were identified. 164 operations resulted in some form of error.

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

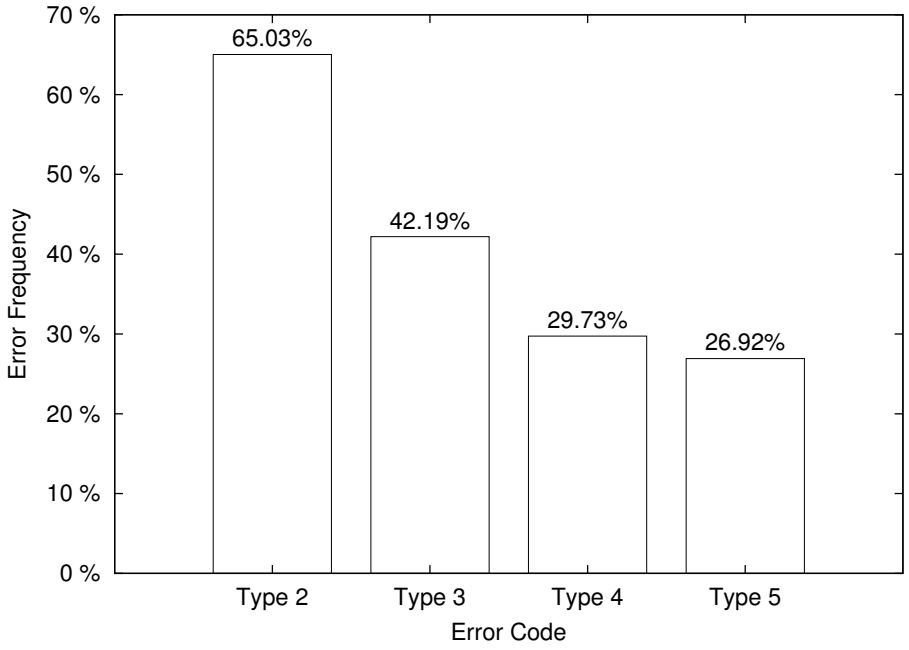
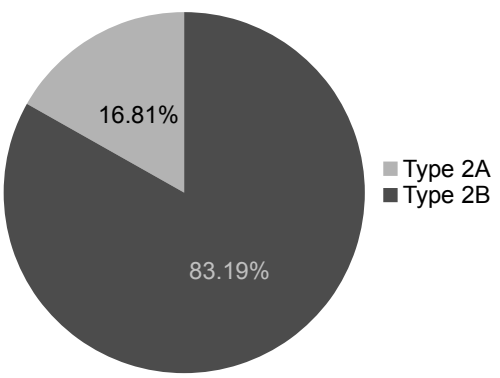
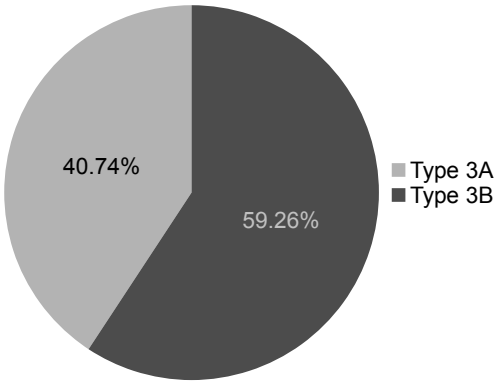


Figure 3.3: The percentage of errors occurring at each stage of the pipeline. A lower proportion of errors occurs at each stage.



(a) Type 2A vs. Type 2B



(b) Type 3A vs. Type 3B

Figure 3.4: The sub distributions for type 2 and 3 errors.

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 3.2.

3.2.1 Inter-Annotator Agreement

To assess the reliability of using human judgement we asked three annotators to assign 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 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 ($p < 0.0001$). The score 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 the wide range of choices will have contributed to the reduced kappa. Although the agreement could be improved by reducing the number of choices available to the annotators, the descriptive power of the error categorisations would be reduced.

3.3 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 which is a result which could have been altered by adjusting the threshold used in CW 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 MWEs. These errors were not found in the Kučera-Francis frequencies or in WordNet and so were assigned zero frequency and no substitutions, despite being easily understandable. Using a more comprehensive dictionary or some compound word splitting may have aided in this task.

Overall, a stronger notion of LC 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 resources provide a more accurate notion of LC. To identify CWs, LS 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 MWEs for LS. See Chapter 5 for more on MWEs.

The next most frequent error is type 3 those caused by a failure at the substitution generation stage. It may be intuitive to believe 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 3.4b indicate that Error 3B (the case of substitutions being generated, but none being sufficient to simplify the original word) is more frequent. This result shows that as well as improving the scope of resources, it is also important to directly generate simpler synonyms for CWs. Previous work in simplification has generated sets of simple synonyms (Yatskar et al., 2010; Biran et al., 2011) as discussed in Chapter 2.

Type 4 errors occurred when a word with the wrong sense was selected by the system. 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 been proven empirically not to be the case (Lal and Rüger, 2002). Here, we need a technique for WSD. Some recent publications have sought to apply WSD algorithms to the LS pipeline (Thomas and Anderson, 2012; De Belder et al., 2010) with some 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. An accurate measure of LC is important here. A way of measuring LC permits a system to sort words in order of their ease of understanding. LC changes with context and word sense and so is difficult to accurately determine. The strategy of this chapter has been to use word frequency as a measure of LC, assuming that more frequent words will be easier to understand as they are encountered more often. However, many other factors can also affect LC (Specia et al., 2012).

It is surprising to note that type 2 and 3 errors far outweigh types 4 and 5. Much of the research to date in LS has focused 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 CWs and the generation of substitutions. Early errors will be an important aspect of future research into LS. As we only identified the first error at each simplification, there may be a masking effect on the later stages which prevents erroneous cases from reaching the later modules in the pipeline. However, Figure 3.3 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.

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 listed below.

Type 2A: Because a CW 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 CW remains in the final text. The same is true 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. A nonsensical sentence may also result.

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.

The results shown here call into question the work of some research papers. It is common to see a proposition for a simplification system which uses similar techniques to those presented here, but they do not mention the high error rate which we exposed. These systems provide some improvements to the pipeline and so it is reasonable to assume that they may have a success rate higher than the 19 correct simplifications presented here. However, we might assume that there will still be a high error rate. The results of these systems should be read in the context of the high rate of errors presented here. Although some papers may present sentences which appear well formed, it is possible that these are only a fraction of the true performance of a system.

This chapter has brought to light certain dangers associated with LS. We have seen that errors can occur at each stage of the pipeline, crippling a system's efficiency. A successfully deployed LS system must take into account the errors arising from each component in 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 WSD 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, providing a robustly analysed baseline simplification system.

Chapter 4

The CW Corpus

In Chapter 2, we saw that CW identification is an important preliminary step in a LS system (See Figure 2.1). We must decide which words require simplification before a text can be simplified. The evaluation of the identification of CWs is often a forgotten task. As we saw in Chapter 3, many errors occur at this stage which result in either difficult words or poor simplifications in the final text. This chapter presents the construction of a corpus of sentences annotated with CWs (Section 4.1) which can be used to evaluate techniques for CW identification.

Previous approaches to LS (see Chapter 2) have often omitted an evaluation of their method for CW identification. This gap in the literature highlights the need for evaluation. To evaluate a system which identifies CWs, gold standard data is needed for comparison. To this end, we created the CW corpus, a dataset of 731 examples of sentences with exactly one annotated CW per sentence.

A CW is defined as one which causes a sentence to be more difficult for a user to understand. For example in the following sentence:

> "The workforce needs remuneration"

The presence of the word 'remuneration' would reduce the understandability for some readers. It would be difficult for those readers to work out the sentence's meaning, and if the reader is unfamiliar with the word 'remuneration', they will have to infer its meaning from the surrounding context. Replacing this word with a more familiar synonym, such as "payment", improves the understandability of the sentence whilst retaining the original semantics.

4.1 Corpus Design

Our corpus contains examples of simplifications which have been made by human editors during their revisions of Simple Wikipedia articles. These sentences contain one marked word which requires simplification. To help in the task of discovering instances and evaluating the corpus, we limit the discovered simplifications to one word per sentence. If an edited sentence differs from its original by more than one word, we do not include it in our corpus.

4.1.1 SUBTLEX

The SUBTLEX dataset (Brysbaert and New, 2009) is a collection of word frequencies collected from film subtitles. In this research, we assume that words which occur

Table 4.1: The results of different experiments on the SemEval LS data (De Belder and Moens, 2012), showing the SUBTLEX data's superior performance over several baselines. Each baseline gave a familiarity value to a set of words based on their frequency of occurrence. These values were used to produce a ranking over the data which was compared with a gold standard ranking using kappa agreement to give the scores shown here. A baseline using the Google Web 1T dataset was shown to give a higher score than SUBTLEX, however this dataset was not available during the course of this research.

System	Score
SUBTLEX	0.3352
Wikipedia Baseline	0.3270
Kučera-Francis	0.3097
Random Baseline	0.0157

more frequently are simpler. We compare words using the SUBTLEX frequencies. We also use SUBTLEX as a dictionary for testing word existence: if a word does not occur in the dataset, it is not considered. This may occur in the case of very infrequent words or proper nouns. The SUBTLEX data was chosen over the more conventional Kučera-Francis frequency (Kučera and Francis, 1967) as it has been shown to be more representative of natural language frequencies (Brysbaert and New, 2009). A preliminary experiment confirmed this finding. We used the SemEval LS data (De Belder and Moens, 2012) and ranked various sets of synonyms in frequency order. These were compared with human judgements of their simplicity to give the results in Table 4.1. More information is presented in Chapter 5.

4.1.2 Prior Work

The CW corpus was built following the work of Yatskar et al. (2010) in identifying paraphrases from Simple Wikipedia edit histories. Their method extracts lexical edits from aligned sentences in adjacent revisions of a Simple Wikipedia article. These lexical edits are then processed to determine their likelihood of being a true simplification. They presented two methods for determining this probability. The first uses conditional probability to determine whether a lexical edit represents a simplification. The second uses metadata from comments to generate a set of trusted revisions, from which simplifications can be detected using pointwise mutual information. Our method (further explained in Section 4.2) differs from their work in several ways. Firstly, we

seek to discover only single word lexical edits. Secondly, we use both article metadata and a series of strict checks against a lexicon, a thesaurus and a frequency dictionary to ensure that the extracted lexical edits are true simplifications. Thirdly, we retain the original context of the simplification as LC is thought to be influenced by context (Biran et al., 2011; Bott et al., 2012).

We chose to automatically mine edit histories as this method quickly provides many instances at a low cost. Another method of creating a similar corpus would have been to ask human annotators to produce hundreds of sets of sentences, and mark up the CWs in these. Human annotators are used to create further corpora in Chapter 5, where a comparison of the two methods for corpus construction will be presented.

4.2 Method

In this section, we explain the procedure to create the corpus. There are many processing stages as shown in Figure 4.1. The stages in the diagram are further described in the sections below. The edit histories of Simple Wikipedia can be viewed as a set of pages P, each with an associated set of revisions R. Every revision of every page is processed iteratively until P is exhausted. Examples of the type of simplifications extracted through this method are shown in Section 4.2.1.

Content Articles

The Simple Wikipedia edit histories were obtained from a database dump dated 4th February 2012. The entire database was very large, so only main content articles were considered. All user, talk and meta articles were discarded. Non-content articles are not intended to be encyclopædia entries and so may not always reflect the same level of simplicity as the rest of the site.

Revisions which Simplify

When editing a Simple Wikipedia article, the author has the option to attach a comment to their revision. Following the work of Yatskar et al. (2010), we only consider those revisions which have a comment containing some morphological equivalent of the lemma ‘simple’, e.g. simplify, simplifies, simplification, simpler, etc. This method allows us to search for comments where the author states that they are simplifying the article.

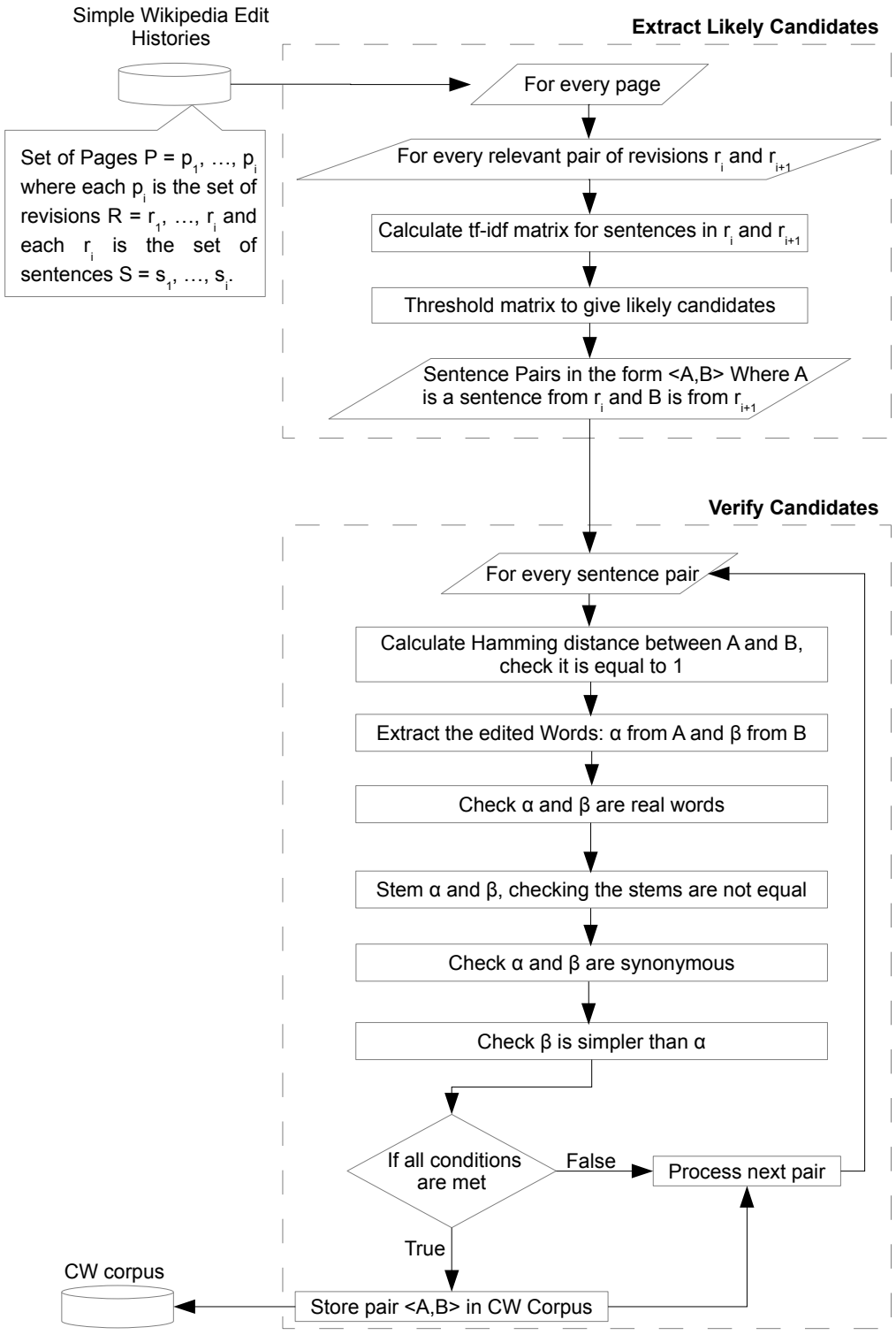


Figure 4.1: A flow chart showing the process undertaken to extract instances of LS. Each part of this process is further explained in Section 4.2. Every pair of revisions from every relevant page is processed, although the appropriate recursion is omitted from the flow chart for simplicity.

TF-IDF Matrix

Each revision is a set of sentences. As changes from revision to revision are often small, there will be many sentences which are the same in adjacent revisions. Sentences which are likely to contain a simplification will only have one word difference and sentences which are unrelated will have many different words. TF-IDF (Salton and Yang, 1973) vectors are calculated for each sentence. We also calculated the matrix containing the dot product of every pair of sentence vectors from the first and second revision. Vectors which are the same will have a score of one. As TF-IDF treats a sentence as a bag of words it is also possible for two sentences to give a score of 1 if they contain the same words, but in a different order. This is not a problem as if the sentence order is different, there is a minimum of 2 lexical edits and we will discount this pair at a later stage. TF-IDF also allows us to easily see which vectors are so different that they could not contain a one word edit. We set a threshold at $0.9 <= X < 1$ to capture those sentences which were highly related, but not exactly the same.

Candidate Pairs

The above process resulted in pairs of sentences which were very similar according to the TF-IDF metric. These pairs were then subjected to a series of checks as detailed below. The checks were designed to ensure that as few false positives as possible would make it to the corpus. Some true positives may also have been discarded, however the cautious approach was adopted to ensure a higher corpus accuracy.

Hamming Distance

We are only interested in those sentences with a difference of one word, because sentences with more than one word difference may either contain several simplifications or be a rewording. It is difficult to distinguish whether these are true simplifications. We calculate the Hamming distance between sentences (using wordforms as base units) to ensure that only one word differs. Any sentence pairs which do not have a Hamming distance of 1 are discarded.

Reality Check

The first check searches our lexicon for both pairs, ensuring that there is SUBTLEX frequency data for these words and also that they are valid words. This stage may remove some valid words, which are not found in the lexicon. However, we find it

more preferable than to allow words that are slang, vandalism or otherwise low in quality.

Inequality Check

It is possible that although two words are different, they are morphological variants of one another rather than a simplification, e.g. due to a change in tense. To identify this case, we stem both words and compare the stems. If the stems are the same, then they are unlikely to be a simplification and the pair is discarded.

Synonymy Check

Typically, LS involves the selection of a word's synonym. WordNet is used as a thesaurus to check if the second word is listed as a synonym of the first. The accuracy of this step is dependent upon the reliability of WordNet. Some valid simplifications may not be identified as synonyms in WordNet, however we choose to take this risk and discard all non-synonym pairs.

Simplicity Check

Finally, we check that the second word is simpler than the first using the SUBTLEX frequencies. All these checks result in a pair of sentences with one word difference. The differing words are synonyms and the change has been towards a word which is simpler than the original. Given that these conditions have been met, we store the pair in our CW Corpus as an example of a LS.

4.2.1 Examples

Complex word: functions
Simple word: uses
A dictionary has been designed to have one or more _____ that can help the user in a particular situation.

Complex word: difficult
Simple word: hard
Readability tests give a prediction as to how _____ readers will find a particular text.

4.3 Corpus Analysis

To determine the validity of the CW corpus, six 50-instance samples from the corpus were turned into questionnaires. The samples were selected at random, without replacement. No two questionnaires contained the same examples. One questionnaire was given to each of 6 volunteer annotators who determined whether each sentence was a true example of a simplification. If so, they marked the example as correct, otherwise they marked the example as incorrect. The binary choice simplified the task for the annotators. A mixture of native and non-native English speakers was used, although no marked difference was observed between the two groups.

We created a single validation set of 20 random instances which we gave to each annotator. This sample was randomly interspersed among their 50 examples and is used to calculate the inter-annotator agreement. This data consisted of 10 examples from the CW corpus and 10 examples which were filtered out during the earlier stages of processing. The sample gave sufficient positive and negative data to show the annotator's understanding of the task. These examples were hand-picked to represent positive and negative data and are used as a gold standard.

Inter-annotator agreement is calculated using Fleiss' kappa, as in the evaluation of a similar task presented in De Belder and Moens (2012). In total, each annotator was asked to label 70 examples. A small sample size was used to reduce the effects of annotator fatigue. Six trials took place and in total 300 instances of LS were evaluated, covering over 40% of the CW corpus.

4.4 Results

Of the six annotations, four show the exact same results on the validation set. These four identify each of the 10 examples from the CW corpus as a valid simplification as well as each of the 10 examples that were filtered out as an invalid simplification. This is expected as these two sets of data were selected as examples of positive and negative data respectively. The agreement of these four annotators further corroborates the validity of the gold standard. Annotator agreement is shown in Table 4.2.

The 2 other annotators did not strongly agree on the validation sets. Calculating Cohen's kappa (a measure of annotator agreement) between each of these annotators and the gold standard gives scores of 0.6 and 0.4 respectively. The value for Cohen's kappa between the two non-agreeing annotators is 0.2, indicating that they do not agree

Table 4.2: The results of different annotations. The kappa score is given against the gold standard set of 20 instances. The sample accuracy is the percentage of the 50 instances seen by that annotator which were judged to be true examples of a LS. Note that kappa is strongly correlated with accuracy (Pearson's correlation: $r = 0.980$).

Annotation Index	Cohen's Kappa	Sample Accuracy
1	1	98%
2	1	96%
3	0.4	70%
4	1	100%
5	0.6	84%
6	1	96%

with each other and that they agree with the gold standard to a greater degree than they agree with each other.

Analysing the errors made by these 2 annotators on the validation set reveals some inconsistencies. For example, one annotator marked as incorrect a change from the fragment 'to be discovered' into 'to be found'. However, the same annotator marked a sentence as correct which transformed the fragment 'Galileo Galilei discovered' to 'Galileo Galilei found'. Sentences of such similarity should be marked the same. This level of inconsistency and the low agreement with the other annotators shows that these annotators had difficulty with the task. They may not have read the instructions carefully or may have not fully understood the task.

Corpus accuracy is defined as the percentage of instances that were marked as being true instances of simplification (not counting those in the validation set). This value is out of 50 for each annotator and can be combined linearly across all six annotators.

Taking all six annotators into account, the corpus accuracy is 90.67%. Removing the worst performing annotator (kappa = 0.4) increases the corpus accuracy to 94.80%. If we also remove the next worst performing annotator (kappa = 0.6), leaving us with only the four annotators who were in agreement on the validation set, then the accuracy increases again to 97.5%.

There is a very strong Pearson's correlation ($r = 0.980$) between an annotator's agreement with the gold standard and the accuracy which they give to the corpus. Given that the lower accuracy reported by the non-agreeing annotators is in direct proportion to their deviation from the gold standard, the reduction is a result of the

lower quality of those annotations. The two non-agreeing annotators were discounted when evaluating the corpus accuracy, giving a final value of 97.5%.

4.5 Discussion

This corpus was developed because of a lack of similar resources. CW identification is a hard task which is even more difficult if evaluation is ignored. With this resource, CW identification becomes much easier to evaluate. The specific target application for this work is LS systems as previously mentioned. Although the corpus was developed in 2012, no similar resources have been built in the intervening time.

Methodologically, the corpus is simple to use and can be applied to evaluate current systems. Techniques using distributional semantics (Bott et al., 2012) may require more context than is given by just the sentence. This is a shortcoming of the corpus in its present form. If necessary, context vectors may be extracted by processing Simple Wikipedia edit histories (as presented in Section 4.2) and extracting the required information at the appropriate point.

A CW identification system will have to classify both simple words and CWs. When evaluating a CW identification system in the lab, it is important to present the system with both simple and complex words during training and evaluation. The corpus can be used to gain examples of simple words (those which do not require simplification) in two ways. Firstly, the simplifications which are provided by the lexical edits could be used. These are words which are known to be simpler than their original counterparts, and they are included in the corpus for this purpose. Secondly, the context of the sentence could be used as an example of simple language. If only one word in the sentence has been simplified, then the rest of the sentence is likely to not require any further simplification. This second method may give a broader view of the type of language which does not require simplification than the first method. However, the second method may also return words which are still considered complex, but have not been chosen for simplification in a given instance. It is useful to have roughly equal classes if possible and so the first method will be appropriate in most cases.

There are 731 lexical edits in the corpus. Each one of these may be used as an example of a complex and a simple word, giving us 1,462 points of data for evaluation. This is larger than a comparable data set for a similar task (De Belder and Moens, 2012). Ways to further increase the number of instances are discussed in Chapter 8.

It would appear from the analysis of the validation sets (presented above in Section 4.4) that two of the annotators struggled with the task of annotation, attaining a low agreement against the gold standard. The low agreement is most likely due to the annotators misunderstanding the task. The annotations were done at the individual's own workstation and the main guidance was in the form of instructions on the questionnaire. It may be useful to allow annotators direct contact with the person administering the questionnaire. Direct contact would allow clarification of instructions where necessary.

The corpus accuracy of 97.5% implies that there is a small error rate in the corpus. Errors occur due to some non-simplifications slipping through the checks. The error rate means that if a system were to identify CWs perfectly, it would only attain 97.5% accuracy on the CW corpus. CW identification is a difficult task and systems are unlikely to have such a high accuracy that the small error rate in the corpus will cause an issue during evaluation. If systems do begin to attain very high accuracies then a more rigorous corpus will become necessary.

Chapter 5

Lexical Complexity

In Chapter 3, we saw that a major source of errors in the LS pipeline arises from poor CW identification. Now that we have developed a corpus of complex and simple words in Chapter 4, we are going to look at strategies to better identify words which might be difficult for an end user. We will then create a generalised framework for LC which can be used for both CW identification and synonym ranking.

In readability assessment it is important to be able to determine which words will be difficult for a user to understand. Traditional formulae have relied on simple heuristics such as the Dale-Chall easy word list (Dale and Chall, 1948) or a simple syllable count (McLaughlin, 1969). Modern approaches have looked at the use of machine learning to automatically classify which words are difficult to understand (Medero, 2014; Gonzalez-Dios et al., 2014).

At the lexical level, word difficulty is referred to as *lexical complexity* (LC, as previously defined). Depending on the task at hand, LC may be used to discover complex or simple words. When identifying and examining simple words, one may also refer to the simplicity or understandability of words. In this chapter we refer to 'LC'. It is worth noting that complexity correlates positively with word length, but negatively with word frequency, whereas the inverse is true for simplicity.

The rest of this chapter is structured as follows:

- Section 5.1 presents previous work which is relevant to this chapter.
- Section 5.2 presents a large scale analysis of potential features for LC.
- Section 5.3 presents an experiment to classify words from the CW corpus as either complex or simple. We create a framework for classifying complex words which is developed through the next sections.
- Section 5.4 presents an investigation into the relative nature of complexity. We show that LC can be defined as a relative concept. We also show that considering LC as a relative concept results in higher classification accuracy on the CW corpus.
- Section 5.5 presents a further investigation into the adaptation of frequency resources for specific domains and groups. We show that tuning the frequency to a specific genre yields an increase in classification accuracy, whereas tuning the frequencies for specific user groups yields no significant increase.
- Section 5.6 presents a final adaptation of the framework to MWEs. We compare techniques for combining the features of constituent phrases. We are able to

show that a high corpus accuracy can be obtained by focusing on the features pertaining to the difficult words in a phrase. We also show that no significant effect can be obtained by removing stop words from a MWE.

- Section 5.7 presents critical discussion of the results.

5.1 Related Work

LC is useful for the domains of readability assessment and simplification. In readability assessment, it serves as an indicator of which words are easy to understand for a user. Lexical features are often combined with higher level features to give a complexity measure for sentences and documents (Medero, 2014; Vajjala and Meurers, 2014; Gonzalez-Dios et al., 2014). A comprehensive overview of readability assessment is given in Collins-Thompson (2014). LS also requires a measure of complexity to determine which words require simplification and whether the selected substitutes are easier to understand. An overview of LS is given by both Siddharthan (2014) and Shardlow (2014).

Although many have considered LC as a black and white issue, there are exceptions. In her PhD thesis, Medero (2014) argues that each word has a varying degree of complexity. To this end, she develops a regression system to classify CWs. Johannsen et al. (2012) also consider relative LC in their contribution to the SemEval 2012 LS task (Specia et al., 2012). They build a binary classifier capable of determining which is the simpler of two words. However, they do not evaluate whether the relative feature values used in their binary classifier are more effective for classification than raw feature values. At the sentence level, Vajjala and Meurers (2014) also look at building a binary classifier which they show to be effective in distinguishing between simple and complex sentences.

When developing a measure of LC, word frequency in some form is used almost ubiquitously. The first LS system (Devlin and Tait, 1998) used Kučera-Francis frequency. Modern systems typically use frequencies from both English and Simple Wikipedia (Biran et al., 2011; Johannsen et al., 2012; Sinha, 2012; Wilkens et al., 2014; Medero, 2014; Horn et al., 2014; Ligozat et al., 2013), the Google Web1T corpus (Brants and Franz, 2006; Sinha, 2012; Horn et al., 2014; Jauhar and Specia, 2012; Leroy and Kauchak, 2014) or some other large corpus analysis (Bott et al., 2012; Zeng et al., 2005; Wilkens et al., 2014). It has been suggested that frequency may be the only feature needed for determining LC (Wilkens et al., 2014). Word length is often

combined with frequency (Biran et al., 2011; Bott et al., 2012; Johannsen et al., 2012; Sinha, 2012; Zeng et al., 2005). Other common features include language models at the character level (Medero, 2014; Johannsen et al., 2012) and at the lexical level (Horn et al., 2014; Ligozat et al., 2013; Jauhar and Specia, 2012; Johannsen et al., 2012). The source corpus of a language model is very important and using corpora with simple language can improve the modelling of LC (Kauchak, 2013).

Domain adaptation has been investigated for when to split sentences for syntactic simplification (Štajner and Saggion, 2013b), where it was discovered that a model trained specifically for one type of user and genre could be used for another. Nunes et al. (2013) propose a model for domain adaptation based on frequency resources, but provide no evaluation of the adaptability of their model.

MWEs have seen little investigation into their complexity. Amoia and Romanelli (2012) look at selecting a single constituent word from a MWE for which they compute a frequency feature. To select the word they follow a set of hand-written rules which were designed using a training corpus. Although a few rules are published in their paper, the full set is not available. Recently, Abrahamsson et al. (2014) looked at the application of LC to compound words in Swedish. Compound words are linguistically similar to MWEs and so are relevant to our work. If a compound word was not present in the Swedish medical dictionary used, then a search was made for substrings of the original word and its substitutions. If more substrings of a substitution could be found in the dictionary than could be found for the original word, then the substitution was made

5.2 Features

Let us consider what kind of features are appropriate at the lexical level. Following previous work, we want to use machine learning to distinguish between complex and simple words. We need a wide set of features for each word. Some features which are appropriate at a sentence or document level, such as discourse or readability features, will not be useful here. The features below were found by looking at the literature and by using general intuition. We have taken a broad approach considering as many features as possible. These fall into the following five categories:

Frequency: We took several frequency features into account. We looked at unigram, bigram and trigram frequency. The list of frequency features is displayed in Table 5.1.

Table 5.1: The frequency features used in this work.

1	User Unigram Frequencies
2	User Bigram Frequencies (1 word after)
3	User Bigram Frequencies (1 word before)
4	User Trigram Frequencies (2 words before)
5	User Trigram Frequencies (middle word)
6	User Trigram Frequencies (2 words after)
7	Genre Unigram Frequencies
8	Genre Bigram Frequencies (1 word after)
9	Genre Bigram Frequencies (1 word before)
10	Genre Trigram Frequencies (2 words before)
11	Genre Trigram Frequencies (middle word)
12	Genre Trigram Frequencies (2 words after)
13	Document Frequency (user)
14	Document Frequency (genre)
15	TF-IDF (user)
16	TF-IDF (genre)
17	$\frac{\text{User}}{\text{Genre}}$ Unigram Frequencies
18	$\frac{\text{User}}{\text{Genre}}$ Bigram Frequencies (1 word after)
19	$\frac{\text{User}}{\text{Genre}}$ Bigram Frequencies (1 word before)
20	$\frac{\text{User}}{\text{Genre}}$ Trigram Frequencies (2 words before)
21	$\frac{\text{User}}{\text{Genre}}$ Trigram Frequencies (middle word)
22	$\frac{\text{User}}{\text{Genre}}$ Trigram Frequencies (2 words after)

Length: Length is a very well established indicator of a word's complexity. We looked at the numbers of characters, syllables, phonemes and morphemes. These features were calculated for both the wordform and lemma in every case. The list of length features is displayed in Table 5.2.

WordNet: We take into account a series of features derived from WordNet. We look at sense and synonym counts, as well as hypernym and hyponym counts. The list of WordNet features is displayed in Table 5.3.

PsychoLinguistic: Psycholinguistic metrics contain a wealth of lexical information. We incorporate data from the MRC Psycholinguistic norms (Coltheart, 1981) and the English Lexicon Project (Balota et al., 2007). These features are listed in Table 5.4.

Table 5.2: The length features used in this work.

23	Number of Characters in Wordform
24	Number of Characters in Lemma
25	Number of Syllables in Wordform
26	Number of Syllables in Lemma
27	Number of Phonemes in Wordform
28	Number of Phonemes in Lemma
29	Number of Morphemes in Wordform
30	Number of Morphemes in Lemma
31	Number of $\frac{\text{Characters}}{\text{Syllables}}$ in Wordform
32	Number of $\frac{\text{Characters}}{\text{Syllables}}$ in Lemma
33	Number of $\frac{\text{Characters}}{\text{Phonemes}}$ in Wordform
34	Number of $\frac{\text{Characters}}{\text{Phonemes}}$ in Lemma
35	Number of $\frac{\text{Characters}}{\text{Morphemes}}$ in Wordform
36	Number of $\frac{\text{Characters}}{\text{Morphemes}}$ in Lemma

Table 5.3: The WordNet features used in this work.

37	Number of Senses
38	Number of Synonyms
39	Number of Distinct Synonyms
40	Number of Hypernyms
41	Number of Hyponyms
42	Shortest Distance to Root Node
43	Longest Distance to Root Node
44	Shortest Distance to Leaf Node
45	Longest Distance to Leaf Node

Orthographic: The final feature models the orthography of English spelling. This value was created by training a language model on the British National Corpus (BNC). A high score from this model implies that the word follows normal spelling conventions, whereas a low score indicates an uncommon character pattern. The character language model is feature 56.

Each feature was calculated for every word in the CW Corpus. For each corpus entry, we calculated the feature vector for the original CW and for the simple word. We generated $1,462$ instances in total. 731 instances were in the negative 'simple' class and 731 instances were in the positive 'complex' class. We calculated the Pearson

Table 5.4: The psycholinguistic features used in this work. Deeper explanation of the features is given in the references for the English Lexicon Project (Balota et al., 2007) and the MRC Psycholinguistic data (Coltheart, 1981). LDT stands for Lexical Decision Time, this value reflects how long it takes for a subject to determine if a word exists in the English Language.

46	Age of Acquisition
47	Concreteness
48	Imageability
49	Familiarity
50	Mean Colerado Meaningfulness
51	Mean Pavio Meaningfulness
52	Mean LDT Reaction time
53	Mean LDT Accuracy
54	Mean Naming Reaction Time
55	Mean Naming Accuracy

correlation between each feature and the class label, such that a positive correlation implies an increase in LC. We were able to see the individual strength of each feature. The 15 features with the strongest correlations to the class label are shown in Table 5.5. We can see how well each feature predicts whether a word is complex or simple. We see that length based features (23–26, 28, 29) perform very well with little distinction between using the lemma and the wordform. Frequency features (1, 13–15, 17) also perform well with several ranked highly. We also see features from WordNet (37, 40) and Psycholinguistics (49). The Character n-gram Model (56) also ranked highly in correlation, showing it to be useful for discriminating between complex and simple words.

5.3 CW Identification

Now that we have a set of features from the previous section and a corpus of labelled data from the previous chapter, we can proceed to build a classifier. We will also implement two baseline techniques which are prevalent in the literature. The baseline techniques are simplifying everything (Devlin and Tait, 1998) and frequency based thresholding (Zeng et al., 2005). We want to know if machine learning techniques can be used to improve the performance of CW identification over the baseline techniques. We chose the following research question and hypothesis:

> **RQ 5.1:** How can we detect CWs without using a simple threshold or simplifying every word?

Table 5.5: The top 15 features according to individual feature correlation with the class label. A positive correlation indicates that as this feature increases, so does LC. A negative correlation indicates that as this feature increases, LC decreases.

Category	Name	Feature Index	Correlation
Length	No. Characters in Wordform	23	0.5897
Length	No. Characters in Lemma	24	0.5828
Length	No. Syllables in Wordform	25	0.5472
Length	No. Syllables in Lemma	26	0.523
Frequency	Document Frequency (user)	13	−0.4925
Frequency	$\frac{\text{User}}{\text{Genre}}$ Unigram Frequencies	17	−0.4773
Length	No. Morphemes in Wordform	29	0.4243
Frequency	Document Frequency (genre)	14	−0.4128
Orthographic	Char n-gram Model	56	−0.3707
WordNet	No. Senses	37	−0.3665
Frequency	User Unigram Frequencies	1	−0.3421
Frequency	TF-IDF (user)	15	0.3418
Psycholinguistic	Familiarity	49	−0.3405
Length	No. Phonemes in Lemma	28	0.3373
WordNet	No. Hypernyms	40	−0.3356

RH 5.1: A feature based machine learning technique will provide higher accuracy in the CW identification task than the baseline techniques.

Simplify Everything: The first method takes a brute force approach in which a simplification algorithm is applied to every word. This approach assumes that words which are already simple will not require any further simplification. The simplification may be limited to certain classes of words such as nouns and verbs.

A standard baseline LS system was implemented following the system proposed by Devlin and Tait (1998). This algorithm generated a set of synonyms from WordNet and then used the SUBTLEX frequencies to find the most frequent synonym. If the synonym was more frequent than the original word then a substitution was made. This technique was applied to all the words. If a CW was changed, then it was considered a true positive; if a simple word was not changed, it was considered a true negative.

Thresholding: The second technique is frequency-based thresholding. This technique relies on each word having an associated familiarity value provided by the SUBTLEX corpus. Whilst this corpus is large, it will never cover every possible

word, and so words which are not encountered are considered to have a frequency of 0. The effect on comparison is negligible as the infrequent words are likely to be complex.

To distinguish between complex and simple words, we implemented a threshold. The threshold was learnt from the CW corpus by examining every possible threshold across a training set of examples taken from the corpus. Firstly the training data was ordered by frequency, then the accuracy of the algorithm was examined with the threshold placed in between the frequency of every adjacent pair of words in the ordered list. Accuracy is defined as the proportion of data that was correctly classified. We repeated the experiment using 5-fold cross validation and determined the mean threshold. The final accuracy of the algorithm was then determined on a separate set of testing data. The training data contained 146 instances from the corpus and the testing data contained 585 instances.

K-Nearest-Neighbour (KNN): A KNN classifier retains a labelled set of example feature vectors in memory. When classifying a new instance, it returns the class of the closest feature vector it knows of. In a KNN classifier, the K closest feature vectors are examined and the most common class label is assigned to the input. In our experiment, we set K to 2. This classifier is capable of learning complicated relations between two classes. However it is slow, uses lots of memory and loses performance as the feature space grows. We used WEKA (Hall et al., 2009) with the standard implementation of this classifier (Aha et al., 1991).

Naïve Bayes: A naïve Bayes classifier (John and Langley, 1995) assumes that each feature is conditionally independent of all others. The assumption is usually incorrect, and certainly so in our case (as we have several features from very similar categories). Nonetheless, naïve Bayes often performs well when this assumption is violated. The classifier learns the probability of each class given the feature values.

Support Vector Machine (SVM): The SVM (Platt, 1998) is a popular tool for classification. A boundary is learnt between two classes according to features in the training data. The SVM finds the best discriminating boundary between the classes. This boundary can be made non-linear by applying transformation kernels in the feature space. We did not experiment with the parameters of this model, but instead used the default parameters provided in WEKA. It is possible that we could have gained some improvement by performing a grid search over

these parameters. Similar to the KNN classifier, performance decreases as the number of features increases.

Rule Based Classifier: This classifier learns a series of rules based upon the feature values in the training data. We used the JRip implementation in WEKA. See Cohen (1995) for further details. We selected this class as rule based learners tend to perform well with lots of features and do not need lots of fine tuning.

Decision Tree: This classifier is similar to the rule based classifier above. A decision tree is an ordered set of rules which are arranged and processed in a tree structure. We used the J48 implementation from WEKA (Quinlan, 1993).

Random Forest: A random forest is so called because it is a collection of randomised decision trees. Each individual tree may learn different rules that separate the data in different ways. New instances are classified by polling each decision tree and the results are combined into a final class label. We used the standard implementation of this classifier in WEKA (Breiman, 2001). This classifier was selected as it is known to give higher performances than a stand-alone decision tree.

The results of the experiments in identifying CWs are given in Table 5.6. We performed 5-fold cross-validation for the 'Simplify Everything' and 'Thresholding' techniques. We performed a 10x10 fold cross validation for the machine learning techniques. The machine learning techniques were much better than the baseline techniques at CW identification in all respects except for recall. Therefore, we should accept RH 5.1. The high recall in the simplify everything method is expected — as it explicitly assumes that every word in the text is a complex word. The high recall in the thresholding method implies that the best threshold favoured minimising false negatives over minimising false positives. The Random Forest yields the highest corpus accuracy and precision. This method is significantly better ($p < 0.0001$) than the next best technique (the decision tree).

5.4 Relative Complexity

During the analysis of the results from the previous section, we examined and compared corpus entries from the CW Corpus. It became apparent to us that, although simplification took place within each entry, each entry reflected a different level of

Table 5.6: The results of classification experiments for the three systems as well as the baselines. The highest values for Accuracy, Precision and Recall are reported in bold. Standard deviation (STD) is measured in percentage points.

System	Accuracy	STD	Precision	STD	Recall	STD
Thresholding	78.54%	1.38	70.88%	1.36	96.97%	0.56
Simplify Everything	82.07%	0.77	73.75%	0.84	**99.60%**	**0**
SVM	82.82%	0.32	82.41%	0.43	83.69%	0.23
KNN	82.92%	0.42	79.46%	0.45	89.00%	0.53
Naïve Bayes	86.10%	0.19	87.88%	0.16	83.88%	0.43
Rule Based	89.22%	0.58	89.98%	1.26	88.47%	1.17
Decision Tree	89.36%	0.47	89.74%	0.76	89.03%	0.89
Random Forest	**92.00%**	**0.57**	**91.72%**	**0.78**	92.49%	0.82

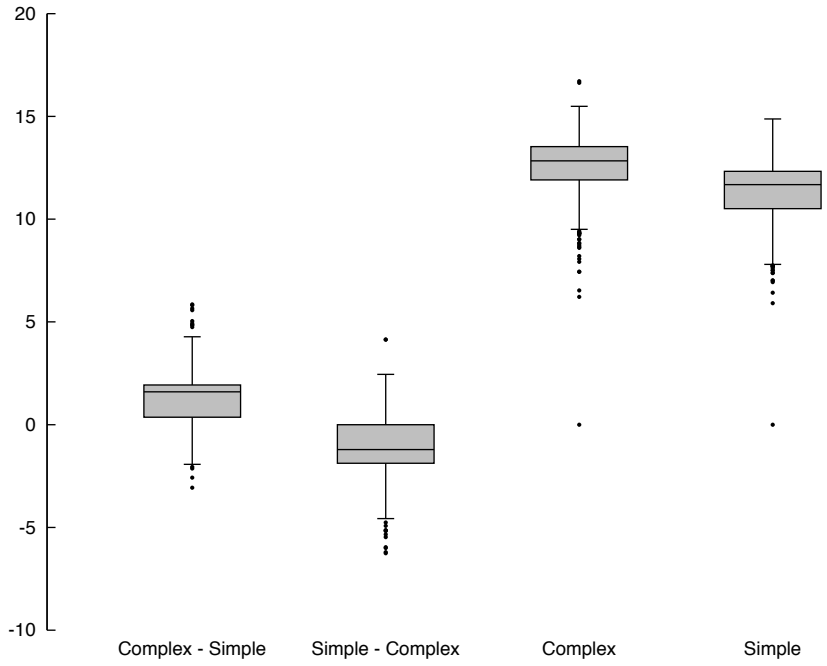
simplification. For a target user, a simple word from one pair may be more difficult to understand than a CW from another. This led to the following research question and hypothesis:

RQ 5.2: Is LC a relative concept?

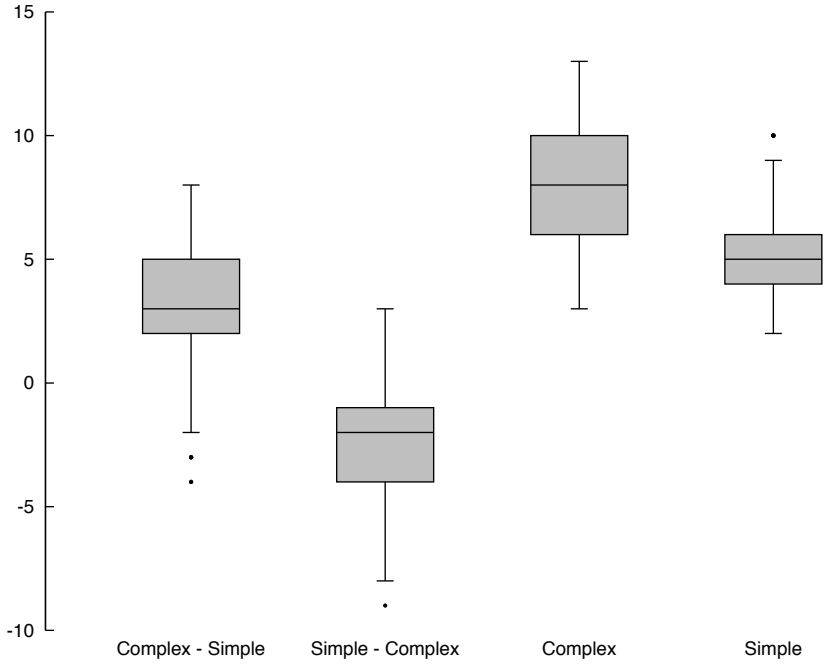
RH 5.2: The relative feature values of two words will be a stronger predictor of LC than the raw value.

To investigate this hypothesis, we first produced the Tukey box plots in Figure 5.1. For each word, we looked at the length in characters and the frequency in English Wikipedia. To produce the box plots, we created two datasets. One for the simple words in the corpus and another for the CWs. We calculated the length in characters and the log frequency of each word. We then separated each dataset into two equal portions. For the first half of the corpus we subtracted the features of the CWs from the simple words. We did the opposite for the second half of the corpus and substracted the features of the simple words from the CWs. The data is depicted in the box plots in Figure 5.1. It is clear from these box plots that a greater separation of the data is attained by subtracting one feature from another within a corpus pair.

We calculated all 56 of the features from Section 5.2 for each entry in the CW corpus. These were labelled and stored appropriately. This gave us two data sets. The first is a set of features which are labelled as either a simple word subtracted from a CW, or a CW subtracted from a simple word. The second dataset is a set of features corresponding to either a CW or a simple word. The first dataset contains 731 instances, the second contains $1,462$.



(a) Log Frequency



(b) Length

Figure 5.1: Tukey box plots showing the distribution of the raw and relative data for both frequency and length. In each graph, the first two boxplots represent the relative data and the second two box plots represent the raw data. Although some separation is present for the raw data, it can be seen that the separation increases in the relative data.

Table 5.7: The results of classification. The relative feature values resulted in a statistically significant ($p < 0.001$) increase in accuracy. STD is measured in % points.

Corpus	Accuracy	STD
Relative	95.44%	0.56
Raw	89.36%	0.76

We used these datasets as the input to a machine learning tool built in WEKA. We performed 10x10 fold cross validation on each dataset, using a J48 decision tree as the classifier. This gave us the results in Table 5.7. It can be seen that taking the difference between the features of two words is a significantly better predictor of complexity than using the raw feature values. A two tailed t-test allows us to reject the null hypothesis that there is no significant difference between results for each dataset. This allows us to accept RH 5.2.

These results are further discussed in Section 5.7. The rest of this chapter will look at two extensions of this work. Firstly, we will examine the role of the frequency resources and look at their adaptability. Secondly, we will look at techniques for applying LC to MWEs.

5.5 Frequency Resources

LC has been applied to many text genres and many user groups. A key method of adapting to the user and genre is to modify the underlying frequency resource to reflect the inherent vocabulary of the target. For the genre, we wish to model the type of language typically occurring in the area. The modelling may be done by means of calculating frequencies from a large corpus of texts from within the genre. For users, we wish to model the type of language a user is familiar with. We may look at corpora of documents written for or by the target user group to model the language appropriately. Our next research question is as follows:

> **RQ 5.3:** How do environmental factors affect corpus frequencies?

This led to the following two research hypotheses:

> **RH 5.3:** Users' decisions as to which words are difficult to understand can be modelled by adapting a frequency resource to their needs.

> **RH 5.4:** Users' decisions as to which words are difficult to understand can be modelled by adapting a frequency resource to the genre of the text.

We tested these hypotheses using the framework described in the previous section. We selected two domains following the criteria that they should be in need of simplification for a general audience and that there should be a sufficiently large corpus of text from which to create frequency resources. We selected English Wikipedia and articles from BioMed Central as our two domains. We calculated genre specific frequencies for each domain using the large volume of text that is available in each case.

To create two corpora, each text was processed automatically by an in-house human aided machine simplification system. In each case, 100 single word edit instances were identified. Each entry in the corpus consisted of two sentences which were identical, except for one single word edit. No judgement was made about the complexity of the substitution during the creation of the corpus. Only word sense and grammaticality were taken into account.

To label the corpora we used the CrowdFlower platform (Van Pelt and Sorokin, 2012) to crowdsource annotations. To combat selection bias, the order of the instances was randomised as well as the order in which the two sentences in each instance appeared. 211 annotators were shown examples from the corpus until we had at least 11 judgements for each instance. Due to the randomisation, some instances had more than 11 judgements.

To distinguish between two separate user groups, we presented each user with a short reading comprehension test as well as a short grammar test. The reading comprehension test had 6 questions of increasing difficulty. The grammar test had 15 questions of increasing difficulty. We separated the annotators into two groups based on their scores. Each corpus instance had at least 5 annotations from both groups.

To test RH 5.3, we conflated the users' decisions across corpora and separated the labels by user group. The result was two sets of labels, henceforth called 'low ability' and 'high ability'. Labels were based on a majority decision by the annotators. 6 instances were excluded as no majority emerged. To test RH 5.4, we conflated the user data across genres, which produced two further datasets, 'all users Wikipedia instances' and 'all users Biomedical instances'. Again, labels were based on a majority decision, with 6 instances removed due to no majority. See Table 5.8 for further clarification on the content of each label set.

Overall, we were able to run two experiments to test RH 5.3 and two experiments to test RH 5.4. For each experiment we constructed a baseline model and an adapted

Table 5.8: The content of each set of labels (in bold face) can be found by following its row or column.

	Wikipedia Labels	**Biomedical Labels**
High Ability Labels	Wikipedia High Ability Labels	Biomedical High Ability Labels
Low Ability Labels	Wikipedia Low Ability Labels	Biomedical Low Ability Labels

model. The models were all adapted by using appropriate frequency resources. As corpus size can influence predictive power for LC (Kauchak, 2013), we scaled all corpora to be the same size (34 million tokens). For the baseline model, we randomly selected two appropriately sized sections of the BNC. The BNC was used to minimise domain specificity. For the various adapted models we took the following strategies:

Low Ability Users: We used frequency counts taken from Simple Wikipedia. Simple Wikipedia is specifically written for users who will not understand English Wikipedia. At 34 million tokens, Simple Wikipedia was the smallest frequency resource available and all other resources were scaled to the same size.

High Ability Users: We used frequency counts taken from English Wikipedia, which is often a difficult to understand resource and represents an accomplished level of English proficiency.

Wikipedia: We used the same frequency counts as before from English Wikipedia. Here, they were used as they best reflect the text in the corpus.

BioMedical: We used frequency counts taken from articles in BioMed Central. The counts are the best reflection of the underlying corpus domain.

We used WEKA with 10x10-fold cross-validation and a J48 decision tree to evaluate each baseline and each adapted model. The results of the adaptation are displayed in Table 5.9. They clearly show that whilst the adaptation for the users had no significant improvement, there was a significant improvement when adapting for genre. Therefore, we can reject RH 5.3 in favour of the null hypothesis, but we should accept RH 5.4.

We also note that the accuracy reported here is much lower than that in Sections 5.3 and 5.4. We are using a different corpus here and different resources for the frequency features, both of which may combine to make the task more difficult for the classifier.

Table 5.9: The results of the user and genre adaptation experiments. The first four columns show the parameters of each experiment. The last three show the results. Each experiment consisted of two runs of 10x10-fold cross-validation, between which one frequency resource was changed. The P Values show the level of significance between each set of runs. Whilst user adaptation did not result in a significant improvement, genre adaptation provided a significant improvement for both Biomedical and Wikipedia text. STD is measured in % points.

Corpus	**User Resource**	**Genre Resource**	**Labels**	**Accuracy**	**STD**	**P Value**
Combined	BNC User	BNC Genre	Low Ability	67.57%	1.63	0.3793
Combined	Simple Wikipedia	BNC Genre	Low Ability	66.64%	2.82	
Combined	BNC User	BNC Genre	High Ability	67.57%	1.63	0.5066
Combined	English Wikipedia	BNC Genre	High Ability	68.17%	2.29	
Biomedical	BNC User	BNC Genre	Biomedical	63.40%	2.77	0.0022
Biomedical	BNC User	Biomedical	Biomedical	68.12%	3.13	
Wikipedia	BNC User	BNC Genre	Wikipedia	63.48%	2.54	0.0022
Wikipedia	BNC User	English Wikipedia	Wikipedia	68.88%	2.55	

5.6 Multiword Expressions

In this final section of experiments we will look at the application of LC to MWEs. We have focussed excusively upon compositional MWEs, as these were most apparent in corpora. Single word edits are much easier to process programmatically and so far we have only focused on these. However, many words cannot be simplified by a single lexical edit and a phrase must be used instead. This phenomenon was particularly apparent whilst creating the biomedical corpus. Take, for example, the word 'hypertension'. There is no single word which could be substituted to simplify this word, however the phrase 'high blood pressure' seems to suffice well.

MWEs are notoriously difficult to handle in a text. Although they may occur infrequently, they are often made up of words which do occur frequently. It is not obvious how to assign complexity to these phrases. One option would be to treat the phrase itself as a lexical unit and calculate features based on this unit, however many problems arise as the features used in this work are not designed to accommodate multiword units. For example, the string 'hypertension' occurs 3 times more often than 'high blood pressure' in Wikipedia as well as being five characters shorter, spaces not included. These mismatches between intuition and feature values imply that MWEs should not be treated as lexical units when calculating LC. Another strategy would be to use the features of the constituent words in a MWE to make inferences about the features of the MWE itself. A similar technique has been successfully explored for compound words in the Swedish language (Abrahamsson et al., 2014). We formulate the following research question and hypothesis:

RQ 5.4: How can we assign feature values to MWEs for LC?

RH 5.5: MWEs derive their LC features from their constituent words.

In the work on Swedish compounding previously mentioned, the compound words are assigned complexity values based on the number of substrings found in a dictionary. We would expect most of our substrings to be found in a dictionary, so we opt to take the average of the features for each component word and compare this with both a random baseline (where random feature values are assigned for each instance) and the relevance rules technique (Amoia and Romanelli, 2012). We seek to explore the general effects that constituent words have on a MWE. Particularly, we are interested in the effect on classification of features which signify simple or complex words. We came to the following Research Question and Hypotheses:

RQ 5.5: How do the features of constituent words combine to create features for a MWE?

RH 5.6: The most difficult features in a MWE will cause the expression to be difficult to understand.

RH 5.7: The most simple features in a MWE will cause the expression to be simple to understand.

To test these hypotheses, we created a corpus of 50 sentences each with a single edit using a MWE. The examples from the corpus were given to annotators using the crowdsourcing platform as previously described. We aggregated the decisions of all users who annotated the data and took a majority decision as to which version of each sentence was the easiest to understand. The edits were both single to multiword and multiword to multiword. All the edits were for sentences from the biomedical domain.

For the averaged values, we calculated the arithmetic mean using the standard formula. We implemented the relevance rules proposed by Amoia and Romanelli (2012). Although only a small subset of the rules are published, they were sufficient for the cases in our corpus. These rules seek to find the most relevant lexeme within a MWE and the features are calculated based on this lexeme alone. As we were only dealing with 50 data instances, we applied these rules by hand, although a programmatic implementation would be feasible for larger tasks. We used a 10x10-fold cross-validation scheme and the features proposed earlier in this chapter. We used the same parameters as for the adapted biomedical domain experiment from Table 5.9. The results of this experiment can be seen in Table 5.10, which shows that the relevance rules yield a strong performance within our framework. This result implies that single words within each MWE give rise to the features of that expression, which in turns allows us to accept RH 5.5.

To investigate RH 5.6 and RH 5.7, we look at the effect of maximising and minimising features with respect to the correlation of the features with the complex class label. A positive correlation implies that as the feature value gets larger the word becomes more complex. A negative correlation implies that as the feature value gets larger, the word becomes less complex. These features are calculated before the relative complexity model is employed. Maximising the feature value with respect to correlation with the class label will give precedence to the most complex features, whereas minimising the feature value will give precedence to the most simple features. Results were computed using 10x10-fold cross-validation as before. The results are shown

Table 5.10: The results of averaging feature values for MWEs. The relevance rules gave a strong improvement in performance. The improvement implies that the LC of a MWE is derived from the most relevant lexical unit within the expression. We also see that minimisation is significantly stronger than maximisation $(p < 0.0001)$ implying that the most complex features dictate the complexity of a MWE. Standard deviation is measured in percentage points.

Technique	Accuracy	STD
Random Baseline	44.65%	6.55
Arithmetic Mean	54.80%	7.58
Relevance Rules	69.25%	4.53
Minimisation	45.30%	5.55
Maximisation	62.60%	9.48

Table 5.11: The effect of removing stopwords from MWEs for each technique. Overall, only minimisation showed a significant change from removing stop words. For most cases removing stop words will have no significant effect on the final outcome. More data is required to further investigate this hypothesis.

Technique	Change	P Value
Arithmetic	+4.40%	0.1555
Minimisation	+6.00%	0.0278
Maximisation	−1.75%	0.664

in Table 5.10. It is clear that maximisation is significantly better than minimisation, which implies that the difficult features cause a MWE to become hard to understand and allows us to accept RH 5.6 and reject RH 5.7.

We also looked at the effect of removing stop words from MWEs. Each technique was recalculated with stop words discounted. We found a small increase for the minimisation technique, but no significant change for the other two techniques. We did not include the relevance rules in this technique as stop words may be taken into account in the rules. The results can be seen in Table 5.11.

5.7 Discussion

In Section 5.2, we collected and analysed a set of features for CW identification. These features were drawn from five categories. Each category provided at least one feature which correlated well with the class label. No feature exhibited an overwhelmingly positive correlation. The most informative feature category was the length features, although frequency also ranked highly. In Table 5.5 we show only the top 15 features,

as the correlations reduced to less important levels after this point. Some features performed very poorly in our correlation assessment, having little or no correlation with the class label at all. These features could be removed from the classification scheme without impact on the final results.

For the length features, the number of characters and the number of syllables in a word rank at the top in our correlation analysis. In both cases, the length of the word-form scores higher than the length of the lemma, however the difference is not great and may not be significant. The high performance of this category of features shows that length is very important in the identification of CWs. This finding is intuitive, as we often see that long words are replaced by shorter words to help a reader. In simplification, length is sometimes combined with frequency to give an indicator of LC (Biran et al., 2011).

Document frequency, for both user and genre frequencies, correlates well with the class label. Document frequency ranks more highly than unigram frequency for both the user and genre resources. This result implies that unigram frequencies may be skewed by individual documents which mention them repeatedly. Using the document frequency creates a smoothing effect and gives a more true representation of how likely a word is to occur. Unigram frequency is sometimes used to represent how likely a word is to appear (Devlin and Tait, 1998). Our results indicate that for simplification, document frequency may be a better feature than unigram frequency. Another frequency feature which performs well is feature 17: $\frac{\text{User Unigrams}}{\text{Genre Unigrams}}$. This feature combines two sources of information, which may account for its performance.

The WordNet category yielded two features in the top 15. These were the number of senses and the number of hypernyms. Both have a negative correlation with LC implying that more synonyms or hypernyms leads to lower LC. At first, a negative correlation seems counterintuitive. We might expect that a word with more senses would be more ambiguous and hence more complex. Similarly, we may expect that a word with few hypernyms would be more difficult for a user, as they would have fewer points of reference by which to remember the word. However, that is not so. If we consider a familiar word such as ‘run’, we can imagine that there are many senses of this word. To run a race. To run in an election. To run a tap. To run a successful business. Each is a different meaning of the word, yet we would not expect a user to have trouble with ‘run’. If we consider a less familiar word, say ‘remunerate’, we can see that there is only one sense: “The employer remunerates the worker”. No other senses exist. Devlin and Tait (1998) claimed that this effect may mean that WSD would not be necessary for LS. Later efforts have shown that disambiguation is required (Thomas and Anderson, 2012; Biran et al., 2011; Bott et al., 2012).

From the group of psycholinguistic features, word familiarity ranks as the strongest correlated feature. Familiarity is the only feature from this group which makes it into the top fifteen features. These features were based on external resources which listed the feature values against words. The resource may have limited the predictive power of these features as some words were not available and so were assigned a dummy value. A more extensive feature resource may yield a higher correlation. The collection of these resources is based on extensive studies involving many participants, so collecting extra information for infrequent words may be difficult.

The character n-gram model gives a moderate negative correlation with the class label. A negative correlation implies that as the sequence of characters in a word becomes more likely, the complexity of a word reduces. This is expected as common sequences of characters will be easier for a user to process. There may also be some self-selection as words with common character sequences will be more memorable and hence will be used more frequently, making them easier to understand. Words with infrequent character sequences are harder to process and remember and hence lead to CWs.

Our main feature categories contained several features which are highly correlated with each other. For example, the length of a wordform versus the length of a lemma will always be similar. We used many similar features as we wanted to discover which features might be best for the purpose of identifying CWs. The overlap introduces a degree of redundancy into our classification scheme. Redundancy can be dangerous as it increases the number of dimensions in a classification problem. If there are more dimensions then there is a larger space within which to classify and a lower density of examples in that space. We could have reduced the dimensionality by either performing Principal Components Analysis (Jolliffe, 1986) or by simply eliminating features which were below a certain threshold for correlation. It is a possibility that our corpus accuracy could have improved if we had undertaken some dimensionality reduction. However, we had a high score in the first instance. The aim of this experiment was not to get the highest possible score on the corpus but instead to test RH 5.1, which we were able to do without dimensionality reduction.

Using the feature set described in Section 5.2, we tested several machine learning frameworks using the WEKA package. We were able to prove RH 5.1 and show that there is a better way of identifying CWs than either simplifying everything or thresholding on frequency values. The results from Chapter 3 indicate that the identification of CWs is particularly difficult in a non-controlled environment. It may be the case

that our experiment overestimates the performance of each technique and that the best reported rate may be lower in practice. Nonetheless, we have shown that machine learning techniques have the ability to outperform other, more naïve techniques.

The random forest gives the best performance of any technique. A random forest is a collection of decision trees, each of which is tuned to some small subset of the feature space. It is unsurprising that the random forest outperforms the decision tree as the random forest uses the results from many decision trees. Conversely, the SVM and KNN classifiers were not strong in their performances. The KNN is capable of distinguishing between closely related classes, however some examples may be misclassified if there are insufficient similar examples in the training data. The SVM is again capable of learning complex relations between classes. There are several parameters associated with the SVM which can be fine tuned to improve its performance. We chose to use the standard values as provided by WEKA. It is possible that the performance of the SVM could be improved by using a grid search to find better values for these parameters.

The thresholding method performed significantly worse $(p < 0.0001)$ than the 'simplify everything' method. Thresholding takes more data about the words into account and would appear to be a less naïve strategy than blindly simplifying everything. The thresholding used here may be limited by the resources, and a corpus using a larger word count could yield an improved result. Simplifying everything naturally gives a very high recall as almost every CW will have a simpler alternative. The precision of this technique is much lower than its recall and reflects a high degree of simple words which are falsely identified as complex.

In Section 5.4, we were able to show that LC is a relative concept, most easily defined between two words. Many previous efforts to define a LC measure have looked at defining LC as an absolute concept, such that a classifier should decide for each word whether it is simple or complex. Two such approaches from the literature both report an f-score of just over 0.8 on their respective datasets (Grabar et al., 2014; Wilkens et al., 2014). Although a comparison of these techniques is left to future work, the results presented here imply that these techniques could both be improved by treating LC as relative rather than absolute.

In Section 5.5, we were able to show that frequency resources should be adapted to reflect the genre of the corpus. We were also able to show that adapting the frequency resources to the users' vocabulary style had no significant effect. It should be noted that it is easier to adapt a resource to a genre than it is to adapt a resource to a specific

user. It may be the case that our user groups were either too widely defined or the frequency resources were not suitable for the given users.

That being said, the English Wikipedia resource was used for adaptation in both the genre experiment (to reflect the Wikipedia text) and the user experiment (to reflect the high ability users). If the user groups were not well defined, then we would have expected to see a similar result in the user experiment as we saw in the genre experiment. However this was not so. The genre experiment resulted in a highly significant result, whereas the user experiment resulted in a highly insignificant result. Therefore we would conclude that the groups were well defined enough for the context of this experiment and that adaptation between low and high ability users is not a profitable venture. It may be possible to adapt resources for other groups, and further analysis of groups from differing backgrounds and needs is necessary to justify this claim.

Section 5.6 showed that MWEs can be incorporated into a LC scheme with relative ease. By using the relevance rules we were able to get a corpus accuracy for the MWEs (69.25%) similar to the one we attained for the adapted biomedical domain experiment (68.12%). This result implies that certain key words within a MWE contain the key features for classification of LC. Further work into these relevance rules would be profitable to improve the accuracy of our classification scheme.

We were also able to show that maximising the feature values with respect to the label correlation was significantly better than minimisation. This result shows that the most difficult features of the constituent words in a MWE have the most impact on the LC of the expression. Stop words were found to have little effect on the performance of each technique.

Chapter 6

Generating Good Substitutions

We have previously seen that the LS pipeline produces errors at each stage. We have already examined techniques to reduce Type 2 and Type 5 errors, which depend on LC. In this chapter we wish to address the Type 3 and Type 4 errors, which depend upon the substitutions which might be generated. Type 3 errors corresponded to the case where no relevant substitutions could be found for a word. Type 4 errors corresponded to the case where a substitution altered the meaning of a sentence.

We have split this work into two sections: Section 6.2 focuses on the generation of substitutions for LS. Section 6.3 focuses on WSD strategies for LS. In Section 6.2, we define the concept of thesaurus coverage, i.e. how many words in a thesaurus have a simpler synonym? We then attempt several strategies to improve the coverage of WordNet using automated and hand crafted resources. We combine several large thesauri and show that the result leads to a positive increase in thesaurus coverage.

In the LS literature, we see two clear approaches to the combined task of substitution generation and WSD. The approaches can be defined by which operation they perform first. In the first case, a system will perform substitution generation, yielding a set of candidate replacements for the original term. The system will then proceed to select which replacements are suitable for the original context (De Belder and Moens, 2012; Biran et al., 2011). In the second case, the process is reversed. The system will first perform WSD, grounding the original term in a sense inventory (such as WordNet or BabelNet). It will then use the relations of the sense inventory to generate a suitable set of replacement terms (Bott et al., 2012; Thomas and Anderson, 2012). Although we have not directly compared these two methods, our results in this chapter give insights as to best practice. These insights are further discussed in Section 6.4.

In Section 6.3, we compared several techniques which can be found in the LS literature. We used three evaluation methods, each of which highlights a different aspect of the WSD task. We evaluate each of our WSD techniques using each of the evaluation methods.

6.1 Related Work

Substitution generation is an ancient task, which can be traced through the history of thesauri. The first work which we might refer to as a thesaurus dates back to Philo of Byblos in the second century A.D. (Baumgarten, 1981). A later example is the Amarakosha, from the 4th century A.D. (Nair, 2011). The first thesaurus in English

is Roget's Thesaurus (Roget, 1852). In information retrieval, thesauri are used as controlled vocabularies for indexing purposes (Baeza-Yates and Ribeiro-Neto, 1999). The Simple Knowledge Organization System (SKOS) (Miles and Bechhofer, 2009) is often used to organise technical thesauri for interpretation by machine. Thesauri typically only deal with the synonym relation, although technical thesauri often capture loose hyper/hyponymy through the use of the broader/narrower term relation.

Previous work has focused on learning substitutions from large scale corpora. Often, paraphrases have been learnt which can later be applied to a sentence. For example, Yatskar et al. (2010) mine the edit histories of Simple Wikipedia for instances of simplifications. Paraphrasing has also been applied for technical medical language (Elhadad and Sutaria, 2007; Deléger and Zweigenbaum, 2009) with a view to enhancing understandability for lay readers.

With regards to WSD, this chapter can only cover a fraction of the ground required to do it justice. For WSD we recommend the survey of Navigli (2009). The first LS system (Devlin and Tait, 1998) did not make any attempt at disambiguation, but instead relied on the hypothesis that complex words would typically be monosemous. De Belder et al. (2010) applied WSD to LS via the 'Latent Words Language Model' (Deschacht et al., 2012), however with limited results. Context vectors were first applied for synonym selection (Biran et al., 2011) and later for sense selection (Bott et al., 2012). The latter was performed for Spanish as part of the wider Simplex project (Saggion et al., 2013). SenseRelate (Pedersen and Kolhatkar, 2009) was also applied for the LS pipeline (Eom et al., 2012) in a reading aid tool which sought to improve the vocabulary acquisition of second language learners. Finally, PPR (Agirre and Soroa, 2009) was applied to LS in an attempt to gather a reduced vocabulary set for English (Thomas and Anderson, 2012). The study in Thomas and Anderson (2012) is most similar to the work presented here as they empirically compare several techniques of WSD for LS.

6.2 Substitution Generation

The work of Kilgarriff and Yallop (2000) defines four distinct types of thesauri. Firstly, we have manual book-style thesauri which are intended to help an author find a different word. Secondly, we have lexical databases, which encode semantic relations between lexemes. The third type is information retrieval thesauri, which are created

Table 6.1: The percentage of words from SUBTLEX with a simpler synonym in WordNet. The result does not improve greatly when the automated thesaurus, learnt from a large corpus, is added in.

Thesaurus	Coverage
WordNet	23.73%
WordNet + Automated	24.19%

for the purpose of using the correct terminology in a specific domain and to facilitate document retrieval. Lastly, automated thesauri are learnt directly from corpora by looking at words with similar contexts. In this section, we will first look at automated thesauri and show why they are insufficient for simplification. We will then look at the possibility of combining the remaining thesaurus types to improve results.

To apply substitution generation to LS, we must consider the LC of thesaurus entries. If a substitution is found, but it is more difficult to understand than the original term, then it will not be useful for simplification. It is impossible for every word to have a simpler synonym — as at least one word must be simpler than any other. In order to be useful for simplification, a thesaurus should exhibit a low proportion of words with no simpler synonyms and those words with no simpler synonyms should be easy to understand. We use the SUBTLEX frequencies (Brysbaert and New, 2009) as an approximation of LC in this chapter.

WordNet is an extensively used resource in computational linguistics. It may be used as a thesaurus if only the synonym relations are considered. We are interested in how many words from SUBTLEX have a simpler synonym in WordNet and so use the following definition of coverage:

> **Coverage:** The percentage of words in SUBTLEX which have at least one simpler synonym in WordNet.

We can see from Table 6.1 that the coverage of WordNet is low. A post-analysis of the non-included words indicated that low coverage was present at both high frequencies and low frequencies in the lexicon. At high frequencies, the low coverage was the result of function words and words which could not be further simplified. At lower frequencies, the low coverage was a case of WordNet either containing no entry for the given word, or containing a limited entry which provided one or two synonyms which were no easier to understand than the original term. Only 23.73% of words in

SUBTLEX can be simplified using WordNet. The low rate of simplification is certainly problematic and must be mitigated. Although we do not expect to be able to get to 100% coverage, we do expect that it would be possible to improve on this figure.

We call this discrepancy the *thesaurus-lexicon gap*. It arises due to the finite capacity of a thesaurus' lexicon. A cut-off must be placed somewhere and rare words are typically omitted as they are less likely to be required by the thesaurus' user base.

To address the thesaurus-lexicon gap, we first turned our attention to distributional semantics (Clark, 2012) as previous literature has suggested that this technique may be a viable option (Curran, 2004; Agirre and Soroa, 2009). This approach corresponds to the automatic thesauri identified in Kilgarriff and Yallop (2000). We built context vectors using the full text of English Wikipedia (database download dated 14th January 2013). We stripped all metadata from the articles, reducing the file to 18GB of raw text. A window size of 5 words (2 either side of the target word) was employed. We did not perform lemmatisation, but we did use log-smoothing to lessen the impact of some common bigrams.

We used the cosine metric, as calculated by the dot product of two vectors, to compute word similarity. A threshold was established from labelled thesaurus data (taken from WordNet relations) and used to classify relations for words which were not covered. Word pairs with a similarity above a threshold were included in a new thesaurus. The effects can be seen back in Table 6.1. Previously, we calculated the coverage of the corpus, which is shown in the first row. The second row shows the improvement attained by adding in the relations learnt with distributional semantics. Only a minimal improvement was attained. Many context vectors were small as they were learnt from only a few examples in the training corpus. Vectors with a magnitude of less than 50,000 were discarded as we empirically found these vectors to lead to false positives. As with any statistical language processing technique, the most information is available for the more frequent words. However, to increase coverage, we need information for the rarer words in the lexicon. It is very difficult to get enough information for the rarer words and so automated thesauri are not appropriate at this stage. As this avenue of research turned out to be of little use, we decided to address the original research task with manually curated resources.

6.2.1 Thesauri

In order to overcome the thesaurus-lexicon gap we will now analyse several thesauri and demonstrate how they can be used individually and in combination. Different thesauri will contain different relations and capture different aspects of a lexicon. We look at four general thesauri and nine specialised thesauri as detailed below. Following the classification scheme previously mentioned (Kilgarriff and Yallop, 2000), the first two general thesauri correspond to the lexical database class and the second two correspond to the book-style class. The special thesauri all correspond to the Information Retrieval class.

The general thesauri are as follows:

WordNet: The most widely used lexical database. It contains many lexical relations and groups lexemes into semantically related synsets. WordNet can be adapted for use as a thesaurus by exploiting these synset relations.

BabelNet: Styled as a multilingual WordNet, BabelNet (Navigli and Ponzetto, 2012b) incorporates data from several sources and languages to give a database with a similar structure as that described for WordNet, but with a much wider range of vocabulary. Only the English portion of BabelNet was used for this experiment.

Roget's: The original English thesaurus, with later revisions. The full text of Roget's thesaurus is available via Project Gutenberg (Hart, 1992). As this source is quite old, some of the terminology may be out of date.

Moby: Part of the Moby Project (Ward, 1996). A very extensive thesaurus. Also available via Project Gutenberg. The Moby Thesaurus is provided as a raw text file with each line containing comma separated terms and phrases which are semantically similar.

Everything: As each thesaurus has a different focus, it seemed prudent to assess the effect of combining all the thesauri described above. The special thesaurus, described below, was also added in. In the worst case the combined 'Everything' thesaurus would be no worse than the best performing thesaurus.

The specialised thesauri are taken from private and public bodies which provide the data for public use. Each thesaurus defines a standard of vocabulary to be used in a specific sector. These thesauri capture specific domain information, which we expect to be lacking from the general thesauri.

ICPSR: Interuniversity Consortium for Political and Social Research. Defining vocabulary for social sciences.

NAL: National Agricultural Library. Lots of terms to do with the agriculture industry. This thesaurus is the largest specialised thesaurus.

DTIC: Defense Technical Information Center. Information on concepts relevant to the defence sector. Also contains terms relevant to engineering.

NASA: National Aeronautics and Space Administration. Astronomical terminology as well as engineering terms.

ERIC: Education Resources Information Center. Terminology for education.

FISH: Forum on Information Standards in Heritage. Terminology covering several areas which are important to the heritage sector.

IPSV: Integrated Public Sector Vocabulary. A thesaurus containing terms related to local governance.

VOCED: Vocational Education Thesaurus. Similar to ERIC, but with a focus on vocational education.

TRT: Transportation Research Thesaurus. Terminology related to the transport sector.

We combined the specialised thesauri into one set of relations, termed the 'special thesaurus'. We combined each general thesaurus with the special thesaurus to see if the coverage would improve.

To compare the thesauri, we calculated the following statistics:

Lexicon Size: The number of unique words contained in the thesaurus.

Number of Edges: The amount of relations between two words in the thesaurus. Note that we count edges as unidirectional and so a bidirectional edge is counted twice.

Coverage: As previously calculated. Now applicable to any given thesaurus.

Incremental Coverage: We calculated which words from SUBTLEX had either no substitutions or no simpler substitutions. This information is presented as a graph of the coverage as lexicon size increases in Figure 6.1.

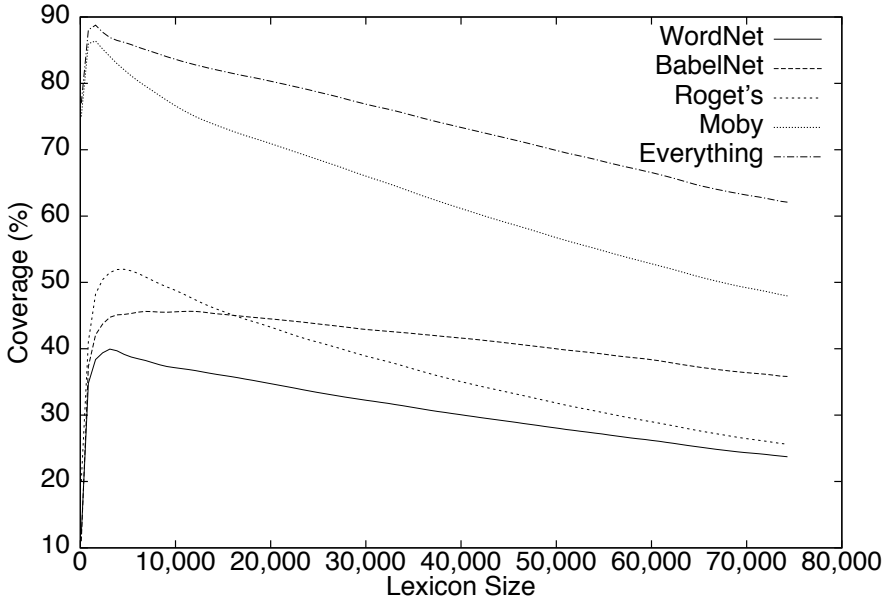


Figure 6.1: This graph was produced by incrementing the lexicon size in order of word frequency and calculating the coverage at each stage. We see that all thesauri have a higher coverage on the left side of the graph.

Table 6.2 presents the statistics calculated for the various thesauri. We have included the special thesaurus on its own for comparison. The incremental coverage curves are shown in Figure 6.1. These results are presented for all the non-specialised thesauri. In Figure 6.2 we show the effects of adding in the specialised thesaurus to both WordNet and Moby.

6.2.2 Discussion

It is clear from Figure 6.1 that the most comprehensive single thesaurus is Moby. Adding in the synonyms from the other thesauri, as well as the special relations, did improve coverage and help to reduce the thesaurus-lexicon gap. All the thesauri in Figure 6.1 exhibit a similar curve which has a sharply increasing coverage over the first two thousand words and reaches a peak at around two thousand words. It then gradually declines over the remaining seventy thousand words. The initial low coverage is because some portion of the high frequency words cannot possibly have any simpler synonyms. The shallow decline that ensues is a result of an increasing likelihood that a word will not be covered by the thesaurus as its frequency increases. This decline takes a different gradient for each thesaurus and so we observe some crossover. This effect would be amplified with a larger lexicon. We can extrapolate these curves forward to

Thesaurus	Lexicon Size	No. Edges	Edges / Word	Coverage
WordNet	44,759	140,150	3.131	23.73%
BabelNet	303,844	1,460,146	4.806	35.82%
Roget's	32,032	308,996	9.646	25.62%
Moby	61,672	110,964,328	1,799.266	47.97%
Special	38,654	95,108	2.460	7.00%
WordNet + Special	79,048	233,934	2.959	29.03%
BabelNet + Special	329,068	1,550,684	4.712	39.47%
Roget's + Special	67,579	403,076	5.965	30.53%
Moby + Special	95,084	111,051,884	1,167.935	51.21%
Everything	355,716	112,512,424	316.298	62.08%

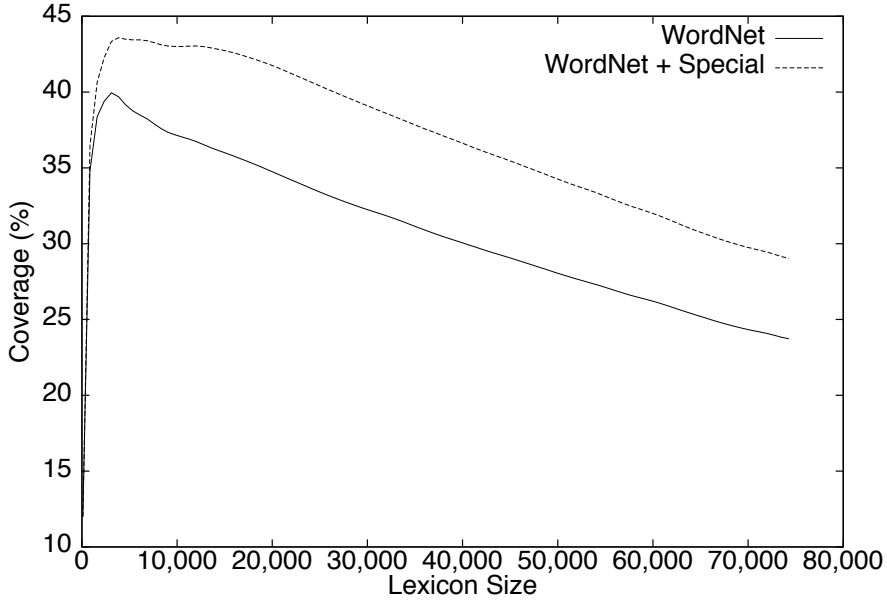
Table 6.2: A table showing statistics for each thesaurus.

see the effect of a larger lexicon. As lexicon size increases, coverage decreases. This finding implies that lexicon size is an important factor in building a substitution generation system. There is an important trade off to be made here, as difficult words are likely to occur at lower frequencies and so the lexicon must be of a sufficient size to capture these words.

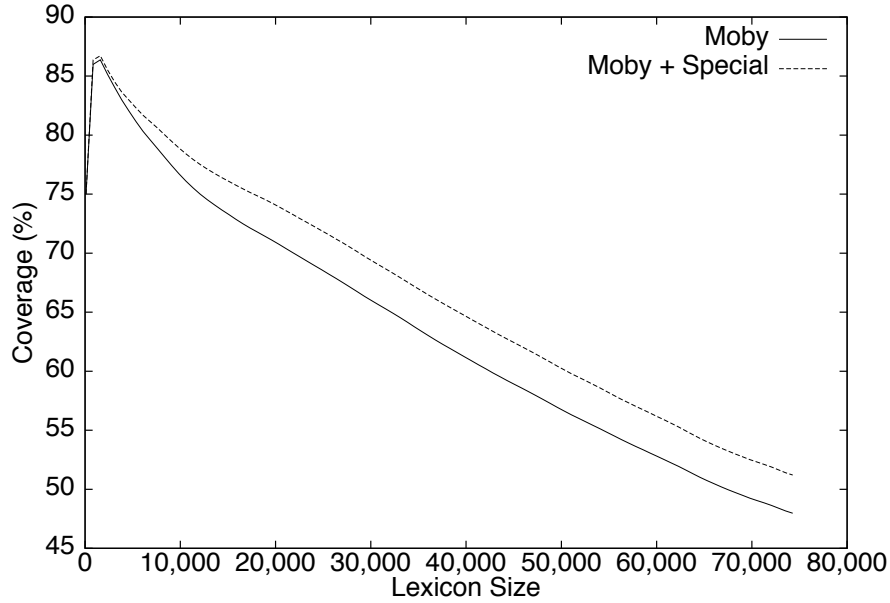
WordNet has the largest thesaurus-lexicon gap of all the thesauri that we employed. It is still used widely as an electronic thesaurus, and this research shows that it should be used with caution, especially for simplification.

There is a contrast between the curves of Roget's and BabelNet in Figure 6.1. Roget's peaks much higher than BabelNet, but then also declines much more sharply. This finding implies that Roget's has a much higher coverage of common words, whereas BabelNet has a better coverage of rare words in the lexicon. BabelNet has the least steep gradient of any thesaurus, which implies that it is the most consistent at all levels of difficulty. The language of the available version of Roget's may be outdated, explaining the lower coverage at higher difficulties.

The Everything thesaurus uses data from all the other thesauri and is able to beat its closest competitor, the Moby thesaurus. Table 6.2 shows that an increase in coverage of 0.141 is observed over Moby, resulting in a final coverage of 0.621. The combined thesaurus has 294,044 more words than Moby, out of a possible 419,289 which could have been added from the other thesauri. Similarly, the combined thesaurus has 1,548,096 more edges than Moby, out of a possible 2,004,400 edges which could have been added from the other thesauri. By combining more thesauri, the coverage could be further improved. All the thesauri used in this work are in the public domain and



(a) WordNet vs. WordNet + Special.



(b) Moby vs. Moby + Special.

Figure 6.2: The effect of adding in the special thesaurus to WordNet and Moby.

may be used for research purposes by anyone who seeks to do so.

We see in Table 6.2 that although Moby has the largest number of relations by a long distance, it does not have the largest lexicon size. Instead, BabelNet has the most words in its lexicon. Moby has a very high number of edges per word, much higher

than any other thesaurus. This high interconnectivity within the Moby Thesaurus explains why it performed so well in our experiments. With so many relations for each word, there was a high likelihood at each point of at least one word being easier to understand than the original.

A factor which has not been considered in this work is the concept of thesaurus tightness. That is, how closely related are the words associated to an entry. We can infer that as tightness decreases, more words will be linked to each entry. The Moby thesaurus has an unusually high number of edges per word, as shown in Table 6.2. It is entirely possible that this increased number of edges is due to a lower tightness in this particular thesaurus. The quality of the simplifications may be affected, as a substitution must convey the same concept as the word it is replacing. When using the Moby thesaurus for LS, it will be important to ensure that the returned words still convey the author's original intention. With so many substitutions to choose from, the likelihood of finding a valid candidate is increased.

Each of the thesauri would be larger if we had chosen to include MWEs, however coverage would not increase as the SUBTLEX frequencies only contain single words. To incorporate MWEs we would need to use a measure of LC as proposed in Chapter 5. As no such measure was available at the time of this research, we were unable to assess the effect of incorporating MWEs.

The effect of the special thesaurus can be seen in Figure 6.2. We show the effect on WordNet and Moby in Figures 6.2a and 6.2b respectively. The special thesaurus clearly yields an improvement in both examples, whereas for WordNet the increase happens around the peak of the graph. For Moby the increase continues after the peak.

The coverage of the special thesaurus itself is 7.00% and therefore the total possible improvement from using the special thesaurus is by 7 percentage points. We can observe from Table 6.2 that as the coverage of the thesaurus increases, the improvement gained from the special thesaurus reduces. For WordNet, we see an improvement of 5.3 percentage points, whereas for Moby we see an improvement of 3.24 percentage points. This result is expected as the larger the thesaurus, the more specialised terms it will already cover.

The special thesaurus is heavily influenced by the source domains of the base thesauri. In this work, we included nine specialised thesauri to make up the special thesaurus, although more could have been included. The improvement gained from the

special thesaurus shows that these types of thesauri do indeed include the type of vocabulary which is lacking in the general thesauri. The coverage of 7.00% could certainly be improved by adding in more specialised thesauri. If a simplification system is aimed at a specific domain then the results presented here suggest that using a thesaurus with specific terminology for that domain as well as a general thesaurus will yield an improvement against only using a single resource. Such resources may not always be available when working with specific domains, especially when working with languages other than English. If a specialised thesaurus is not available then it may be sensible to create such a resource before attempting LS.

6.3 Word Sense Disambiguation

The WSD literature provides many suitable evaluation resources. Notably, the SensEval (Snyder and Palmer, 2004) and SemEval (Navigli et al., 2007) conferences provide numerous shared tasks for evaluation. These conferences have been highly successful in uniting the WSD community and focusing their efforts around specific tasks. They have also been useful in determining state-of-the-art methods for WSD. The most relevant style of task for LS is the 'all words' task, which asks a system to assign sense keys to all the words in a given corpus. These assignments can then be evaluated against gold standard data. One resource which is heavily used throughout WSD is WordNet. Many of the following techniques use WordNet in differing ways.

Below, we describe a selection of WSD methods which may be used for LS. BabelNet, SenseRelate and SenseLearner represent techniques which have been taken from the WSD literature. Context Vectors and Personalised PageRank represent sections which have been taken from the LS literature. Language Modelling represents a system which was built for this research.

6.3.1 Techniques

BabelNet

BabelNet is a large multilingual semantic network. Concepts in BabelNet are organised into BabelSynsets which are related to each other by various linguistic relations. BabelNet is freely available online and may be used for WSD by looking for the sense of the target word which is most similar (by some measure of similarity) to the context words. In our experiments we used the method described in Navigli and Ponzetto

(2012a) for WSD.

SenseRelate

SenseRelate (Pedersen and Kolhatkar, 2009) is a WordNet based disambiguation tool. It uses a series of WordNet semantic similarity tools for disambiguation. Senses from WordNet are assigned to each word based on the semantic similarity of the target word's sense to the context. The 'all words' adaptation of SenseRelate which we used for our evaluation uses the Lesk disambiguation algorithm (Lesk, 1986). Lesk looks at the best match between a WordNet sense's gloss and the context sentence. It strongly depends upon the accuracy of the glosses and the content of the target sentence.

SenseLearner

SenseLearner (Mihalcea and Csomai, 2005) is the second best performing system from the SensEval-3 all words task (see Section 6.3.2). We initially attempted to implement the best performing system, GAMBL-AW (Decadt et al., 2004), however the system was not available online. SenseLearner trains general models for categories of words. The categories can be of mixed granularity and may focus on syntactic and semantic properties. For example, a category may be all nouns or all words relating to 'move'. Feature vectors are created for each word in a semantic category and the TiMBL package (Daelemans et al., 2009) is used for learning.

Context Vectors: Synonym Selection

Context vectors are the result of recording the common co-occurrents of a word in a large corpus (Clark, 2012). Context vectors can be used for WSD by assessing the similarity of two word-vectors. We present two distinct applications of context vectors.

We followed the method described in Biran et al. (2011), each substitution is evaluated to determine how likely it is to fit in the given context. A common context vector is created from the target and substitution words by taking the minimum of any common dimensions. The cosine similarity between the common vector and a ten word token around the target word is calculated and the substitution is accepted only if the similarity is above 0.01. This low threshold effectively permits most synonyms which have some common context with the target word.

The context vectors were created using English Wikipedia as a corpus. A vector was created for each word with the co-occurrents in a 10 token window (5 words either

side of the target). After processing, any dimensions with a frequency less than 2 were discarded. The text was lemmatised before processing to conflate morphological variants.

Context Vectors: Sense Selection

We also present a second context vector method. This method is described by Bott et al. (2012) as part of a LS system for the Spanish language Simplext project. The context vectors were very similar to those of Biran et al. (2011). Two important differences are that no discarding was performed, and also a much smaller corpus (8 million tokens) was used. For our purposes, we use the English Wikipedia corpus for the former context vectors. Although we do not discard any synonyms for this method. The corpus used in the Simplext project was in Spanish so it was necessary to use an English resource for this research.

Whereas the context vector method in the previous section uses context vectors to select appropriate synonyms, this method selects the most appropriate sense of a word. Given a set of sense-separated data for a wordform, such as WordNet synsets or thesaurus categories, the context vectors for the words of each sense are combined by addition to give sense vectors. The cosine distance is calculated between each sense and a window of 8 tokens (4 either side) around the target word. The narrow context of 8 tokens aims to ensure that the most relevant information is captured. The sense with the lowest cosine distance is selected.

Personalised PageRank

The UKB WSD system (Agirre and Soroa, 2009) has previously been used for WSD in the LS pipeline by Thomas and Anderson (2012). It uses personalised PageRank (Brin and Page, 1998) to determine the strength of links between synsets in the WordNet graph structure. Although PageRank is typically associated with information retrieval, it has been shown to be effective for graph-based WSD as well. Personalised PageRank extends PageRank by allowing biasses to be introduced to the initial weighting. In this case, the weightings are created using the tag-count feature which is associated with each WordNet synset. The tag count is used as an approximation of the frequency of a given synset.

Language Modelling

Using the Google Web 1T frequencies (Brants and Franz, 2006), we built a large language model (Brants et al., 2007) which was capable of assigning a likelihood score to a sentence. The language model allowed us to see how likely a sentence would be to contain a given word. In preliminary experiments we also looked at the use of the Latent Words Language Model (Deschacht et al., 2012), which has been previously applied for WSD in LS (De Belder et al., 2010). During implementation, we found this model difficult to train with sufficiently large corpora. The trained models did not produce intelligible results, consistently over-producing synonyms for a given target word.

6.3.2 Evaluation Methods

We describe three evaluation methods for WSD. We refer to these as A, B and C for clarity. Each evaluation method captures a different aspect of the WSD problem as described below. Adaptations to each WSD technique were made where necessary. These adaptations are described below.

Method A

The classic formulation of WSD is as follows:

> A word w exists in a sentence S. Assign the most likely sense to w in S from the potential senses $s_1, s_2, \ldots s_i$.

This description focuses heavily on choosing the correct sense of a word from a lexical knowledge base such as WordNet. Once the sense has been selected, an LS system may use synonymy relations to select replacements which will not change the meaning of the final sentence.

To evaluate the success of this method, we use data from the SensEval-3 all words task (Snyder and Palmer, 2004). The task involves assigning senses to a wide variety of words across a set of documents. The only information available to a system is the context of the word. The system must select one sense from WordNet. A gold standard data set is also provided, in which the correct sense for each word is given. Several annotators were used to attain a consensus of the correct sense.

The most recent 'all words' data comes from SensEval-3, a predecessor of the SemEval workshops. We chose to use the all-words data as it provides a more general

coverage than the specific lexical sample tasks. The data is available online, along with a scoring metric, which was used for evaluation.

The baselines, along with any specific adaptations for each technique are as follows:

Random Baseline: A random sense was selected at each point.

Maximal Sense Baseline: The most frequent sense according to WordNet was selected.

Language Model: We calculated the language model score for each synonym in each candidate synset. The mean score for each synset was calculated and the highest scoring synset was selected.

BabelNet, SenseRelate, SenseLearner, Personalised PageRank: These methods are designed to select specific senses and required no significant adaptation.

Context Vectors - Sense Selection: As there are two contrasting context vector methods, we only apply the sense selection method here and we only apply the synonym selection method to the next two evaluation measures. If we tried to transform the synonym selection method to work with this evaluation method we would likely get something very similar to the sense selection method, and likewise in reverse.

Method B

In Method B, we look at the WSD task from a different perspective. Method A required a specific sense to be assigned to each word in a text — a classic formulation of the WSD task. However, for LS it has two disadvantages. Firstly, the sense inventory may not cover the vocabulary necessary for simplification. Rare words are often the focus of simplification and unfortunately they tend to be sparsely represented in resources. Secondly, the construction of a resource must make distinctions between senses at a given level of granularity. The usage of a word may not reflect any of the senses in the resource or may be used at a different level of granularity. Another approach towards WSD for LS may be to focus on selection at the individual synonym level. This approach can be described formally as follows:

> A word w in the sentence S has the potential replacements: $r_1, r_2, \ldots r_i$.
> Which replacements retain the same meaning in S as w?

We modified the LS data from SemEval-2012 task 1 (Specia et al., 2012) to create a new evaluation method. This data is in the form of sentences with one word annotated with a set of good-fit synonyms. For our purposes we assume that the synonyms associated with a word are all equally good-fits in the context. There are several example sentences for each wordform, each with a subtly different sense of the word and with different associated synonyms. The synonyms associated with each word were assumed to be a gold standard set. To create a suitable test set, we conflated all the synonyms associated with each word into one long list of synonyms. The task was to pick which synonyms from this large list were appropriate in the context of each sentence. To assign each system a score, we used the Jaccard index to measure the similarity between the selected set of synonyms S_s and the gold standard set of synonyms S_g:

$$J(S_s, S_g) = \frac{|S_s \cap S_g|}{|S_s \cup S_g|}$$

The baselines, along with any specific adaptations for each technique, are as follows:

Random Baseline: A random subset of the synonyms was selected. Each synonym had a one in three chance of inclusion.

Open Door Baseline: The Open Door baseline chose every synonym for inclusion. This technique simulates a system with high recall, but low precision.

Language Model: We perform clustering on the scores using Otsu's Method (Otsu, 1975) to separate a class of synonyms for inclusion and a class for discarding. Several clustering methods were explored, however Otsu's method was the most effective in this case.

BabelNet, SenseRelate, SenseLearner, Personalised PageRank: These methods all return a synset. Only the synonyms which are part of the synset were selected.

Context Vectors — Synonym Selection: The synonyms were evaluated as described previously. The synonyms which were selected by the method formed the final set for evaluation.

Method C

To assess the effect that the disambiguation measures had on the simplification process, we also used the raw data from SemEval-2012 task 1 for evaluation. This data consists

of sentences, each with a valid set of synonyms, and asks systems to rank the synonyms in order of simplicity. A scoring mechanism is provided which we used for evaluation. To rank the synonyms, we used the following formulation of LS:

> A word w in the sentence S has the potential replacements: $r_1, r_2, \ldots r_i$.
> Rank $r_1, \ldots r_i$ in order of semantic distance to w in S.

This formulation assumes that words which are semantically close to the original term will be simpler. Some of the WSD techniques were difficult to adapt to this measure and so in these cases we used the Google Web 1T frequencies to rank the synonyms, but gave an automatic rank of last for any which were considered out of context by the evaluation measure. For those methods which did provide a semantic distance, we ranked the synonyms by their distance to the original word.

The baselines, along with any specific adaptations for each technique, are as follows:

Random Baseline: This baseline is provided as part of the SemEval-2012 task 1 data. It ranks the synonyms in a random order.

Simple Frequency Baseline: This baseline is also provided as part of the SemEval-2012 task 1 data. It ranks synonyms according to their frequency in the Google Web1T data.

Language Model: The synonyms are ranked according to the score assigned by the language model.

BabelNet, SenseRelate, SenseLearner, Personalised PageRank: All the synonyms which were not selected by the given method are automatically assigned a rank of joint last. The rest are ranked according to the Google Web1T frequencies.

Context Vectors Synonym Selection: The synonyms were ranked by their similarity to the context.

6.3.3 Results

For readability, the following acronyms are used in all following tables and figures:

RB: Random Baseline

MS: Maximal Sense Baseline

OD: Open Door

SF: Simple Frequency

LM: Language Model

BN: BabelNet

SR: SenseRelate

SL: SenseLearner

PPR: Personalised PageRank

CV1: Context Vectors — Sense Selection

CV2: Context Vectors — Synonym Selection

For evaluation method A, Table 6.3 presents the three metrics: precision, recall and percent attempted. We use the same definitions of precision and recall as in the WSD literature. Note that the WSD definitions are different to the traditional metrics of precision and recall as used in the typical information retrieval literature. The WSD versions of precision and recall may be defined as follows:

Precision$_{WSD}$: the number of instances that the system attempted which were correctly classified.

Recall$_{WSD}$: The number of instances that were correctly classified, counting unattempted instances as false.

For example, if the system is asked to classify one hundred instances but only attempts eighty, and if forty of the eighty instances were correctly classified, then the recall would be $\frac{40}{100} = 0.4$ whereas the precision would be $\frac{40}{80} = 0.5$.

The percentage attempted shows the number of instances which the system attempted to classify. Reasons for non-classification occur due to errors in external resources (such as part-of-speech tagging and lemmatisation) or the word not being found in WordNet. The precision and recall are also shown in Figure 6.3.

For evaluation method B, we calculated the Jaccard index for every data instance and then calculated the mean and standard deviation across the entire data set for each technique. We found that this method led to wide ranging distributions. We used a t-test to investigate statistical significance between methods. The results of the significance tests can be found in Table 6.4 and results of the experiment can be found in Table 6.5 and Figure 6.4.

Table 6.3: The results of Method A (SensEval-3 Data).

Method	$\text{Precision}_{\text{WSD}}$	$\text{Recall}_{\text{WSD}}$	Attempted
RB	0.271	0.268	98.87%
MS	0.577	0.570	98.87%
LM	0.387	0.382	98.87%
BN	0.438	0.374	85.45%
SR	0.458	0.443	96.86%
SL	0.639	0.576	90.15%
PPR	0.431	0.425	98.48%
CV1	0.29	0.286	98.87%

Table 6.4: The significance of the results from Method B (LS Conflation). Three levels of significance are reported. Values which were not significant ($p \geq 0.05$) are reported as the raw p value. Values which were significant ($p < 0.05$) are reported as '$\star$'. Values which were highly significant ($p < 0.001$) are reported as '$\diamond$'.

	CV	PPR	SL	SR	BN	LM	OD
RB	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$	$\diamond$
OD	$\diamond$	0.0974	$\star$	0.3483	$\star$	0.6324	
LM	$\diamond$	$\star$	0.1143	0.5952	$\diamond$		
BN	0.1661	0.224	$\diamond$	$\diamond$			
SR	$\diamond$	$\star$	0.3535				
SL	$\diamond$	$\star$					
PPR	$\star$						

Table 6.5: The results of Method B (LS Conflation).

Method	Mean Jaccard Index	STD
RB	0.17720	0.14186
OD	0.28023	0.13163
LM	0.28240	0.13362
BN	0.26339	0.19272
SR	0.28527	0.17899
SL	0.29097	0.18021
PPR	0.27120	0.18271
CV2	0.25497	0.16142

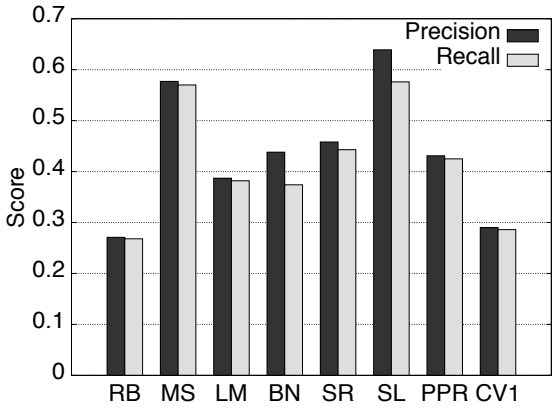


Figure 6.3: A bar chart representing the data from Table 6.3 (Method A). The discrepancy between precision and recall is a result of systems which did not attempt to classify every instance.

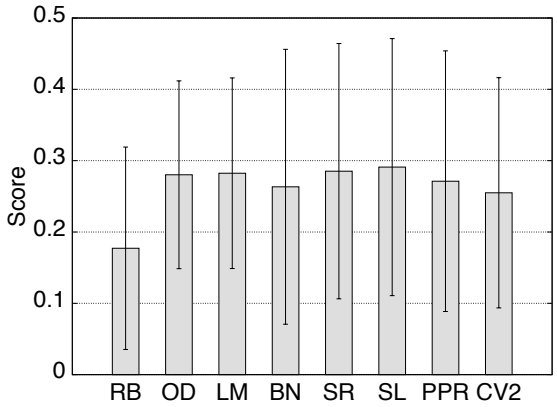


Figure 6.4: A bar chart representing the data from Table 6.5 (Method B). The wide error bars are a result of the high standard deviations. Significance values are reported in Table 6.4.

For evaluation method C, we present the results of the analysis in Figure 6.5. These results were produced using the gold standard data and scoring metric provided as a part of the SemEval data. Inter-annotator agreement is calculated between the rankings created by each system and the rankings in the gold standard. The latter was produced using a crowdsourcing system to ask many annotators to rank sets of synonyms. The resultant rankings were created by combining the decisions of the human judges. The agreement is calculated using Cohen's kappa. It should be noted that the value given by this metric is highly dependent on factors such as the number of annotators and the number of judges. Thus, the values are best interpreted on a case-by-case basis. Particularly, this method is useful for determining whether one method returns a higher agreement than another.

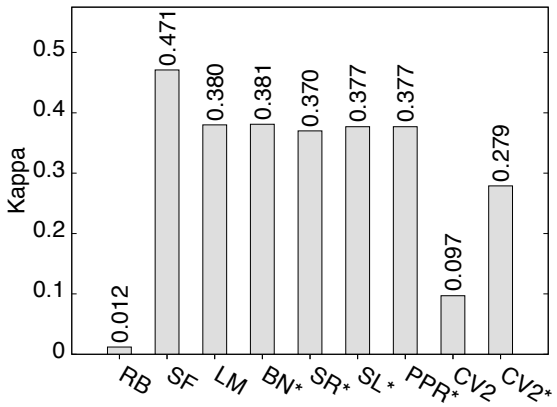


Figure 6.5: A bar chart representing the results of Method C (SemEval-2012 Task 1 data). The methods which were combined with the Web 1T rankings are denoted by an asterisk (*).

6.3.4 Discussion

WSD is not easy, as clearly shown from the results of this study. Whilst the random baselines are easily beaten by all techniques, very few experiments were able to beat the common sense baselines (Maximal Sense / Open Door / Simple Frequency). These baselines are very simple and, intuitively, should be easy to overcome. However, as shown from these results, they perform well. Certain techniques perform consistently across the three evaluation methods. SenseLearner does well on methods A and B, equalling or beating all others. By contrast, the context vector techniques yield a consistently poor performance. The reduced score may be a factor of our context vectors, but it does show that context vectors can be difficult to tune and apply in a system. Other techniques have a more varied set of results. For example, SenseRelate performs well on Method C, yet has the second lowest recall for Method B.

BabelNet, SenseLearner and SenseRelate all have reduced attempted rates. Apparently, some factor of these techniques makes them unable to classify certain words with a WordNet sensekey. On further inspection, it was found that BabelNet did not always return a concept which could be translated into a WordNet synset. Babelnet has a much wider semantic network than that covered by WordNet. SenseLearner must be trained on a large amount of data and it would appear that the models which were provided did not cover all the words in the test data.

The best system for Method A is SenseLearner. To some degree, this result was expected as SenseLearner was developed specifically for the SensEval-3 task. None of the other methods was able to outperform the maximal sense baseline or even approach

the same score. SenseLearner is clearly very good at this task. The context vector and language model methods are not particularly well suited to this task. Accordingly, the CV method is only just better than the random baseline. The LM method gives an improvement, yet still gives a low score.

Figure 6.4 must be viewed in light of the significance values in Table 6.4 to be properly understood. It can be seen in Figure 6.4 that the evaluation technique yields very high standard deviations. Table 6.4 demonstrates that, despite the high variation in each data point, there are still some important significances to be explained. The high standard deviations indicate that there is a large variability in difficulty between the different data instances. To further improve the significance values in Table 6.4, data of a more consistent difficulty level would be required.

All techniques for Method B significantly outperformed the random baseline. This result indicates that each technique was suited to the task, although to varying degrees. Although 3 techniques (the language model, SenseRelate and SenseLearner) were able to outperform the Open Door baseline, only SenseLearner did so significantly. This result is surprising as the Open Door baseline is a very naïve method. Similar to the other techniques, this finding serves to show the difficulty of the task. SenseLearner yields the best score for this evaluation measure, however it is shown to not be significantly better than either SenseRelate or the language model. The language model is well suited to this task as it can make a decision about each individual synonym. SenseRelate and SenseLearner are designed to operate at the synonym level which was initially thought to put them at a disadvantage. However this was clearly not the case as they performed well in this evaluation. It is interesting to note the low performance of the CV technique. The CV technique was designed to give a high score to words which are in context and so was expected to perform well in this task. This expectation was not met and the CV method scored significantly lower than the Open Door baseline.

For evaluation method C, we observe that all techniques outperform the random baseline, but do not beat the simple frequency baseline. This result is largely in line with the results of the original shared task from which this data was taken (Specia et al., 2012) where only 1 system (Jauhar and Specia, 2012) beat the simple frequency baseline. In our experiment, BabelNet performed the best but was closely followed by the language model, SenseLearner, Personalised PageRank and SenseRelate. The benefit of interpolating the systems with the Web1T data can be seen in the results of the context vector method. Interpolation with the Web1T data results in a dramatic increase in score. The context vectors performed particularly worse than the other

methods, even when combined with the Web1T data. This result shows that they are not well suited to disambiguation for this task.

This data is specifically designed to assess how well a system can replicate a human's performance in the task of ranking words in order of simplicity. It is interesting to note that adding a WSD component to the simple frequency baseline causes the score to decrease. More sophisticated methods of combining the systems may eventually help to improve the final score.

6.4 Recommendations

There are two important characteristics that we should strive for when generating substitutions for LS. The substitutions should both preserve meaning and be easier to understand than the original word. However, there would appear to be a trade off between these two characteristics. The more substitutions that are produced, the lower the proportion of meaning-preserving substitutions. The tighter the meaning preservation, the fewer the simpler substitutes. This tension is difficult to resolve.

We have presented two pieces of research in this area. The work on substitution generation shows that more simpler synonyms can be gained by increasing the number of thesauri in use, especially by using specialised technical thesauri. We would expect all of these substitutions to be correct in a given context, however they will not all be applicable in every context. It is clear that we must use these thesauri with a WSD technique. Our experiments comparing WSD for LS have shown mixed results. WSD is not a solved task and improvements are regularly made in the field. Although SenseLearner (amongst other techniques) performed well, it was not at a strong enough level to produce error-free text.

Regarding the two approaches to the combined task of substitution generation and WSD, we are left with the following two options. Should we perform substitution generation first and rely on the WSD method to select substitutions of the correct sense? Or should we perform WSD first and rely on the lexical database to return substitutions which retain the meaning of the original word? We have seen that BabelNet and WordNet both have a low coverage. If we perform WSD to select a sense from WordNet or BabelNet, we would only expect a coverage of 23.73% or 35.82% respectively. Conversely, if we perform substitution generation first, then we must rely on our WSD technique to remove the out of context synonymy. We saw from Method B that there is a large amount of noise in this task and that no single method which we evaluated

gave a strong performance in this task.

Going forward, LS research must be mindful of the limitations of thesauri and WSD resources. The resources must be carefully selected for each task and may require human intervention to ensure accuracy. As these resources are improved further there will be a direct benefit to LS. The dream of a fully-automated error-free LS system is clearly some way off. Until that point we must continue to improve resources, mitigate errors and intervene with human annotators where necessary.

Chapter 7

User Study

In this final chapter of results, we will use the pipeline which we have previously discussed in a preliminary user study conducted with sufferers of aphasia. Previous chapters have focused on errors and improvements in the LS pipeline. This chapter will use the pipeline to simplify documents for users. We are specifically interested whether the type of simplifications produced by LS improve reading ability for people with aphasia.

Aphasia is a neurological condition which can occur due to brain damage after a stroke (Pedersen et al., 1995). Sufferers experience a reduction in language comprehension and production, both verbally and written. Aphasia is a spectrum disorder and severity of impairment in each affected area varies widely with each sufferer (Carroll, 1986). People with aphasia often also experience some physical impairments as well as language impairment, which complicates their stroke recovery. With appropriate therapy and sufficient time, impairments that result from a stroke can recover, although they will often not return to the levels experienced before a stroke (Carlomagno et al., 2001; Kertesz, 1984).

We are interested in using LS as a tool for stroke recovery. After a stroke, it is a significant shock for a patient to discover that they can no longer interact with the world around them as before. Appropriate information sources are limited, with aphasia sufferers reporting several types of information need which are not met (McKevitt et al., 2010). Aphasia-friendly information sources are expensive to produce and keep up to date. Aphasia-friendly guidelines exist which propose strategies such as clear text, short sentences and one message per point (Howe et al., 2004; The Stroke Association, 2012). As a long term goal, we would like to build a tool which is capable of automatically converting content written for a general audience into aphasia-friendly content. As an initial step towards this aim, this study has focused on whether the kinds of simplifications produced in LS help people with aphasia to read and understand a document. This research is governed by the following research question:

> **RQ 7.1:** Does automated LS improve text comprehension for people with aphasia?

This RQ leads to the related research hypotheses:

> **RH 7.1:** People with aphasia will read documents with LS more quickly than equivalent documents without simplification.
>
> **RH 7.2:** People with aphasia will understand documents with LS better than equivalent documents without simplification.

RH 7.3: People with aphasia will judge documents with LS to be easier to understand than equivalent documents without simplification.

7.1 Related Work

We have previously mentioned the work of Carroll et al. (1998), in which simplification is applied as an assistive technology for people with aphasia. Although our study is inspired by theirs, the key difference is that we isolate lexical simplifications and only evaluate their effects upon a text, whereas Carroll et al. (1998) look at both syntactic and lexical simplifications together. In her thesis, Devlin (1999) records the results of another experiment which sought to apply simplifications for people with aphasia. In that experiment, the texts were manually simplified following a structured algorithm. Lexical simplifications were applied, as well as some syntactic simplifications to split the sentences into individual clauses. The results of this experiment showed an increase in the patient's understanding when simplification was applied to the documents. This work was continued in the HAPPI project (Devlin and Unthank, 2006), which sought to improve online interactions with text for people with aphasia. No results were published for the HAPPI project.

Other studies have also sought to isolate and evaluate lexical simplifications. Leroy et al. (2013b) evaluate the effect of lexical simplifications on the health literacy of lay readers, as we do here. The results of their experiment show that LS leads to a drop in understanding as measured by a Cloze test, although their users rated the simple texts as easier to understand. A later study by the same authors shows an overall positive effect for LS when it is incorporated in a writing aid tool for authors of health literature (Leroy et al., 2013a). Rello et al. (2013b) also isolate lexical simplifications and evaluate their effects on the reading abilities and understanding of people with dyslexia. They find that reading ability is improved by the substitution of frequent words and that understanding is improved by the substitution of shorter words.

7.2 Experimental Setup

To test our hypotheses, we presented each participant with two documents. One document had been simplified using an LS system and the other had not been simplified. We measured an array of statistics for each document and repeated the experiment several weeks later with the simplification of the documents reversed. This format allows

Table 7.1: The BNT scores for each participant.

Participant	BNT/60	BNT/100
1a	2	3
1b	25	42
1c	26	43
2a	27	45
2b	30	50
3a	32	53
3b	34	57
4a	37	62
4b	51	85

us to perform two comparisons. Firstly, we can compare the statistics for the complex and simple version of each document. Secondly, we can compare the statistics for the complex and simple documents presented at each visit.

7.2.1 Participants

We recruited participants who had previously taken part in research at the University of Manchester. Baseline assessments, including BNT scores, were available for all participants. All regularly take part in research activities. We visited each participant in their own home. The research required two visits each ranging between 25 and 35 minutes depending on the participant's ability.

The BNT score tells us how well a participant can name objects and gives an indication of the severity of their aphasia (Kaplan et al., 1983). The BNT/60 and BNT/100 scores reflect how many items, from a list of 60 and 100 respectively, a participant can name. We used the BNT scores to assign similar ability pairings to the participants. These pairings were used to counterbalance certain factors in our experiment as we will describe later. There is a wide range in the participants' BNT scores, as shown in Table 7.1.

7.2.2 Simplification System

We created a LS system using the pipeline described in this thesis, augmented with some improvements from the literature. We used frequencies from Simple Wikipedia

for identifying CWs. If a word occurred fewer than 20,000 times then it was considered to be complex. We used BabelNet to ground the instance via a graph based disambiguation method described in Navigli and Ponzetto (2012a). BabelNet was chosen as it performed similarly to other methods in our analysis in Chapter 6, yet the lexical database which backs the instances is much larger than WordNet. We used the BabelSynsets returned from the disambiguation method to generate a set of substitutions. These substitutions were ranked using the frequencies from Simple Wikipedia and the synonym with the highest frequency was selected.

We used a technique described in Biran et al. (2011) to preserve morphological inflections between the original word and the selected substitution. This technique required the use of MorphAdorner to generate all the possible inflections of a pair of words and then select the correct pair. Preserving the inflections was essential as the WSD step lemmatised its input, returning the lemma of the substitution. So for example, if the input sentence with the target word italicised was: 'The cat *perched* on the mat', then the system would search for synonyms of the lemma 'perch'. Pairs for the substitution 'sit' would include: ⟨perch, sit⟩, ⟨perched, sat⟩, ⟨perching, sitting⟩, ⟨perches, sits⟩. The system would select 'sat' as the correct replacement.

We manually assessed the output of the pipeline and felt that the number of errors would be too great to present to end users. The errors would mask the effect of the simplifications and conflict with our results. We hand reverted erroneous simplifications and allowed a human editor to suggest simplifications to the system when none could be found in WordNet. These extra suggestions were added to the system's lexical database and stored for later retrieval. Our approach here is similar to Leroy et al. (2013a), where errors are hand corrected to mitigate their effect.

7.2.3 Documents

We wanted to make the experiment as interesting as possible to the participants. We selected biographies from Wikipedia which we felt would be of general interest. We chose Wikipedia as it is a source of information which people with aphasia may struggle to interact with. Although Simple Wikipedia provides easier to understand articles, the style of writing is often still too complex for a low ability reader (Xu et al., 2015). Wikipedia has a self-regulated article quality measure. The top level is 'Featured Articles', which indicates that these are the highest level of quality for information accuracy and writing style. We selected from this group to ensure the quality of the article. We selected the entirety of the introduction section for each article. We examined 17

Table 7.2: Statistics about the two selected documents. The readability index gives an indication of the U.S. Grade level that this text would be appropriate for. The language model score gives an indication of how often the words in a sentence occur nearby each other.

Feature	Document 1	Document 2
Subject	Charlie Chaplin	Julianne Moore
Word Count	430	406
Number of Sentences	29	24
Words per Sentence	14.83	16.92
Characters per Sentence	5.578	5.601
SMOG Readability Index	10.166	10.864
Language model Score	-106.497	-106.314

articles and selected 2 based on their similarity of text length, sentence length and other factors. The final articles were a biography of Charlie Chaplin and a biography of Julianne Moore. Further information about the original version of these articles is given in Table 7.2.

The next task was to process the articles into an easy to read format. We processed the original documents into an aphasia-friendly format. We performed sentence splitting to ensure there was only one key point per sentence. We also performed anaphora resolution. We used the surname of the article's subject to replace the first anaphora in each sentence. We also deleted some information where it was not helpful. We formatted the document by spacing out each sentence, reducing the width of the text, increasing the font size to 12pt and enclosing the text in a thick bordered box.

Finally, the document was processed by the LS system described in the previous section. Altogether we had four documents: two simplified and two original. All four documents were processed into the aphasia-friendly format.

7.2.4 Assessment Methods

We measured the participants' interactions with each document in several different ways. Firstly, we measured how long it took to read a document. The participant was given 8 minutes to read and if they did not finish in this time then the percentage of completion was recorded and used to estimate their finish time. This measure shows how easy the document is to read. We would expect an easier document to be read more quickly than a difficult document.

The next measure of understanding is a series of multiple choice questions. 6

questions of varying difficulty were created for each document. Each question had 3 possible answers from the text, as well as a fourth option 'I don't know'. We read the questions to the participants to maximise their understanding of the question and possible answers. The order of the questions was randomised in terms of the difficulty of the questions and the location of the answers in the text. The participants were not allowed to review the text whilst answering the questions. A high score on this text indicates that the participant has read, understood and remembered key information from the text.

Finally, we presented the participant with two self assessment questions. The first question was 'How easy was the piece to read' and the second was 'How well did you understand the piece'. These questions tell us how well a participant felt that they interacted with each text. For each question, the participant was given a five point Likert scale with the options: '(1) Very Good, (2) Good, (3) OK, (4) Bad, (5) Very Bad'. They were also presented with visual aids in the form of happy and sad faces to help in answering these questions.

7.2.5 Delivery

We undertook two visits with each participant. These visits were separated by 4–5 weeks. The time between visits was enforced to minimise the amount of information a participant would remember. Each participant self-reported remembering very little about each document. In the first visit, a participant was shown one complex and one simple document. In the second visit they were shown the opposite pair. The participants did not know that one of the documents had been simplified.

Participants were asked to read the documents either out loud or silently depending on their preference. The same reading style was preserved across all four documents. After the participant had finished reading each document, we recorded the time and percentage complete. We then asked them to fill in the multiple choice questions and the experience questionnaire. Participants were offered a short break in between documents.

We created four ability groups from our 9 participants based on their BNT scores. The first group contained three participants, whereas the other three groups contained 2 participants. Within each group, we alternated the presentation order of the documents. We showed Document 1 first half the time and Document 2 first half the time. We also showed the unsimplified version of the document first half of the time and the simplified version of the document first the other half of the time. The document

Table 7.3: The order in which we showed each document to each participant. The order of the documents was alternated to give a balanced set in the later analysis.

Patient	Visit 1		Visit 2		Days Between Visits
	1st Doc	2nd Doc	1st Doc	2nd Doc	
1a	C1	S2	C2	S1	35
1b	S1	C2	S2	C1	32
1c	C1	S2	C2	S1	34
2a	S2	C1	S1	C2	32
2b	C2	S1	C1	S2	35
3a	C1	S2	C2	S1	35
3b	S1	C2	S2	C1	27
4a	S2	C1	S1	C2	34
4b	C2	S1	C1	S2	34

schedule is shown in Table 7.3.

7.3 Results

There are two modes of comparison which we can make in our data. Firstly, we can look at the difference between the scores for the complex and simple version of each document. Each participant was shown both versions of each document across the two visits. Secondly, we can look at the difference between a complex and simple document within a single visit. Each visit contained one simple and one complex document. We arranged the documents to balance conflicting factors. Thus, there was a balance between first and second visits when comparing the complex version of a document against its simple counterpart. There was also a balance between both documents when comparing between the first and second visit.

We present Tukey Box Plots of extrapolated time in Figure 7.1. Extrapolated time is calculated as time divided by percentage read. All participants, apart from one, completed the reading task within the allotted time. It is clear from this graph that there is one participant whose scores were classified as an outlier. We have chosen not to exclude this participant as the matched and balanced style of our experiment will mitigate the effect of the outlier. We show a bar chart of the scores from the multiple choice questionnaires in Figure 7.2. We also show bar charts of the Reading and Understanding self-assessment questions in Figure 7.3 and 7.4.

To choose a suitable significance test, we looked at the following constraints. We have one independent variable: whether the document has been simplified. We have

several dependent variables: Extrapolated time is continuous, but not normally distributed; MCQ score is an interval variable; reading ease and understandability are both ordinal variables. The samples were matched within each patient and the groups were balanced. Following these considerations, we chose the Wilcoxon signed-rank test (Wilcoxon, 1945). This test tells us whether there is a significant dependency between an independent and a dependent variable. It is well suited to this task as it can be used with limited numbers of data points.

In our analysis, we have looked at the mean change for each document and visit pairing. The means, standard deviations, deltas and p-values are reported in Table 7.4. It is clear from this table that we did not observe an increase in understanding when simplification was applied to a document. Surprisingly, we observe a significant increase in reading time when simplification is applied in many cases. We see MCQ scores decreasing with simplification in most cases. We see the scores for Reading Ease and Understanding increasing. We assigned 1 to very good and 5 to very bad, so an increase in score indicates a decrease in understanding and reading ease. We also performed a subgroup analysis. We did not analyse at the level of pairs, as we felt that this would leave us with too few data points. Instead, we created 2 ability groups from our data. We created a low ability group consisting of groups 1 and 2. We also created a higher ability group consisting of groups 3 and 4. These results are presented in Tables 7.5 and 7.6. In the sub-group analyses, we present the combined results for both the documents and the visits, as the single document and visit analyses do not have enough information to be reliable when only considering the subgroups.

Given the surprising result that mean time increases with simplification, we decided to investigate further. We created a Stupid-Backoff Language Model trained on 5-grams from the Google Web 1T. We scored each document using this model, we also scored each sentence and took the mean. In both cases the language model score decreased after simplification, which implies a text is less likely to occur. We calculated the Pearson Correlation between the language model scores and the Extrapolated times for each document for all the groups together as well as the sub groups previously mentioned. The results can be seen in Table 7.7.

Finally, we ran a one way ANOVA test for both the document pairs and the visit pairs to investigate the significance of considering high and low ability sub-groups and different documents. Our four groups were: Low ability document 1, low ability document 2, high ability document 1 and high ability document 2. We repeated the ANOVA with the document pairs from each visit. The results of the ANOVA are

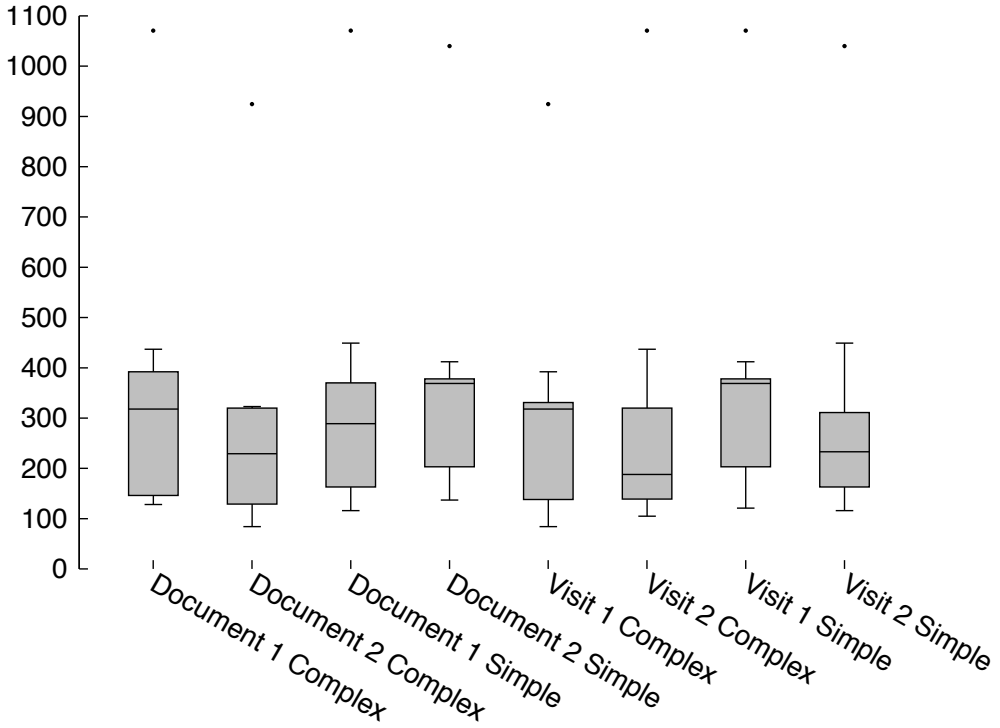


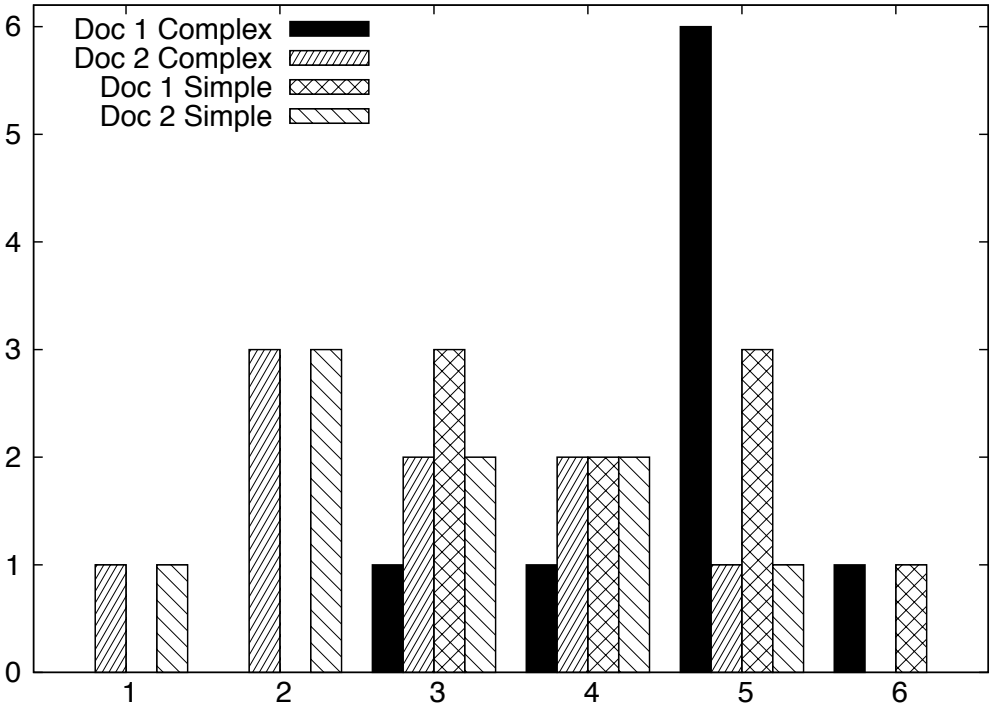
Figure 7.1: Tukey Box plots of the extrapolated time for our experiment.

reported in Table 7.8. We find a statistically significant difference between groups for reading time both for documents ($F(3,32) = 2.92,\ P = 0.0488$) and visits ($F(3,32) = 2.98,\ P = 0.0459$). We also find a statistically significant difference between groups for the multiple choice score between documents ($F(3,32) = 5.97,\ P = 0.0024$).

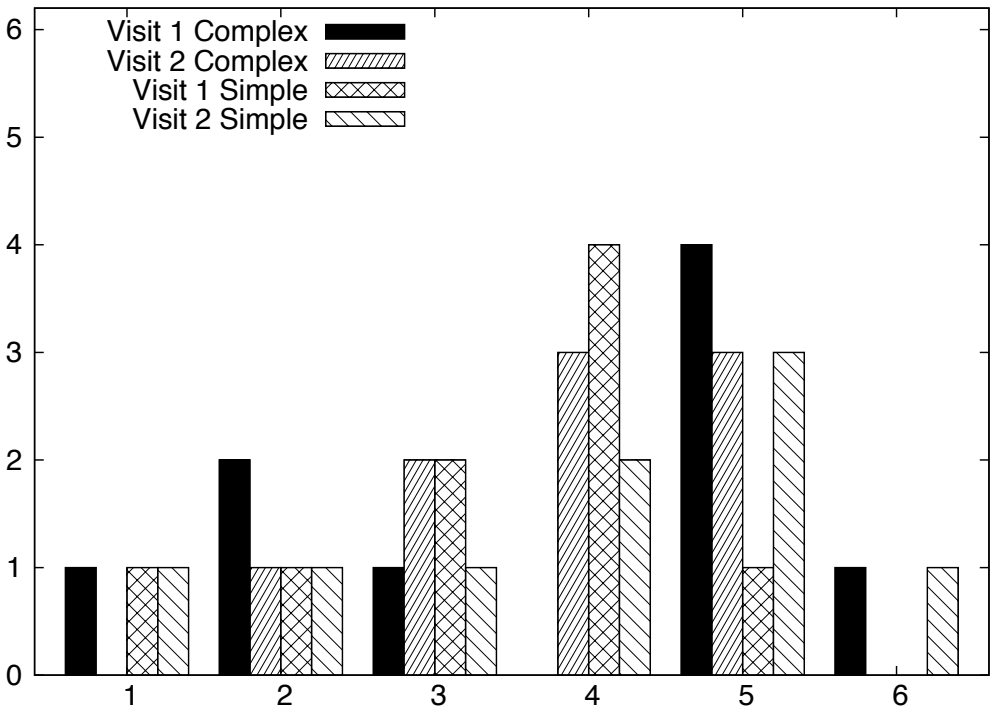
7.4 Discussion

It is clear from our results that simplification did not have a positive effect on the metrics which we have measured. In almost all cases, the significance test showed us that there was a very high probability of no difference in the results after simplification. This finding indicates that most of our results are not significant. With more participants in this study, we may see significant effects emerging, we may also see effects changing in direction and magnitude.

We observed that reading time increased for almost all of our analyses. The only decrease in reading time was observed for document 1, when considering all groups. This decrease was small at 10 seconds compared to other greater increases. It also

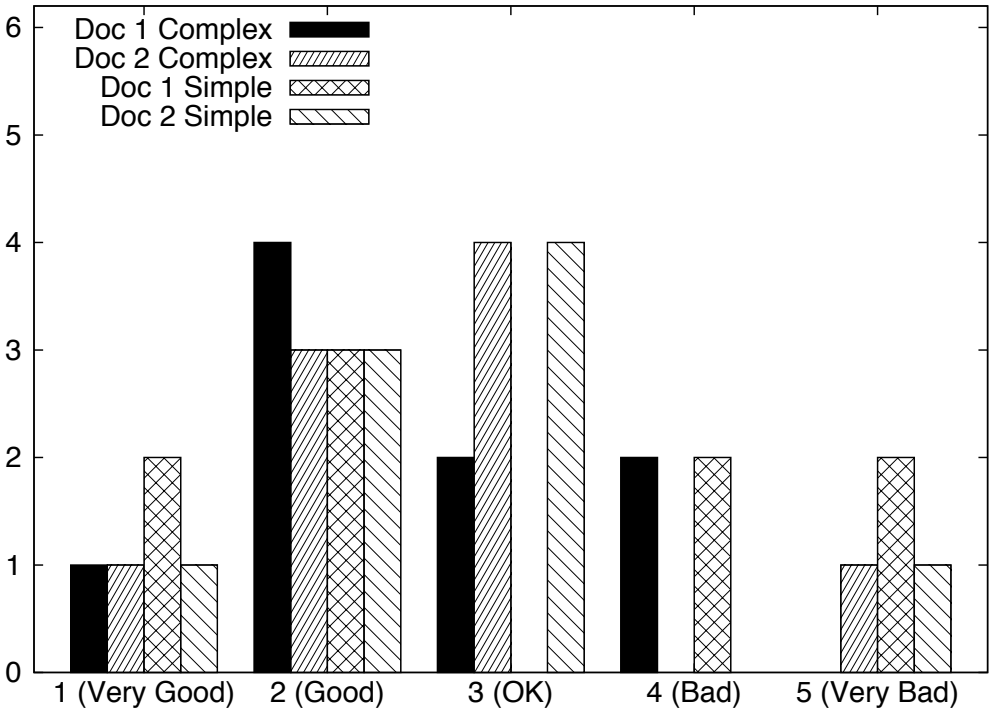


(a) A bar chart showing the scores on the multiple choice questions for each document.

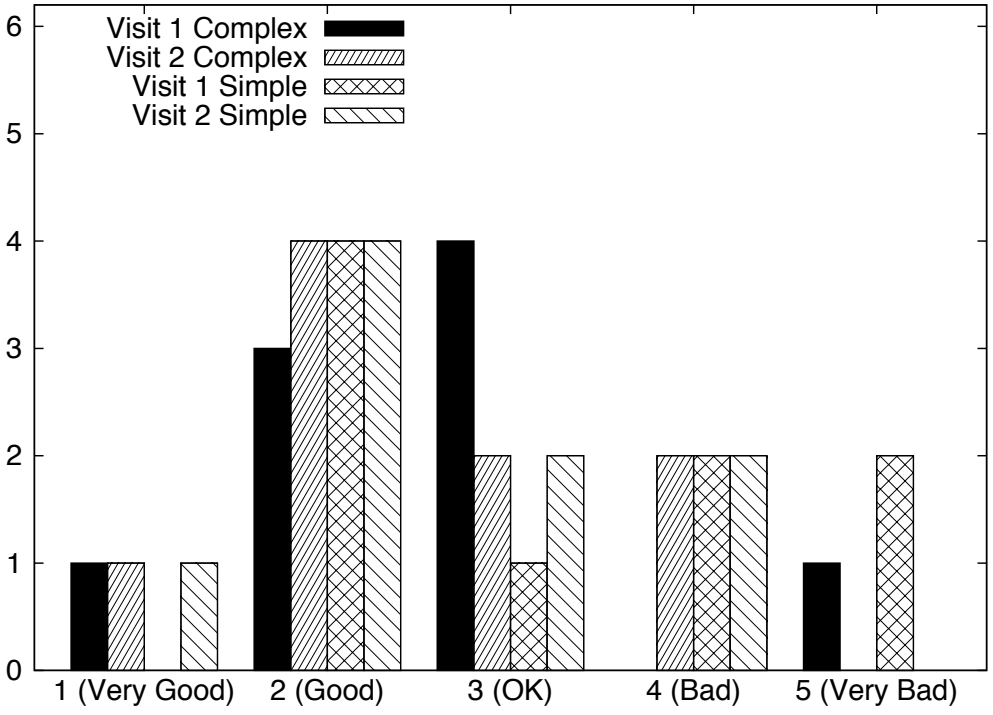


(b) A bar chart showing the scores on the multiple choice questions for each visit.

Figure 7.2: The multiple choice scores for each document and each visit.

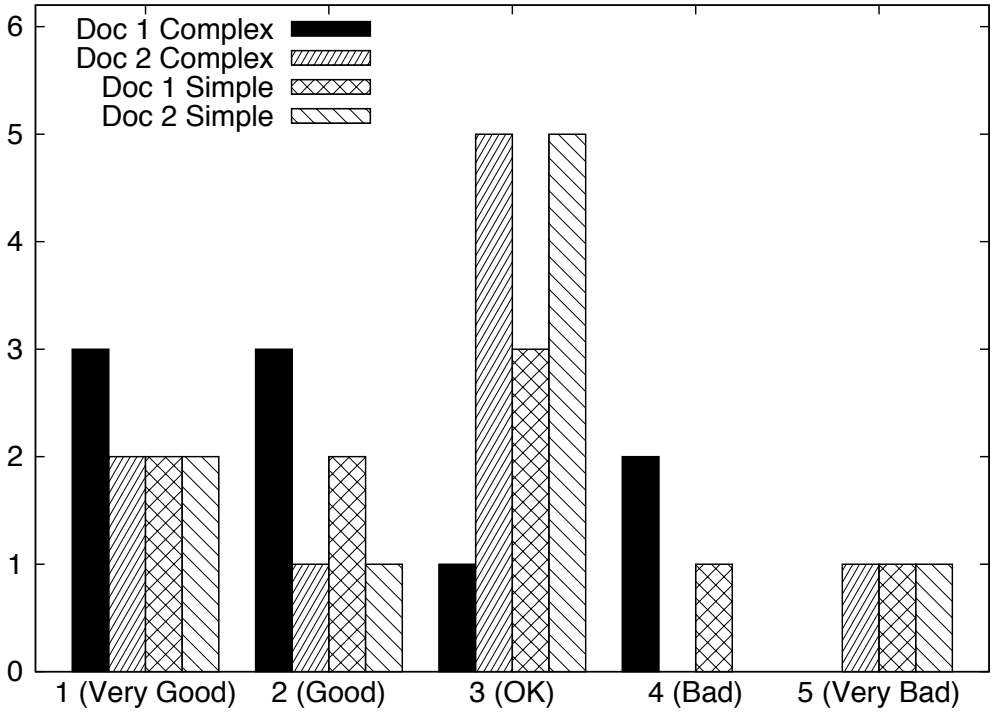


(a) A bar chart showing the patients' assessment of readability for each document.

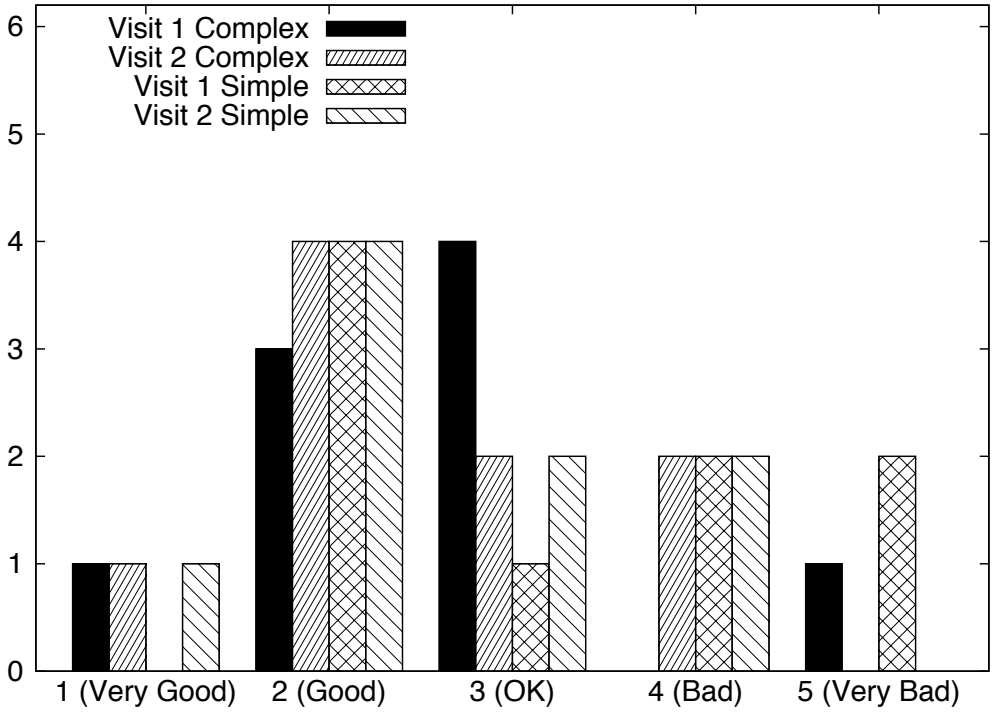


(b) A bar chart showing the patients' assessment of readability for each visit.

Figure 7.3: The answers to the readability question for each document and each visit.



(a) A bar chart showing the patients' assessment of understanding for each document.



(b) A bar chart showing the patients' assessment of understanding for each visit.

Figure 7.4: The answers to the understanding question for each document and each visit.

Table 7.4: The results from our analysis. P values are calculated using a two-tailed Wilcoxon signed-rank test. The delta is the change in score after simplification has been applied. Values which are significant at the $P < 0.1$ level are reported in bold. Values which are significant at the $P < 0.05$ level are reported in bold and italics.

		Extrapolated Time	**MCQ**	**Readability Score**	**Understandability Score**
Document 1	Delta	-10.333	-0.556	0.333	0.444
	P Value	0.8750	0.3125	0.75	0.5312
Document 2	Delta	89.951	0.111	0.000	0.111
	P Value	***0.0078***	0.8594	1.000	1.000
Document 1 + 2	Delta	39.81	-0.22	0.17	0.28
	P Value	**0.0720**	0.7520	0.6836	0.4253
Visit 1	Delta	53.59	-0.44	0.56	0.22
	P Value	***0.0078***	0.6875	0.3125	0.6250
Visit 2	Delta	18.58	0.00	-0.22	0.33
	P Value	0.2031	1.00	0.7500	0.2500
Visit 1 + 2	Delta	36.09	-0.22	0.17	0.28
	P Value	***0.0050***	0.7346	0.6836	0.1250

Table 7.5: The results from our analysis for groups 1 and 2. P values are calculated using a Wilcoxon signed-rank test.

		Extrapolated Time	**MCQ**	**Readability Score**	**Understandability Score**
Document 1 + 2	Delta	48.46	0.30	0.20	0.10
	P Value	**0.0977**	0.5625	0.7656	1.00
Visit 1 + 2	Delta	41.76	0.30	0.20	0.10
	P Value	***0.0488***	0.6211	0.6875	1.00

Table 7.6: The results from our analysis for groups 3 and 4. P values are calculated using a Wilcoxon signed-rank test.

		Extrapolated Time	**MCQ**	**Readability Score**	**Understandability Score**
Document 1 + 2	Delta	29.00	-1.00	0.13	0.50
	P Value	0.4141	0.1250	1.00	0.5000
Visit 1 + 2	Delta	29.00	-0.88	0.13	0.50
	P Value	***0.0156***	0.3750	1.00	0.1250

Table 7.7: The correlations between the extrapolated reading time and the language model score for each document. The correlation is much stronger for groups 1 and 2 than for groups 3 and 4.

	Extrapolated Time (All Groups)	**Extrapolated Time (Groups 1 and 2)**	**Extrapolated Time (Groups 3 and 4)**
Full Text	-0.510	-0.678	-0.247
Mean Sentence	-0.501	-0.633	-0.288

Table 7.8: The results of the one way ANOVA test. These results show the probability of the results for each set of documents and abilities being drawn from the same distribution as each other. Results which are significant at $P < 0.05$ are reported in bold and italics.

	Extrapolated Time	**MCQ**	**Readability Score**	**Understand-ability Score**
Documents	***0.0488***	***0.0024***	0.8224	0.8164
Visits	***0.0459***	0.6240	0.5508	0.5588

showed to be non-significant, according to the Wilcoxon signed-rank test. We observed two other non-significant time changes. The first was the difference between the second visit original and simplified documents, where a small increase was observed. The second non-significant time change was observed between the reading times of groups 3 and 4 when looking at the differences between all of the complex and simple document reading times. A small increase was observed. Otherwise, all other analyses showed that reading time increased significantly when simplification was applied to a document.

For the multiple choice questions no significant effect was observed. Looking at the deltas for MCQ in Tables 7.5 and 7.6, we see a small increase in multiple choice score for both analyses of groups 1 and 2 but we see a decrease in score for groups 3 and 4. The ANOVA results in Table 7.8 show that there is a significant difference in multiple choice scores between the groups. This finding may imply that simplification is best suited for lower ability users. Further testing with the low ability group may reveal a deeper effect. It may also be the case that simplification was not relevant to the higher ability readers, as they were not struggling with the original text. Whereas the lower ability users did struggle with the original text and so some simplification was helpful to them.

For the metrics ‘readability’ and ‘understandability’, the scale we employed marks 1 as ‘very good’ and 5 as ‘very bad’, so an increase in score indicates a shift towards

'very bad'. These scores are not significant in any of our analyses. The scores increase almost ubiquitously, implying a decrease in readability and understandability of the documents after simplification was applied. A small decrease in score is observed for readability on the second visit for all groups, however the result was not significant.

The conclusions of this analysis lead us to reject all three of our original research hypotheses. We find no evidence at this stage that LS is a helpful tool for people with aphasia. Having observed the large increase in reading time, we may reformulate RH 7.1 as follows:

> **RH 7.1b:** People with aphasia will read documents with LS more slowly than equivalent documents without simplification.

Clearly, we can accept this hypothesis, based on the broad trends of our data and the supporting p-values.

The scores from the language model correlate more strongly with the reading times of the lower ability group than with the higher ability group. This result suggests that the problem for the lower ability group was the unfamiliar word sequences introduced by the simplification system. Language model score could be used in the future as a predictor of LC for people with a more severe aphasia. In our experiments, the patients in the lower ability groups had a BNT score of less than 30, whereas the patients in the higher ability group had a BNT score of greater than 30. This threshold could be used to determine whether or not to use the language model as a predictor of LC for people with aphasia, or indeed whether or not simplification should be applied at all. The correlations are based on only four data points, one per document. The amount of data is very small in this case and we must treat these numbers with caution.

Our results corroborate the findings of Leroy et al. (2013b). We have shown that LS can be a hindrance in some instances. It is interesting to note that Leroy et al. (2013a) extend the work mentioned previously to incorporate LS into an author's aid system for authors of health literature, with positive results. We might find a similar positive result if we were to create a similar system for authors of literature intended for people with aphasia.

We saw in Section 7.1 that LS has had a positive effect for other types of text and user groups, such as people with dyslexia. Although our results show that this form of simplification has so far not been effective for people with aphasia, we might find that other forms of simplification and assistive technology are useful. We may also find that LS gives positive results with different subject groups.

Our initial motivation for this research was to use simplification to aid in stroke recovery. Although this experiment has not confirmed our hypothesis that lexical simplifications would be helpful to people with aphasia, this finding does not close off the avenue of research. Future research may look at building an aphasia-friendly measure of LC. There may also be further opportunities to implement assistive technologies to aid in stroke recovery.

Chapter 8

Future Work

In this thesis, we have explored many avenues of research. We have answered important research questions and shed light on new phenomena. We have built resources and made them available to the community. Further extensions may still be applied to this work. In this chapter we have separated the future work into extensions for each individual chapter. We have also provided a general future work section which suggests further research aims for LS.

8.1 Error Study — Chapter 3

In the error study, we categorised the types and quantities of errors in the LS pipeline. We saw that in an unmodified pipeline, the errors are prevalent from the early stages. Future work should concentrate on the mitigation of errors in the LS pipeline, as we have done in this thesis. Our work should help to focus and motivate future research into the LS pipeline.

The form of error analysis shown in the error study 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 incorporate elements of automated evaluation which could also be incorporated into the LS pipeline. It would be possible to provide a degree of certainty to each simplification and flag possibly erroneous simplifications to a user or an author.

It would also be interesting to repeat this analysis with current state-of-the-art resources. We would hope to see the number of successful simplifications increasing. The decrease in each error category would be related to the optimisations applied to the pipeline. If a new CW identification method was used then the errors related to CW identification would decrease. If a new WSD method was used then the errors related to WSD would decrease.

8.2 CW Corpus — Chapter 4

The CW corpus is quite small at 731 instances. It may be grown by mining the revision histories of the main English Wikipedia in the same manner as for simple Wikipedia. Whilst the English Wikipedia revision histories will have fewer valid simplifications per revision, they are much more extensive and contain a lot more data. As simple Wikipedia grows, we can also rerun this experiment with the simple Wikipedia data and discover more simplifications as new edits are made.

We also hope that this resource will be useful to the wider community. It has been useful to this research as we used it to evaluate machine learning techniques for CW identification in Section 5.3. Paetzold (2015) makes use of the CW Corpus in a user study to identify CWs. We hope that future efforts to improve CW identification will also make use of this resource.

8.3 Lexical Complexity — Chapter 5

It would be interesting to perform an in depth analysis of the features used in the classification scheme in the LC chapter. Whereas frequency is known to have a strong effect, we could examine the effect of the other feature groups and individual features. By eliminating features with lower correlations, it may be possible to optimise the classification scheme and further increase corpus accuracy.

Work on how to apply a relative complexity model must also be undertaken. Two tasks from the LS field are complex word identification and substitution ranking. The model could be employed for the identification task by picking a reference set of words and testing to see whether given corpus words are easier or more difficult to understand than the reference set. The model could be employed for the ranking task by implementing some form of pairwise ranking over the elements of a list. Note that there is no guarantee of transitivity in the relations imposed by the model and so tie situations may arise. This case would imply that a set of words is at a similar level of complexity.

It would also be interesting to augment the multiword corpus with entries from other domains. At present we have only investigated MWEs from the biomedical domain and we may see different results when looking at larger datasets and different domains.

8.4 Substitution Generation — Chapter 6

In the substitution generation chapter, we looked at the available resources for substitution generation and WSD. We found that current resources are typically not powerful enough to produce error-free simplifications. Future work in LS must make use of techniques at the cutting edge of WSD. Advances in WSD are often driven through the SemEval shared tasks and future integration with SemEval would be beneficial to LS.

With regards to both disambiguation and thesaurus coverage, it may be the case that performance gains can be attained by focusing on specific genres of text. Whereas this

task is very difficult for a general language application, it may be easier for a narrow domain or controlled language. In such a case there will be fewer simplifications to learn and fewer ambiguous words to simplify.

8.5 User Study — Chapter 7

The user study showed a surprising negative result, which indicated that people with aphasia take more time to read simplified documents than their unsimplified counterparts. This result may have been because the simplification process produced less natural language. We must ask then: how can we use LS to produce more natural sounding language? One possibility would be to modify the measure of LC to take context into account. We could use the language model to only select sentences which have a higher probability than the original sentence.

An appropriate extension to the user study would be to repeat the experiment with non-aphasic controls. The data from the control subjects would tell us whether simplification is useful only in the case of people with a cognitive impairment such as aphasia, or whether simplification is also useful for the general population. We would expect to see a different effect for people with aphasia than the effect we observe for our control participants. The subjects in the control experiment should be matched for similar age and education level as the participants with aphasia from the original study.

Our stated aim in this aspect of our research is to automate the process of producing aphasia-friendly text. LS is only one aspect of the aphasia-friendly guidelines (The Stroke Association, 2012). Further work in automated reading aids for people with aphasia could also focus on other aspects of these guidelines. Syntactic simplification, text summarisation, text presentation and other factors would all be useful in this context.

8.6 General Applications

The development of automated TS systems is an important research challenge. These systems will work as assistive technology, helping people to access information. There are two clear options for the development of publicly available TS systems, as outlined below.

Firstly, TS can be applied at the user's level. In this model, the user receives some

complex text which they automatically simplify by some means. This style of simplification could take on the form of a Web browser plug-in which allows the user to select and simplify text (similar to the FACILITA project for Brazilian Portuguese (Watanabe et al., 2009)). This style could also take on the form of a smartphone application which allows the user to take a picture of some text and then via optical character recognition is able to identify the text in question and simplify it in an augmented reality fashion. Some users may not even require the choice to simplify text. For example, in the context of browsing the Internet, some users may become distracted by the initial difficult text. It may be helpful to automatically reduce the complexity of any text on a Webpage before presenting it to a user.

Secondly, TS may be applied by the author to a text he is creating. In this model, the author may write a document and then use automatic techniques to identify any complexities and to automatically simplify or receive suggestions as to simplifications he may apply. The main advantage is that the author can check the quality of simplifications before the text is presented to a user. Grammaticality, cohesion and intended meaning are definitely preserved, whilst understandability and readability are increased. This model is useful in many different applications where text is being written for audiences who may not necessarily understand the final product. The research challenge here is to develop ways of doing this type of simplification which are helpful and also allow an author to target his text to many levels of understanding.

Chapter 9

Conclusion

The work of this PhD has provided insights into the area of LS. We initially found that the research into LS was sparse, scattered over the 15 years prior to the start of this work. We have used this PhD to collate and better understand the area of LS. We have built resources and created new evaluation measures where necessary.

Our first approach was to survey the literature surrounding LS and the wider field of TS. Through this survey, we found many different types of simplification. For LS, we discovered that most forms of simplification either followed a specific pipeline of operations or could be expressed in terms of this pipeline. We have reproduced the LS portion of the survey in Chapter 2.

Following on from this survey, we felt that it would be appropriate to investigate the pipeline further. We first noticed that no appropriate resources existed for the evaluation of the identification of CWs. To address this need we built the CW corpus, documented it appropriately and made it publicly available. At the time of writing it has been downloaded 20 times from the META-SHARE repository. We also used the CW Corpus to investigate different techniques for CW identification.

Later, we realised that it would be possible to perform a manual evaluation of the types and frequencies of errors in the LS pipeline. Whereas other research appears to have widely omitted to mention the vast frequency of errors encountered in the LS pipeline, we were the first to expose these errors. This work was well received by the community.

We used the results of this error study to further guide the applications of the research. We originally intended to spend time looking at substitution ranking but instead focused on LC, which also has applications in the area of CW identification. We noticed that many of the substitutions were problematic as no simple substitutions could be found, or indeed no simple substitutions with the same meaning. This discovery led to the work in Chapter 6, which evaluated several state of the art resources and made recommendations for future researchers hoping to generate substitutions in the LS pipeline.

Our final research contribution took us on a diversion away from the theoretical and into the practical. The user study was a natural progression of the work on the pipeline that preceded it. We were surprised to discover that LS did not help people with aphasia in our case, but in fact appeared to hinder. We have suggested explanations for this effect and we hope that our negative result will influence future research working to build communication aids for people with aphasia.

The contributions of this research, along with research hypotheses and outcomes were as follows:

- An analysis of the errors produced by the processing pipeline standardly used in LS. We discovered that many errors occur at the earlier stages of the LS pipeline.
- The CW Corpus. A parallel corpus of sentences each with one lexical change, which is a simplification.
- RQ 5.1: How can we detect CWs without using a simple threshold or simplifying every word?
- We were able to prove RH 5.1, which stated: A feature based machine learning technique will provide higher accuracy in the CW identification task than the baseline techniques.
- We confirmed RH 5.2, showing that a measure of lexical complexity can be improved by considering relative feature values.
- RQ 5.3: How do environmental factors affect corpus frequencies?
- We rejected RH 5.3 in favour of the null hypothesis. RH 5.3 stated that: Users' decisions as to which words are difficult to understand can be modelled by adapting a frequency resource to their needs.
- We confirmed RH 5.4, which stated: Users' decisions as to which words are difficult to understand can be modelled by adapting a frequency resource to the genre of the text.
- RQ 5.4: How can we assign feature values to MWEs for LC?
- We found evidence to support RH 5.5, which stated: MWEs derive their LC features from their constituent words.
- RQ 5.5: How do the features of constituent words combine to create features for a MWE?
- We confirmed RH 5.6, which stated: The most difficult features in a MWE will cause the expression to be difficult to understand.
- We rejected RH 5.7, which stated: The most simple features in a MWE will cause the expression to be simple to understand.
- We made recommendations to the field regarding how to build resources which are both comprehensive and produce meaningful substitutions.

- We presented the results of a study investigating how helpful automated LS may be for people with aphasia. We showed a negative result, indicating that people with aphasia take longer to read documents which have been simplified using automated LS.
- The research into LS for people with aphasia was conducted following RQ 7.1, which states: Does automated LS improve text comprehension for people with aphasia?
- We rejected RH 7.1 in favour of the null hypothesis. RH 7.1 stated: People with aphasia will read documents with LS more quicly than equivalent documents without simplification.
- We rejected RH 7.2 in favour of the null hypothesis. RH 7.2 stated: People with aphasia will understand documents with LS better than equivalent documents without simplification.
- We rejected RH 7.3 in favour of the null hypothesis. RH 7.3 stated: People with aphasia will judge documents with LS to be easier to understand than equivalent documents without simplification.
- We reformulated RH 7.1 to give RH 7.1b, which we accepted. RH 7.1b stated: People with aphasia will read documents with LS more slowly than equivalent documents without simplification.

We hope that this research will be useful to the wider community, both as a collective body and as individual portions of work. We aim to publish the final pieces of research in relevant venues in order to make them available to the research community. We are aware of other researchers working in the field of LS and hope that they will find the results contained in this thesis helpful in guiding their own work. As with many PhDs, we have opened more questions than we have resolved. We look forward to seeing answers to these open research questions in future literature.

Bibliography

Abrahamsson E., Forni T., Skeppstedt M., and Kvist M. Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 57–65, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1207.

Adriaens G. Simplified English grammar and style correction in an MT framework: the LRE SECC project. In *Aslib proceedings*, volume 47, pages 73–82. MCB UP Ltd, 1995. URL http://dx.doi.org/10.1108/eb051383.

Agirre E. and Soroa A. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1609067.1609070.

Aha D., Kibler D., and Albert M. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991. URL http://dx.doi.org/10.1007/BF00153759.

Alfano M., Lenzitti B., Lo Bosco G., and Perticone V. An automatic system for helping health consumers to understand medical texts. In *Proccedings of The International Conference on Health Informatics (HEALTHINF2015)*, Lisbon, January 2015. URL http://dx.doi.org/10.5220/0005283606220627.

Aluísio S. M. and Gasperin C. Fostering digital inclusion and accessibility: the PorSimples project for simplification of Portuguese texts. In *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, YIWCALA '10, pages 46–53, Stroudsburg, PA, USA, 2010.

Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1868701.1868708.

Amoia M. and Romanelli M. SB:mmSystem - using decompositional semantics for lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 482–486, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S12-1067.

Angrosh M., Nomoto T., and Siddharthan A. Lexico-syntactic text simplification and compression with typed dependencies. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1996–2006, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL http://www.aclweb.org/anthology/C14-1188.

Aranzabe M. J., de Ilarraza A. D., and Gonzalez-Dios I. First approach to automatic text simplification in Basque. In *Natural Language Processing for Improving Textual Accessibility (NLP4ITA) Workshop Programme*, pages 1–8, 2012. URL http://www.taln.upf.edu/pages/nlp4ita/pdfs/aranzabe-nlp4ita2012.pdf.

Azab M., Hokamp C., and Mihalcea R. Using word semantics to assist English as a second language learners. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 116–120, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-3024.

Baeza-Yates R. and Ribeiro-Neto B. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. URL http://dl.acm.org/citation.cfm?id=553876.

Baeza-Yates R., Rello L., and Dembowski J. Cassa: A context-aware synonym simplification algorithm. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1380–1385, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1156.

Balota D., Yap M., Hutchison K., Cortese M., Kessler B., Loftis B., Neely J., Nelson D., Simpson G., and Treiman R. The english lexicon project. *Behavior Research Methods*, 39:445–459, 2007. URL http://dx.doi.org/10.3758/BF03193014.

Barbu E., Martín-Valdivia M. T., and Ureña López L. A. Open book: a tool for helping ASD users' semantic comprehension. In *Proceedings of the Workshop on Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, pages 11–19, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-1502.

Barbu E., Martín-Valdivia M., Teresa, Martínez-Cámara E., and Ureña López L., Alfonso. Language technologies applied to document simplification for helping autistic people. *Expert Syst. Appl.*, 42(12):5076–5086, July 2015. URL http://dx.doi.org/10.1016/j.eswa.2015.02.044.

Barzilay R. and Elhadad N. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pages 25–32, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1119355.1119359.

Baumgarten A. I. *The Phoenician history of Philo of Byblos: A Commentary*. E.J. Brill, 1981.

Bautista S., Hervás R., Gervás P., Power R., and Williams S. A system for the simplification of numerical expressions at different levels of understandability. In *Proceedings Of The Worshop On Natural Language Processing for Improving Textual Accessibility (NLP4ITA)*, Atlanta, USA, June 2013.

Beigman Klebanov B., Knight K., and Marcu D. Text simplification for information-seeking applications. In Meersman R. and Tari Z., editors, *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, volume 3290 of *Lecture Notes in Computer Science*, pages 735–747. Springer Berlin Heidelberg, 2004. URL http://dx.doi.org/10.1007/978-3-540-30468-5_47.

Biran O., Brody S., and Elhadad N. 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, 2011. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2002736.2002835.

Blake C., Kampov J., Orphanides A. K., West D., and Lown C. Query expansion, lexical simplification and sentence selection strategies for multi-document summarization. In *Proceedings of the Document understanding conference (DUC-2007)*, 2007.

Blum S. and Levenston E. A. Universals of lexical simplification. *Language Learning*, 28(2):399–415, 1978. URL http://dx.doi.org/10.1111/j.1467-1770.1978.tb00143.x.

BNC. The British National Corpus, version 3 (BNC XML Edition), 2007. Distributed by Oxford University Computing Services on behalf of the BNC Consortium.

Bott S. and Saggion H. An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 20–26, Portland, Oregon, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W11-1603.

Bott S. and Saggion H. Text simplification resources for Spanish. *Lang. Resour. Eval.*, 48(1):93–120, March 2014. URL http://dx.doi.org/10.1007/s10579-014-9265-4.

Bott S., Rello L., Drndarević B., and Saggion H. Can Spanish be simpler? LexSiS: Lexical simplification for Spanish. In *Proceedings of COLING 2012*, pages 357–374, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-1023.

Brants T. and Franz A. Web 1T 5-gram corpus version 1.1. *Linguistic Data Consortium*, 2006.

Brants T., Popat A., Xu P., Och F., Dean J., and Inc G. Large language models in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 858–867, 2007.

Breiman L. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Brin S. and Page L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, April 1998. URL http://dx.doi.org/10.1016/S0169-7552(98)00110-X.

Brunato D., Dell'Orletta F., Venturi G., and Montemagni S. Design and annotation of the first Italian corpus for text simplification. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 31–41, Denver, Colorado, USA, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W15-1604.

Brysbaert M. and New B. 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, 2009.

Carlomagno S., Pandolfi M., Labruna L., Colombo A., and Razzano C. Recovery from moderate aphasia in the first year poststroke: Effect of type of therapy. *Archives of Physical Medicine and Rehabilitation*, 82(8):1073 – 1080, 2001.

Carroll D. W. *Psychology of language*. Brooks/Cole Pub. Co., 1986.

Carroll J., Minnen G., Canning Y., Devlin S., and Tait J. 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, 1998.

Chandrasekar R. and Srinivas B. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10(3):183 – 190, 1997.

Chen H.-B., Huang H.-H., Chen H.-H., and Tan C.-T. A simplification-translation-restoration framework for cross-domain SMT applications. In *Proceedings of COLING 2012*, pages 545–560, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-1034.

Clark S. Vector space models of lexical meaning. In Lappin S. and Fox C., editors, *Handbook of Contemporary Semantics second edition*. Wiley-Blackwell, 2012.

Cohen J. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968. URL http://www.ncbi.nlm.nih.gov/pubmed/19586159.

Cohen W. W. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

Collantes M., Hipe M., Sorilla J. L., Tolentino L., and Samson B. Simpatico: A text simplification system for senate and house bills. In *Proceedings of The 11th National Natural Language Processing Research Symposium*, 2015.

Collins-Thompson K. Computational assessment of text readability: a survey of current and future research. *Recent Advances in Automatic Readability Assessment and Text Simplification. Special issue of International Journal of Applied Linguistics*, 165(2):97–135, 2014.

Coltheart M. The MRC psycholinguistic database. *Quarterly Journal of Experimental Psychology Section A-human Experimental Psychology*, 33:497–505, 1981. doi: 10.1080/14640748108400805.

Crossley S. A., Allen D., and McNamara D. S. Text simplification and comprehensible input: A case for an intuitive approach. *Language Teaching Research*, 2012.

Curran J. R. *From Distributional to Semantic Similarity*. PhD thesis, University of Edinburgh, 2004.

Daelemans W., Zavrel J., Sloot K., and Bosch A. V. D. TiMBL: Tilburg Memory-Based Learner, version 6.2, Reference Guide. ILK Technical Report 09-01, ILK, 2009. URL `http://ilk.kub.nl/~ilk/papers/ilk9803.ps.gz`.

Dale E. and Chall J. S. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54, 1948.

De Belder J. and Moens M.-F. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*, pages 19–26, 2010.

De Belder J. and Moens M.-F. A dataset for the evaluation of lexical simplification. In *Computational Linguistics and Intelligent Text Processing*, volume 7182 of *Lecture Notes in Computer Science*, pages 426–437. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-28600-1.

De Belder J., Deschacht K., and Moens M.-F. Lexical simplification. In *1st International Conference on Interdisciplinary Research on Technology, Education and Communication*, 2010.

Decadt B., Hoste V., Daelemans W., and Van den Bosch A. Gambl, genetic algorithm optimization of memory-based WSD. In Mihalcea R. and Edmonds P., editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 108–112, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Deléger L. and Zweigenbaum P. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora*, pages 2–10. Association for Computational Linguistics, 2009.

Deschacht K., De Belder J., and Moens M.-F. The latent words language model. *Comput. Speech Lang.*, 26(5):384–409, October 2012. URL `http://dx.doi.org/10.1016/j.csl.2012.04.001`.

Devlin S. *Simplifying natural language for aphasic readers.* PhD thesis, University of Sunderland, 1999.

Devlin S. and Tait J. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, pages 161–173, 1998.

Devlin S. and Unthank G. 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, 2006. ACM. URL `http://doi.acm.org/10.1145/1168987.1169027`.

Dingli A. and Cachia C. Adaptive ebook. In *Proccedings of the International Conference on Interactive Mobile Communication Technologies and Learning (IMCL14)*, pages 14–19, Nov 2014.

Doddington G. Automatic evaluation of machine translation quality using N-Gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. URL `http://dl.acm.org/citation.cfm?id=1289189.1289273`.

Drndarević B. and Saggion H. Towards automatic lexical simplification in Spanish: An empirical study. In *Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations*, pages 8–16, Montréal, Canada, June

2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-2202.

Drndarević B., Štajner S., Bott S., Bautista S., and Saggion H. Automatic text simplification in Spanish: A comparative evaluation of complementing modules. In Gelbukh A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 7817 of *Lecture Notes in Computer Science*, pages 488–500. Springer Berlin Heidelberg, 2013. URL http://dx.doi.org/10.1007/978-3-642-37256-8_40.

Elhadad N. Comprehending technical texts: Predicting and defining unfamiliar terms. In *AMIA Annual Symposium proceedings*, pages 239–243. American Medical Informatics Association, 2006.

Elhadad N. and Sutaria K. 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, 2007.

Eom S., Dickinson M., and Sachs R. Sense-specific lexical information for reading assistance. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 316–325, Montréal, Canada, June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-2038.

Fellbaum C. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

Feng L. Text simplification: A survey. Technical report, City University New York, 2008.

Fleiss J. L. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76:378–382, November 1971.

Gonzalez-Dios I., Aranzabe J. M., Díaz de Ilarraza A., and Salaberri H. Simple or complex? assessing the readability of Basque texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 334–344. Dublin City University and Association for Computational Linguistics, 2014. URL http://aclweb.org/anthology/C14-1033.

Grabar N., Hamon T., and Amiot D. Automatic diagnosis of understanding of medical words. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 11–20. Association for Computational Linguistics, 2014. URL http://aclweb.org/anthology/W14-1202.

Grigonyte G., Kvist M., Velupillai S., and Wirén M. Improving readability of Swedish electronic health records through lexical simplification: First results. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 74–83, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1209.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I. H. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. URL http://doi.acm.org/10.1145/1656274.1656278.

Hart M. History and philosophy of project Gutenberg. *Project Gutenberg*, 1992.

Hervás R., Bautista S., Rodríguez M., de Salas T., Vargas A., and Gervás P. Integration of lexical and syntactic simplification capabilities in a text editor. *Procedia Computer Science*, 27(0):94–103, 2014. 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion, DSAI.

Hoard J. E., Wojcik R., and Holzhauser K. An automated grammar and style checker for writers of simplified English. In *Computers and Writing*, pages 278–296. Springer Netherlands, 1992. URL http://dx.doi.org/10.1007/978-94-011-2854-4_19.

Horn C., Manduca C., and Kauchak D. Learning a lexical simplifier using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P14/P14-2075.

Howe T., Worall L., and Hickson L. What is an aphasia-friendly environment? *Aphasiology*, 18:1015 – 1037, 2004.

Hwang W., Hajishirzi H., Ostendorf M., and Wu W. Aligning sentences from standard Wikipedia to simple Wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1022.

Inui K., Fujita A., Takahashi T., Iida R., and Iwakura T. Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 9–16, Sapporo, Japan, July 2003. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W03-1602.

Jauhar S. K. and Specia L. 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 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S12-1066.

Johannsen A., Martínez H., Klerke S., and Søgaard A. EMNLP@CPH: Is frequency all there is to simplicity? In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 408–412, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S12-1054.

John G. H. and Langley P. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.

Jolliffe I. T. *Principal Component Analysis*. Springer, 1986.

Jonnalagadda S. and Gonzalez G. Sentence simplification aids protein-protein interaction extraction. In *The 3rd International Symposium on Languages in Biology and Medicine*, 2009.

Kajiwara T., Matsumoto H., and Yamamoto K. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and*

Speech Processing (ROCLING 2013), pages 59–73. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), 2013.

Kandula S., Curtis D., and Zeng-Treitler Q. A semantic and syntactic text simplification tool for health content. *AMIA Annu Symp Proc*, pages 366–370, 2010.

Kaplan E., Goodglass H., Weintraub S., and Goodglass H. *Boston naming test*. Lea & Febiger, 1983.

Kauchak D. 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 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P13-1151`.

Kertesz A. Neurobiological aspects of recovery from aphasia in stroke. *Disabil Rehabil*, 6(3):122–127, 1984. URL `http://dx.doi.org/10.3109/03790798409165934`.

Keskisärkkä R. *Automatic Text Simplification via Synonym Replacement*. PhD thesis, Linköping, 2012.

Kilgarriff A. and Yallop C. What's in a thesaurus. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, pages 1371–1379, Athens, Greece., 2000. European Language Resources Association (ELRA).

Kincaid P., Fishburne R., Rogers R., and Chissom B. Derivation of new readability formulas (Automated Readability Index, Fog Count, and Flesch Reading Ease Formula) for Navy enlisted personnel. *Research Branch report*, 1975.

Klerke S. and Søgaard A. DSim, a Danish parallel corpus for text simplification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Kučera H. and Francis W. N. *Computational analysis of present-day American English*. Brown University Press, Providence, RI, 1967.

Lal P. and Rüger S. Extract-based summarization with simplification. In *Proceedings of the Workshop on Text Summarization*, pages 90–96, July 2002.

Lasecki W. S., Rello L., and Bigham J. P. Measuring text simplification with the crowd. In *Proceedings of the 12th Web for All Conference*, W4A '15, pages 4:1–4:9, New York, NY, USA, 2015. ACM. URL http://doi.acm.org/10.1145/2745555.2746658.

Leroy G. and Kauchak D. The effect of word familiarity on actual and perceived text difficulty. *Journal of the American Medical Informatics Association*, 21(e1):e169–e172, 2014.

Leroy G., Endicott J. E., Kauchak D., Mouradi O., and Just M. 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, 2013a.

Leroy G., Kauchak D., and Mouradi O. A user-study measuring the effects of lexical simplification and coherence enhancement on perceived and actual text difficulty. *International Journal of Medical Informatics*, 82(8):717–730, 2013b. doi: 10.1016/j.ijmedinf.2013.03.001.

Lesk M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM. URL http://doi.acm.org/10.1145/318723.318728.

Ligozat A.-L., Grouin C., Garcia-Fernandez A., and Bernhard D. Annlor: A naïve notation-system for lexical outputs ranking. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 487–492, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S12-1068.

Ligozat A.-L., Grouin C., Garcia-Fernandez A., and Bernhard D. Approches à base de fréquences pour la simplification lexicale. In *Actes de TALN'2013 : 20e conférence sur le Traitement Automatique des Langues Naturelles*, volume 1, pages 493–506,

Les Sables d'Olonne, France, 2013. URL http://hal.archives-ouvertes.fr/hal-00838354.

Max A. Writing for language-impaired readers. In Gelbukh A., editor, *Computational Linguistics and Intelligent Text Processing*, volume 3878 of *Lecture Notes in Computer Science*, pages 567–570. Springer Berlin Heidelberg, 2006. URL http://dx.doi.org/10.1007/11671299_59.

McKevitt C., Fudge N., Redfern J., Sheldenkar A., Crichton S., and Wolfe C. UK stroke survivor needs survey. Technical report, The Stroke Association, 2010.

McLaughlin H. Smog grading – a new readability formula. *Journal of Reading*, May 1969.

Medero J. *Automatic Characterization of Text Difficulty*. PhD thesis, University of Washington, 2014.

Mihalcea R. and Csomai A. SenseLearner: Word sense disambiguation for all words in unrestricted text. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 53–56, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P05/P05-3014.

Miles A. and Bechhofer S. *SKOS Simple Knowledge Organization System Reference*. W3C, 2009. W3C Recommendation.

Nair S. *The Knowledge Structure in Amarakośa*. PhD thesis, University of Hyderabad, 2011.

Napoles C. and Dredze M. Learning simple Wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, pages 42–50, Los Angeles, CA, USA, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-0406.

Navigli R. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, February 2009. URL http://doi.acm.org/10.1145/1459352.1459355.

Navigli R. and Ponzetto S. P. Multilingual WSD with just a few lines of code: the BabelNet API. In *Proceedings of the ACL 2012 System Demonstrations*, pages 67–72, Jeju Island, Korea, July 2012a. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P12-3012.

Navigli R. and Ponzetto S. P. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012b.

Navigli R., Litkowski K. C., and Hargraves O. SemEval-2007 Task 07: Coarse-grained English all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S/S07/S07-1006.

Nelken R. and Shieber S. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proceedings of EACL 2006, the 11th Conference of the European Chapter of the ACL*, pages 3–7, Trento, Italy, April 2006.

Nunes B. P., Kawase R., Siehndel P., Casanova M., and Dietze S. As simple as it gets-a sentence simplifier for different learning levels and contexts. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 128–132, July 2013. URL http://dx.doi.org/10.1109/ICALT.2013.42.

Otsu N. A threshold selection method from gray-level histograms. *Automatica*, 11 (285-296):23–27, 1975.

Paetzold G. Reliable lexical simplification for non-native speakers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 9–16, Denver, Colorado, June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-2002.

Papineni K., Roukos S., Ward T., and Zhu W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P02-1040.

Pearce D. 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, 2001.

Pedersen P. M., Stig Jørgensen H., Nakayama H., Raaschou H. O., and Olsen T. S. Aphasia in acute stroke: Incidence, determinants, and recovery. *Annals of Neurology*, 38(4):659–666, 1995. ISSN 1531-8249. URL http://dx.doi.org/10.1002/ana.410380416.

Pedersen T. and Kolhatkar V. 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*, NAACL-Demonstrations '09, pages 17–20, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1620959.1620964.

Peng Y., Tudor C., Torii M., Wu C., and Shanker V. isimp: A sentence simplification system for biomedicail text. In *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on*, pages 1–6, Oct 2012. URL http://dx.doi.org/10.1109/BIBM.2012.6392671.

Petersen S. and Ostendorf M. Text simplification for language learners: A corpus analysis. In *Proceedings of the Workshop on Speech and Language Technology in Education (SLaTE)*, page 6972, Pennsylvania, USA, 2007.

Pinker S. *The Sense of Style: The Thinking Person's Guide to Writing in the 21st Century*. Penguin, New York, 2014.

Platt J. Fast training of support vector machines using sequential minimal optimization. In Schölkopf B., Burges C. J. C., and Smola A. J., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998. URL http://research.microsoft.com/~jplatt/smo.html.

Quinlan R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

Rello L. *DysWebxia A Text Accessibility Model for People with Dyslexia*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2014.

Rello L., Baeza-Yates R., Bott S., and Saggion H. Simplify or help?: text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, W4A '13, pages 15:1–15:10, New

York, NY, USA, 2013a. ACM. URL http://dx.doi.org/10.1145/2461121.2461126.

Rello L., Baeza-Yates R., Dempere-Marco L., and Saggion H. Frequent words improve readability and short words improve understandability for people with dyslexia. In Kotz P., Marsden G., Lindgaard G., Wesson J., and Winckler M., editors, *Human-Computer Interaction INTERACT 2013*, volume 8120 of *Lecture Notes in Computer Science*, pages 203–219. Springer Berlin Heidelberg, 2013b. URL http://dx.doi.org/10.1007/978-3-642-40498-6_15.

Rello L., Carlini R., Baeza-Yates R., and Bigham J. P. A plug-in to aid online reading in Spanish. In *Proceedings of the 12th Web for All Conference*, W4A '15, pages 7:1–7:4, New York, NY, USA, 2015. ACM. URL http://doi.acm.org/10.1145/2745555.2746661.

Rennes E. and Jönsson A. A tool for automatic simplification of Swedish texts. In *Proceedings of the Nordic Conference of Computational Linguistics (NODALIDA)*, pages 317–320, 2015.

Roget P. M. *Roget's Thesaurus of English words and phrases*. Available from Project Gutemberg, Illinois Benedectine College, Lisle IL (USA), 1852.

Sabou M., Bontcheva K., Derczynski L., and Scharl A. Corpus annotation through crowdsourcing: Towards best practice guidelines. In Calzolari N., Choukri K., Declerck T., Loftsson H., Maegaard B., Mariani J., Moreno A., Odijk J., and Piperidis S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 859–866, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/497_Paper.pdf.

Saggion H., Martínez E. G., Etayo E., Anula A., and Bourg L. Text simplification in simplext. making text more accessible. *Procesamiento del lenguaje natural*, 47: 341–342, 2011.

Saggion H., Bott S., and Rello L. Comparing resources for Spanish lexical simplification. In Dediu A.-H., Martín-Vide C., Mitkov R., and Truthe B., editors, *Statistical Language and Speech Processing*, volume 7978 of *Lecture Notes in Computer Science*, pages 236–247. Springer, Berlin Heidelberg, 2013.

Saggion H., Štajner S., Bott S., Mille S., Rello L., and Drndarević B. Making it simplext: Implementation and evaluation of a text simplification system for spanish. *ACM Trans. Access. Comput.*, 6(4):14:1–14:36, May 2015. ISSN 1936-7228. doi: 10.1145/2738046. URL http://doi.acm.org/10.1145/2738046.

Salton G. and Yang C.-S. On the specification of term values in automatic indexing. *Journal of documentation*, 29(4):351–372, 1973.

Scarton C., de Oliveira M., Candido A., Jr., Gasperin C., and Aluísio S. M. Simplifica: a tool for authoring simplified texts in Brazilian Portuguese guided by readability assessments. In *Proceedings of the NAACL HLT Demonstration Session*, HLT-DEMO '10, pages 41–44, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1855450.1855461.

Shardlow M. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications, Special Issue on Natural Language Processing*, 2014.

Siddharthan A. Syntactic simplification and text cohesion. *Research on Language & Computation*, 4:77–109, 2006.

Siddharthan A. A survey of research on text simplification. *the International Journal of Applied Linguistics*, pages 259–98, 2014.

Siddharthan A. and Mandya A. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731. Association for Computational Linguistics, 2014.

Siddharthan A., Nenkova A., and Mckeown K. Syntactic Simplification for Improving Content Selection in Multi-Document Summarization. In *In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004*, pages 896–902, 2004. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.6424.

Sinha R. Unt-simprank: Systems for lexical simplification ranking. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1:*

Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), pages 493–496, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S12-1069.

Snyder B. and Palmer M. The English all-words task. In Mihalcea R. and Edmonds P., editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Specia L., Jauhar S. K., and Mihalcea R. 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 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/S12-1046.

Taylor W. L. Cloze procedure: a new tool for measuring readability. *Journalism Quarterly*, pages 415–433, 1953.

The Stroke Association. Accessible information guidelines: Making information accessible for people with aphasia, 2012.

Thomas S. R. and Anderson S. WordNet-based lexical simplification of a document. In Jancsary J., editor, *Proceedings of KONVENS 2012*, pages 80–88. ÖGAI, September 2012. URL http://www.oegai.at/konvens2012/proceedings/13_thomas12o/.

Vajjala S. and Meurers D. Assessing the relative reading level of sentence pairs for text simplification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 288–297. Association for Computational Linguistics, 2014. URL http://aclweb.org/anthology/E14-1031.

Van Pelt C. and Sorokin A. Designing a scalable crowdsourcing platform. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 765–766, New York, NY, USA, 2012. ACM. URL http://doi.acm.org/10.1145/2213836.2213951.

Vickrey D., Koller D., and Wants T. Applying sentence simplification to the CoNLL-2008 shared task, 2008.

Štajner S. and Saggion H. Readability indices for automatic evaluation of text simplification systems: A feasibility study for Spanish. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 374–382. Asian Federation of Natural Language Processing, 2013a. URL http://aclweb.org/anthology/I13-1043.

Štajner S. and Saggion H. Adapting text simplification decisions to different text genres and target users. *Procesamiento del Lenguaje Natural*, 51(0):135–142, 2013b. ISSN 1989-7553.

Vu T. T., Tran G. B., and Pham S. B. Learning to simplify children stories with limited data. In Nguyen N. T., Attachoo B., Trawiński B., and Somboonviwat K., editors, *Intelligent Information and Database Systems*, volume 8397 of *Lecture Notes in Computer Science*, pages 31–41. Springer International Publishing, 2014.

Ward G. Moby thesaurus, 1996.

Watanabe W. M., Candido A., Jr., Amâncio M. A., de Oliveira M., Pardo T. A. S., Fortes R. P. M., and Aluísio S. M. Adapting web content for low-literacy readers by using lexical elaboration and named entities labeling. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, W4A '10, pages 8:1–8:9, New York, NY, USA, 2010. ACM. URL http://doi.acm.org/10.1145/1805986.1805998.

Watanabe W. M., Candido A., Jr., Uzêda V. R., de Mattos Fortes R., Pardo T. A. S., and Aluísio S. M. 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, 2009. ACM. URL http://doi.acm.org/10.1145/1621995.1622002.

Wilcoxon F. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6): pp. 80–83, 1945.

Wilkens R., Vecchia A. D., Boito M. Z., Padr M., and Villavicencio A. Size does not matter. frequency does. a study of features for measuring lexical complexity. In Bazzan A. L. and Pichara K., editors, *Advances in Artificial Intelligence – IBERAMIA 2014*, Lecture Notes in Computer Science, pages 129–140. Springer International Publishing, 2014. URL http://dx.doi.org/10.1007/978-3-319-12027-0_11.

Woodsend K. and Lapata M. 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. URL `http://dl.acm.org/citation.cfm?id=2145432.2145480`.

Xu W., Callison-Burch C., and Napoles C. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297, 2015. URL `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/549`.

Yatskar M., Pang B., Danescu-Niculescu-Mizil C., and Lee L. 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, 2010. Association for Computational Linguistics.

Young D. J. Linguistic simplification of SL reading materials: Effective instructional practice? *Modern Language Journal*, 1999.

Zeng Q., Kim E., Crowell J., and Tse T. A text corpora-based estimation of the familiarity of health terminology. In Oliveira J. L., Maojo V., Martín-Sánchez F., and Pereira A. S., editors, *Biological and Medical Data Analysis*, volume 3745 of *Lecture Notes in Computer Science*, pages 184–192. Springer Berlin Heidelberg, 2005.

Zeng-Treitler Q., Goryachev S., Tse T., Keselman A., and Boxwala A. Estimating consumer familiarity with health terminology: a context-based approach. *Journal of the American Medical Informatics Association*, 15:349–356, 2008.

Zhu Z., Bernhard D., and Gurevych I. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1353–1361, 2010.