# Sentence Simplification for Text Processing

Richard Evans

2020

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.



Signature: . . . . . .

Date: 3rd May 2020 . . . . . .

# Abstract

Propositional density and syntactic complexity are two features of sentences which affect the ability of humans and machines to process them effectively. In this thesis, I present a new approach to automatic sentence simplification which processes sentences containing compound clauses and complex noun phrases (NPs) and converts them into sequences of simple sentences which contain fewer of these constituents and have reduced *per sentence* propositional density and syntactic complexity.

My overall approach is iterative and relies on both machine learning and handcrafted rules. It implements a small set of sentence transformation schemes, each of which takes one sentence containing compound clauses or complex NPs and converts it one or two simplified sentences containing fewer of these constituents (Chapter 5). The iterative algorithm applies the schemes repeatedly and is able to simplify sentences which contain arbitrary numbers of compound clauses and complex NPs. The transformation schemes rely on automatic detection of these constituents, which may take a variety of forms in input sentences. In the thesis, I present two new shallow syntactic analysis methods which facilitate the detection process.

The first of these identifies various explicit signs of syntactic complexity in input sentences and classifies them according to their specific syntactic linking

and bounding functions. I present the annotated resources used to train and evaluate this sign tagger (Chapter 2) and the machine learning method used to implement it (Chapter 3). The second syntactic analysis method exploits the sign tagger and identifies the spans of compound clauses and complex NPs in input sentences. In Chapter 4 of the thesis, I describe the development and evaluation of a machine learning approach performing this task. This chapter also presents a new annotated dataset supporting this activity.

In the thesis, I present two implementations of my approach to sentence simplification. One of these exploits handcrafted rule activation patterns to detect different parts of input sentences which are relevant to the simplification process. The other implementation uses my machine learning method to identify compound clauses and complex NPs for this purpose.

Intrinsic evaluation of the two implementations is presented in Chapter 6 together with a comparison of their performance with several baseline systems. The evaluation includes comparisons of system output with human-produced simplifications, automated estimations of the readability of system output, and surveys of human opinions on the grammaticality, accessibility, and meaning of automatically produced simplifications.

Chapter 7 presents extrinsic evaluation of the sentence simplification method exploiting handcrafted rule activation patterns. The extrinsic evaluation involves three NLP tasks: multidocument summarisation, semantic role labelling, and information extraction. Finally, in Chapter 8, conclusions are drawn and directions for future research considered.

# Contents

# List of Tables

# List of Figures

# Acknowledgements

I gratefully acknowledge current and former staff members at the Research Group in Computational Linguistics at the University of Wolverhampton. The research described in this thesis would not have been possible without them:

- Dr. Iustin Dornescu, who implemented the improved sign tagger used in many of the experiments described in the thesis.
- Dr. Le An Ha for encouraging me under his supervision in the early stages of this research. My work since 2009 would have been very different without him.
- Prof. Ruslan Mitkov for his support and his leadership of the research institute, without which this dissertation could not have been written.
- Dr. Constantin Orasan for his careful supervision and contribution in improving the standard of research reported here.

The work described in this thesis also depended on the availability of numerous annotated corpora and datasets. I gratefully acknowledge the contributions of the annotators who helped to create them:

- **Compound clauses and complex NPs** (cross-annotation): Victoria Yaneva;[1]

[1] Never *very* cross annotation.

- **Information extraction templates**: Le An Ha;
- **Semantic role labels** (pilot): Sabi Bulent and Simona Ignatova;
- **Signs of syntactic complexity**: Emma Franklin and Zoë Harrison;
- **Simplified sentences**: Emma Franklin and Laura Hasler

I also gratefully acknowledge the contribution of the respondents to the opinion surveys reported on in Chapter 6: Martina Cotella, Francesca Della Moretta, Ariana Fabbri, and Victoria Yaneva. Thanks are due to Larissa Sayuri for assistance provided in the collation of this survey data, which was initially distributed over numerous inconsistently formatted Google spreadsheets.

My involvement in sentence simplification began in 2008 when I was analysing errors made by an information extraction system designed to identify important facts expressed in clinical notes. Many of the errors made by the system were caused by its inability to correctly identify the relations holding between concepts mentioned in long complex sentences. I began by writing a large number of information extraction patterns to handle different sequences of relations and concepts mentioned in those sentences. Eventually, I realised that a systematic approach to sentence simplification was needed. I'm grateful to Le An Ha, my supervisor at that time, for allowing me time in the project to develop an initial solution to address this challenge.

Parts of the research presented in this thesis were funded by:

1. the National Board of Medical Examiners for the CAID project, which ran

in 2008.

2. the European Commission for the FIRST project, which ran from 2011-2014 (Grant number: FIRST.ICT.2013.608257).

Back in something like 1978, my dad brought home from work the first computer I ever saw, a *Commodore Pet*. My mum used to sit with me or Phil in front of the family *BBC Model B* to help type in computer programs from INPUT *Magazine*, like some new electronic form of knitting. I thank them both for this introduction to computer programming.

Since 1985 and my childhood encounters with Magnetic Scrolls's text-based adventure game *The Pawn* on our family's *Commodore Amiga*, I've been interested in syntactic parsing.[2] This introduction to the concept of syntax and its role in computer games motivated me to learn more about it at college and then as an undergrad and postgrad. Without this initial interest, I think I would have taken a very different approach to the problem of sentence simplification for text processing, if I'd ever encountered the problem at all.

★

[2] *The Pawn* was revolutionary because if you typed "I think therefore I am" the computer would say `Oh, do you?` According to *Computer and Video Games* magazine, this meant that it had a good parser. All the other text-based adventure games of the time said `I don't know how to do that.`

# Chapter 1

# Introduction

The linguistic complexity of a text can adversely affect subsequent text processing. Text simplification is the process of reducing the linguistic complexity of a text, while retaining, as far as possible, the original information content and meaning.[1] Text simplification systems may include components for adapting text by means of various lexical (Devlin and Tait, 1998; Zeng-Treitler *et al.*, 2007; De Belder *et al.*, 2010; Kandula *et al.*, 2010; Yatskar *et al.*, 2010; Biran *et al.*, 2011; Walker *et al.*, 2011; Bott *et al.*, 2012b; Specia *et al.*, 2012), syntactic (Chandrasekar and Srinivas, 1997; Canning, 2002; Siddharthan, 2006; Cohn and Lapata, 2009), and other (Specia, 2010; Coster and Kauchak, 2011; Wubben *et al.*, 2012) transformation operations and components for the generation of assistive content such as definitions (Elhadad, 2006), images (Bosma, 2005; Barbu *et al.*, 2015), and summaries (Barbu *et al.*, 2015).

The ongoing development and democratisation of the World Wide Web has brought increased demand for widely accessible information. Systems implementing text simplification have been developed to improve the accessibility of textual data for various populations, including people with poor literacy (Candido *et al.*, 2009; Siddharthan, 2011) or numeracy (Bautista and Saggion, 2014), people with

[1] Definition adapted from that of Siddharthan (2014).

aphasia (Max, 2000), dyslexia (Rello *et al.*, 2013), autism (Dornescu *et al.*, 2013; Evans *et al.*, 2014; Orăsan *et al.*, 2018), or other cognitive impairments (Bott *et al.*, 2012b) or reading disabilities (Glavas and Stajner, 2013), people with hearing loss (Inui *et al.*, 2003), people who are non-native speakers (Amoia and Romanelli, 2012; Angrosh and Siddharthan, 2014; Paetzold, 2015), and children and language learners (Kajiwara and Yamamoto, 2015).

Automatic sentence simplification is one aspect of text simplification, a topic that has been addressed in several lines of research since the 1990s. Numerous methods for sentence simplification have been developed, including rule-based approaches (Siddharthan, 2006; Evans, 2011) and data-driven methods exploiting machine learning (Yatskar *et al.*, 2010; Coster and Kauchak, 2011; Siddharthan, 2014) and deep learning (Klerke *et al.*, 2015; Zhang and Lapata, 2017; Vu *et al.*, 2018; Shardlow and Nawaz, 2019). These methods have been used to facilitate various language processing tasks, including human text comprehension (Max, 2000; Canning, 2002; Scarton *et al.*, 2017; Orăsan *et al.*, 2018) and automatic NLP applications such as information extraction (Evans, 2011; Niklaus *et al.*, 2016) and semantic role labelling (Vickrey and Koller, 2008). In this thesis, I present a detailed survey of previous work on sentence simplification in Chapter 5 (Section 5.1).

My thesis is concerned with the development of an automatic sentence simplification tool that performs syntactic transformation operations. Specifically, the tool is intended to detect the compound clauses and complex NPs modified by nominally bound non-restrictive finite relative clauses in an input sentence and

to rewrite it as a sequence of sentences, each of which contains fewer clauses. For brevity, the type of complex NPs to be simplified in this work will be referred to as complex$_{\overline{R}F}$ NPs.[2]

The work described here draws on my previous work on the development of a method for sentence simplification for use in biomedical information extraction (Evans, 2011), on the development of a corpus annotated with information about the linking and bounding functions of explicit signs of syntactic complexity (Evans and Orasan, 2013), and on the development of an automatic method to classify such signs (Dornescu *et al.*, 2013).

As noted by Siddharthan (2006), text simplification can be viewed as comprising three processes: analysis, transformation, and post-editing. Evans (2011) presented a rule-based method for sentence simplification that is based on a shallow sentence analysis step and an iterative sentence transformation step. The main contributions of that method were a new approach for automatic sentence analysis and a method for simplifying sentences on the basis of that analysis. The analysis step includes:

1. tokenisation of input texts to enable identification of sentences, words, and a pre-specified set of textual markers of syntactic complexity, referred to as *potential coordinators*,[3]

2. part of speech tagging, and

[2]From *non-restrictive* ($\overline{R}$) and *finite* (F). In the case of $\overline{R}$, I appropriate the notation used in set theory to indicate negation, in this case, of restrictive (R) modifiers.

[3]Comprising commas, conjunctions, and adjacent comma-conjunction pairs, these potential coordinators comprise a subset of the signs of syntactic complexity discussed in Chapter 2 of this thesis.

3. a ML method to categorise potential coordinators.

The classification scheme used in that work provides detailed information about a wide range of clausal and subclausal types of coordination but offers limited information about different types of subordination. As a result, the classifier based on that scheme provides only a limited analysis of sentences containing the types of syntactic complexity prevalent in texts of the registers of news and literature. The sentence simplification method that exploits Evans's (2011) analysis step is unlikely to adequately process texts of these registers. Despite this, his approach proved to be useful in a biomedical information extraction task and compared favourably with an approach based on full syntactic analysis using the Stanford parser. This approach serves as a starting point for the research presented in my thesis.

The method for sentence simplification that I present in this thesis differs from that of Evans (2011) by nature of the fact that Evans's system was designed to process text of a restricted type (clinical vignettes), containing a more restricted range of syntactic structures. For simplification of sentences containing compound clauses, the sentence transformation rule set used in Evans's (2011) system comprised just four rules. Lacking information about many subordinate clause boundaries, his system is unable to simplify sentences containing the types of syntactic complexity that are common in texts of other registers, such as news and literature. It is incapable of simplifying sentences containing finite subordinate clauses. By contrast, the system that I present in Chapter 5 of this thesis is able to simplify sentences containing a wider range of syntactic structures and

was developed for use with texts of multiple registers. In terms of evaluation, the output produced by Evans's (2011) system was not assessed intrinsically or with respect to grammatical correctness, readability, or meaning. It was not evaluated extrinsically via a range of NLP applications. In this thesis, I use these methods and these criteria to evaluate the output of my system and I compare its performance with that of two baseline systems.

Having provided information on the ways in which the work in this thesis goes beyond Evans's (2011) work, in Section 5.1, I also describe the ways in which the methods that I present differ from those presented in two major lines of research on the topic of sentence simplification. These include rule-based and data-driven approaches exploiting a range of NLP tools and resources such as syntactic parsers, plain and syntactically parsed parallel corpora, and methods exploiting deep parsing and semantic analysis.

## 1.1 My Notion of Sentence Difficulty

The main motivation for text simplification is to facilitate subsequent text processing. In this thesis, I use the term *text processing* to denote cognitive processing by humans when reading and automatic processing by NLP applications in tasks such as syntactic parsing, information extraction (IE), and machine translation (MT).

The speed and accuracy of human reading comprehension depends on the linguistic complexity of the text being read. This claim is supported by evidence from eye-tracking (Rayner *et al.*, 2006; Wendt *et al.*, 2014), auditory moving win-

dows (Ferreira *et al.*, 1996), self-paced reading (King and Just, 1991; MacDonald *et al.*, 1992; Caplan and Waters, 1999), act-out procedure (Tager-Flusberg, 1981), picture-matching (Kover *et al.*, 2012), and rapid serial visual presentation (Waters and Caplan, 1996) experiments. The two types of linguistic complexity most relevant to my clause-focused method for automatic sentence simplification are syntactic complexity and propositional density.

Many previous studies of the relationship between linguistic complexity and reading comprehension take *syntactic complexity* as the linguistic variable of interest. Syntactic complexity indicates the difficulty with which human readers can assign a syntactic structure to a sentence and can use that structure to determine its meaning (Caramazza and Zurif, 1976; Norman *et al.*, 1991; Just *et al.*, 1996; Meltzer *et al.*, 2009). Syntactic complexity is one factor that increases the difficulty of working out, according to the sentence, who did what to whom in an event. The most syntactically complex constructions in English are garden-path sentences (1)

(1) a. *The horse raced past the barn fell.*

b. *The experienced soldiers warned about the dangers conducted the midnight raid.*

and object-relativised clauses (2).[4]

[4]Throughout this thesis, when linguistic examples are provided, indented enumerated examples presented in standard font face are extracted from the corpus described in Chapter 2. Indented enumerated examples presented in italics are either invented or appropriated from previous related work. Examples judged to be unnatural by native speakers are preceded by an asterisk.

(2) a. *The man the woman the child hugged kissed laughed.*

b. *Oysters oysters oysters split split split.*

In *garden path sentences*, incremental processing of words in the sentence forces the reader to postulate two or more candidate hypotheses of its structure. By default, readers prefer hypotheses that obey the principles of *late closure*[5] and *minimal attachment*[6] (Frazier and Rayner, 1982). In (1-a), the principle of minimal attachment favours the hypothesis that the temporarily ambiguous noun phrase *the barn* is the simple direct object of the verb *raced* (Fig. 1.1), rather than being the object of the relative clause modifying the matrix subject (Fig. 1.2), since the former analysis requires the postulation of fewer nodes in the sytactic structure.[7] On processing subsequent words in the sentence, readers must backtrack, discarding this hypothesis and accepting the less favourable one. Given that processing of garden path sentences requires access to non-deterministic syntactic parsers and that this type of complexity is not usually signalled by explicit textual signs, I consider the development of methods to automatically simplify garden path sentences to be beyond the scope of this thesis.

In sentences containing *object-relativised clauses*, readers may have insufficient working memory to retain the intermediate products of computation that are produced when deriving their complex syntactic structure (Caplan and Wa-

[5]In which incoming lexical items are attached into the clause or phrase currently being processed (i.e. the lowest possible nonterminal node dominating the last item analysed).

[6]When possible, attach incoming material into the phrase-marker being constructed using the fewest nodes consistent with the well-formedness rules of the language.

[7]In this formalism, 9 non-terminal nodes before the stranded verb *fell* is analysed vs. 11 non-terminal nodes for the parse motivated by the late closure principle.

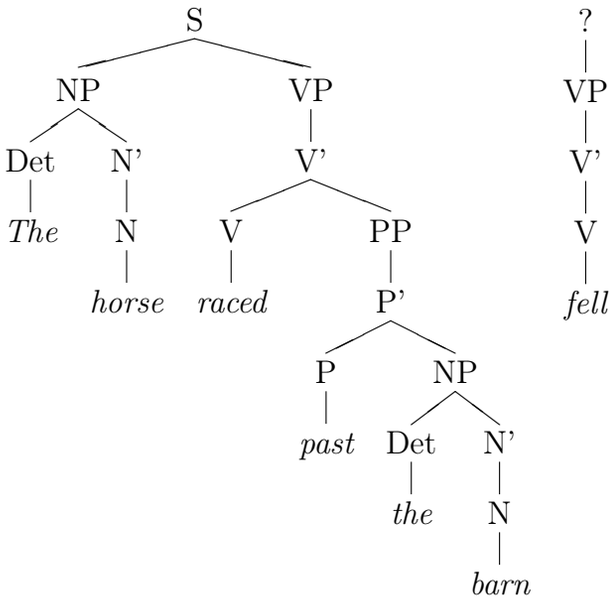



Figure 1.1: *Hypothesis motivated by the principle of minimal attachment*

ters, 1999). Psycholinguistic approaches to syntactic processing based on working memory propose that the storage of temporally incomplete head-dependencies in phrase structure (such as the dependencies between each of the noun phrases in Sentence (2-a) and the verbs to be encountered later in the sentence, for which they are objects) may exceed readers' working memory capacity (Gibson and Thomas, 1996; Gibson, 1998). The situation is further exacerbated by the fact that the noun phrases being held simultaneously in working memory are syntactically and semantically similar, causing interference effects that may adversely affect sentence comprehension (Gordon *et al.*, 2001). The method presented in this thesis is designed to simplify sentences on the basis of the various explicit signs of syntactic complexity occurring in them (Chapter 2). This includes sentences containing object-relativised clauses.

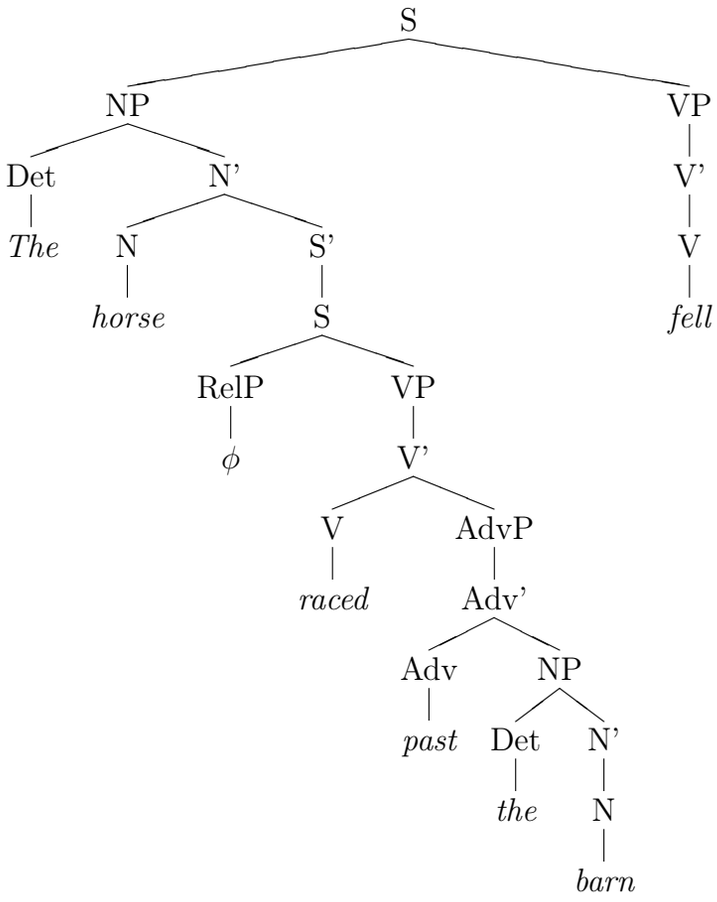



Figure 1.2: *Hypothesis motivated by the principle of late closure*

Having considered syntactic complexity, the second relevant type of linguistic complexity is *propositional density*: the average number of propositions conveyed by each sentence of the text. Propositions are atomic statements that express simple factual claims (Jay, 2003). They are considered the basic units involved in the understanding and retention of text (Kintsch and Welsch, 1991). To illustrate, Sentence (3)

(3) *A series of violent, bloody encounters between police and Black Panther Party members punctuated the early summer days of 1969.*

contains seven propositions:[8]

1. (SERIES, ENCOUNTER)
2. (VIOLENT, ENCOUNTER)
3. (BLOODY, ENCOUNTER)
4. (BETWEEN, ENCOUNTER, POLICE, BLACK PANTHER)
5. (TIME : IN, ENCOUNTER, SUMMER)
6. (EARLY, SUMMER)
7. (TIME : IN, SUMMER, 1969)

Caplan and Waters (1999) report that the findings of many psycholinguistic experiments in sentence comprehension can be explained by reference to the number of propositions conveyed by the sentences presented to readers. In short, the greater the number of propositions expressed in a sentence, the more difficult it is for readers to perform concurrent memory tasks such as memorising information presented in previous sentences or memorising sequences of numbers. This interference, known as the *number-of-propositions effect*, affects all readers and is particularly strong for people with reduced working memory capacity (Caplan and Waters, 1999). Reducing the propositional density of input sentences can reduce the working memory resources needed when reading them, freeing up those resources for other concurrent memory tasks required in text comprehension.

[8]This example is taken from Kintsch and van Dijk (1978). While propositions may be expressed using different formalisms, this is the one used by the authors in their article.

Kintsch and van Dijk (1978) note that the automatic derivation of propositions from sentences is an open research problem and one that is unlikely to be solved in the near future.[9] However, various automatic methods for estimating the propositional density of text have been proposed. These include the metrics integrated within the *CPIDR* (Covington, 2012) and *Coh-Metrix* (McNamara *et al.*, 2014) readability assessment tools. Those tools estimate propositional density to be the ratio of the number of verbs, adjectives, adverbs, prepositions, and conjunctions to the total number of words in the sentence (Brown *et al.*, 2008; DeFrancesco and Perkins, 2012). The new approach to sentence simplification that I present in this thesis does not operate on the underlying logical representation of input sentences and does not simplify their propositional structure and reduce their propositional density directly. However, syntactic transformation operations applied by my method will reduce the numbers of words of the aforementioned categories in the output sentences generated. By definition, a sentence simplification method that reduces the number of words of these categories in a sentence reduces the estimated propositional density of the sentence.

In many cases, the propositional density of a sentence is proportional to its length. Previous related work has shown that the accuracy of NLP applications such as syntactic parsing (Tomita, 1985; McDonald and Nivre, 2011), IE (Evans, 2011), and MT (Gerber and Hovy, 1998) is inversely proportional to the length of the sentences being processed. Muszyńska (2016) notes that:

[9] Of the grammar on which such derivations would depend, they state that "no such grammar is available now, nor is there hope for one in the near future" (Kintsch and van Dijk, 1978).

> Some approaches to parsing have space and time requirements which are much worse than linear in sentence length. This can lead to practical difficulties in processing. For example, the ACE processor running the English Resource Grammar (ERG) (Copestake and Flickinger, 2000) requires roughly 530 MB of RAM to parse sentence (4). In fact, longer and more complicated sentences can cause the parser to time out or run out of memory before a solution is found.

(4) *Marcellina has hired Bartolo as her counsel, since Figaro had once promised to marry her if he should default on a loan she had made to him, and she intends to enforce that promise.*

When split into four shorter sentences, Muszyńska notes that each of the shorter sentences can be parsed with less than 20 MB, "requiring in total less than a fifth of the RAM needed to parse the full sentence." This observation provides further motivation for the development of an automatic sentence simplification tool that will reduce the propositional density of input sentences.

Syntactically complex sentences containing clause compounds and nominally bound relative clauses convey relatively large numbers of propositions and have a large propositional density. As a result they can adversely affect the speed and accuracy of syntactic processing of a wide range of readers. In this thesis, when discussing sentence simplification and its evaluation, I refer to sentences containing compound clauses as *Type 1 sentences* and sentences containing finite

non-restrictive nominally bound relative clauses as *Type 2 sentences*. Sentences containing both types of complexity may be referred to using either term, as appropriate in the context.

In this thesis, I consider sentences to be difficult and in need of simplification when they are syntactically complex and have a relatively great propositional density. Throughout this work, and especially in Chapter 4, I use the term *complex constituent* to refer to phrases and clauses of any grammatical category which contain finite subordinate clauses. Complex NPs are one type of complex constituent. The methods that I present are designed to simplify sentences containing complex$_{\overline{RF}}$ NPs and compound clauses. The simplification of sentences containing NPs modified by non-finite relative clauses, including adjectival, prepositional, nominal (appositive), and verbal[10] relative clauses[11] is not the focus of this research.[12]

## 1.2 A New Approach to Automatic Sentence Simplification

The method for sentence simplification that I propose in this thesis (Chapter 5) is based on the automatic identification of various explicit textual markers of syntactic complexity, which I refer to as signs of syntactic complexity, and a sentence transformation step that exploits information about these signs. The

[10] Including past-participial and *-ing* clauses.

[11] The interested reader can find examples of sentences containing non-finite relative clauses in Chapter 4, examples (24), (26), (27), and (28).

[12] A user requirements analysis conducted in previous work to improve text accessibility for people with autism indicated no demand for that type of sentence simplification (Martos *et al.*, 2013).

method is designed to detect the compound clauses and complex$_{\overline{RF}}$ NPs in an input sentence and to rewrite it as a sequence of sentences, each of which contains fewer clauses. For example, the method will convert a sentence such as (5) into the sequence of sentences (6). The propositional density of the text is thus reduced, as is the minimum working memory span required for comprehension of each of its sentences.

(5) Blumenthal, who has three Michelin stars, set the restaurant up with his ex-wife Zanna and the pair turned it into a multi-million pound business.

(6) a. Blumenthal set the restaurant up with his ex-wife Zanna.

b. Blumenthal has three Michelin stars.

c. The pair turned it into a multi-million pound business.

The sentence simplification method thus has the potential to facilitate text processing by addressing one of the two extreme types of syntactic complexity mentioned earlier in this chapter (object-relativised clauses) and by reducing the propositional density of input sentences.[13]

## 1.3 Research Questions

The goal of this thesis is to propose a pipeline for automatic sentence simplification and to assess the usefulness of its output for subsequent text processing. This goal is achieved by addressing five research questions:

[13]The method is not designed to reduce the complexity of garden-path sentences.

**RQ-1** Are there reliable and explicit textual signs which can indicate the occurrence of compound clauses and complex$_{\overline{RF}}$ NPs in English sentences? What are these signs and what are their functions?

This question is addressed in Chapter 2, in which I posit a small set of words and punctuation marks as potential indicators of compound clauses and complex$_{\overline{RF}}$ NPs in the sentences of a text. Information about these signs and their syntactic linking and bounding functions is annotated in a corpus. Manual annotation of the corpus is described and information on the syntactic linking and bounding functions of the signs is provided. The distribution of signs in texts of three registers is discussed, as is the reliability of the annotation.

**RQ-2** Can signs of syntactic complexity be automatically and reliably classified according to their specific syntactic functions?

In Chapter 3, I present the development of a tagger which automatically classifies signs of syntactic complexity in accordance with the annotation

scheme presented in Chapter 2. **RQ-2** is answered by quantitative evaluation of the sign tagger, which is presented in Sections 3.3 and 5.2.4 of the thesis.

**RQ-3** To what extent can an iterative rule-based approach exploiting automatic sign classification and handcrafted patterns convert sentences into a form containing fewer compound clauses and fewer complex$_{\overline{RF}}$ NPs?

My response to **RQ-3** includes the development of the automatic sentence simplification tool described in Chapter 5 and its quantitative evaluation, described in Chapter 6 of the thesis. This approach to sentence simplification integrates the sign tagger presented in Chapter 3 to perform shallow syntactic analysis of input sentences and handcrafted rule activation patterns which exploit this analysis and are used to implement a set of sentence transformation schemes.

Determination of the extent to which this iterative rule-based approach successfully reduces the numbers of compound clauses and complex$_{\overline{RF}}$ NPs in input sentences is made through quantitative evaluation of the system (Chapter 6). My evaluation method is based on a comparison of the output of the sentence simplification tool with simplifications of texts made by linguists aiming to convert the sentences that they contain into a form

containing no compound clauses and no complex$_{\overline{RF}}$ NPs. This comparison is made on the basis of the metrics described in Section 6.1.2. I also use automatic methods to estimate the readability of the output of the system, and I survey human opinions about the grammaticality, comprehensibility, and meaning of this output.

**RQ-4** How does the accuracy of automatic sentence simplification compare when using a machine learning approach to detect the spans of compound clauses and complex$_{\overline{RF}}$ NPs and when using a method based on handcrafted patterns?

Chapter 5 of this thesis presents an iterative approach to sentence simplification which is based on a set of sentence transformation schemes implemented as rules to simplify input sentences. The rules are based on activation patterns which identify different elements of input sentences that are used to generate output sentences. These elements include the conjoins of compound clauses and the finite relative clauses modifying complex$_{\overline{RF}}$ NPs. Chapter 5 presents examples of the handcrafted rule activation patterns used to identify such elements. This comprises the first part of my response to **RQ-4**. In Chapter 4, I present new methods exploiting machine learning (sequence labelling) to automatically identify the spans of compound clauses and complex constituents, including clause conjoins, nomi-

nally bound relative clauses, and the superordinate NPs that they modify. Chapter 4 includes an evaluation of the machine learning methods used. This comprises the second part of my response to **RQ-4**. I conclude my response to this research question in Chapter 6 which includes a comparative evaluation of sentence simplification methods exploiting handcrafted and machine-learned rule activation patterns. This comparative evaluation is made by reference to the similarity of system output to human simplification of input sentences and automatic assessments of the readability of system output.

**RQ-5** Does the automatic sentence simplification method facilitate subsequent text processing?

In this thesis, the term *text processing* is limited to denote only text processing by machines in NLP applications such as MT and IE. Although such studies are very much in scope, in this thesis, I lacked the resources necessary to run reading behaviour studies using methods from cognitive science such as eye tracking and self-paced reading. My response to **RQ-5** is presented in Chapter 7, which evaluates the contribution made by the approach to sentence simplification based on handcrafted rule activation patterns to the NLP applications of multidocument summarisation, semantic role labelling, and information extraction.

## 1.4 Structure of the Thesis

Figure 1.3 provides a schematic overview of the structure of my thesis. Chapters 2 and 3 are concerned with the identification and classification of explicit lexical and punctuational markers of syntactic complexity in English sentences: the signs of syntactic complexity. These chapters present a human annotated corpus containing text of three registers and an automatic classifier derived using a machine learning method applied to this corpus. This sign tagger is exploited by the additional syntactic analysis method and the sentence simplification methods presented in Chapters 4 and 5, respectively.

Chapter 4 presents a new machine learning method to identify the spans of compound clauses and complex constituents in English sentences. It includes a description of the annotated resources developed to support development and training of the method. The chapter includes an evaluation of its accuracy when classifying tokens in input sentences as occurring within compound clauses or occurring within several different types of complex constituents, including complex$_{\overline{RF}}$ NPs.

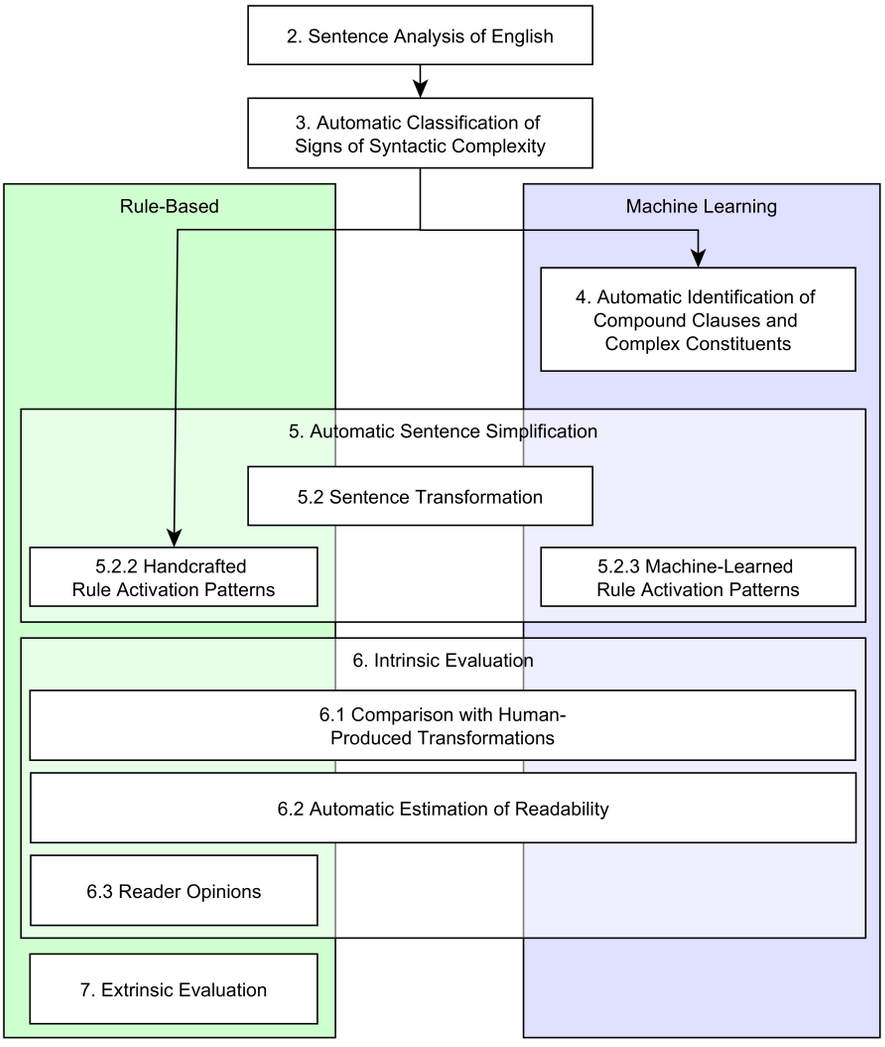
Chapter 5 presents my approach to automatic sentence simplification. This approach is based on several sentence transformation schemes to simplify sentences containing compound clauses and complex$_{\overline{RF}}$ NPs. The schemes are implemented as rules comprising rule activation patterns associated with transformation operations that exploit those patterns. Two systems are presented, one which implements handcrafted rule activation patterns (Section 5.2.2) and one which exploits the machine learning approach presented in Chapter 4 to imple-

ment machine-learned rule activation patterns (Section 5.2.3).

Chapter 6 presents intrinsic evaluation of both sentence simplification systems. This evaluation is made using overlap metrics which compare automatically simplified sentences with human-simplified sentences (Section 6.1) and using automated assessments of the readability of system output (Section 6.2). In Section 6.3, the system exploiting handcrafted rule activation patterns is also evaluated via surveys of the opinions of human readers with respect to the grammaticality, comprehensibility, and meaning of its output. Chapter 7 of the thesis presents extrinsic evaluation of that system. The extrinsic evaluation is made via automatic NLP applications for multidocument summarisation, semantic role labelling, and information extraction.

Chapters 2–5 and Chapter 7 of the thesis each contain surveys of related work. These chapters and Chapter 6 also include sections detailing contributions made to the previously listed research questions. Finally, Chapter 8 synthesises this information and discusses the extent to which the main goal of the thesis was achieved. It includes indications of directions for future work relevant to each of the preceding chapters.

Figure 1.3: *Structure of the thesis*



# Chapter 2

## Sentence Analysis of English

The research described in this chapter addresses research question **RQ-1** of the thesis, which is concerned with the existence of explicit textual signs indicating the occurrence of compound clauses and complex$_{\overline{RF}}$ NPs in English sentences. My response to this research question is one part of the more general task of providing a shallow syntactic analysis of English sentences. In this context, the signs are explicit markers of a potentially wide range of compound constituents and subordinate clauses modifying complex constituents. In this chapter, I specify the set of signs and their syntactic linking and bounding functions. In my research, I supervised development of a corpus annotated with information about these signs of syntactic complexity. I describe and present an analysis of this corpus.

Compound clauses are one type of compound constituent occurring in natural languages, including English. Compound constituents are those that contain two or more syntactic constituents linked by coordination. Quirk *et al.* (1985) define coordination as a paratactic relationship that holds between constituents at the same level of syntactic structure. The linking function occurs between conjoins[1]

[1]In this thesis, I employ the terminology used by Quirk *et al.* (1985). In related work, the term *conjunct* has been used rather than *conjoin*, but Quirk et al. use the former term to denote "linking adverbials".

that match, to a greater or lesser extent, in terms of form, function, and meaning. Sentence (7) contains a compound verb phrase (VP).[2]

(7) She knew the risks [and] still insisted the operation should go ahead, Dr Addicott said.

Compound constituents are structures containing conjoins and their linking coordinators.

Subordination is defined as a hypotactic relationship holding between constituents at different levels of syntactic structure, referred to as superordinate and subordinate constituents. Sentence (8) contains a non-finite subordinate prepositional clause linked to a superordinate noun phrase (NP) in the main clause of the sentence.

(8) McKay[,] of Wark, Northumberland, denies five charges of contaminating food.

Grammatically, subordinate constituents are clausal.[3] Relative clauses are subordinate constituents that modify their superordinate constituents and depend for their meaning on those constituents (i.e. they are not independent clauses). Complex$_{\overline{RF}}$ NPs are those which are modified by non-restrictive relative clauses.

[2]In example sentences containing signs of syntactic complexity, signs in focus are indicated using square brackets while coordinated conjoins or subordinate constituents are underlined. Where appropriate, the location of elided elements is indicated using $\phi$. Occurrences of $\phi$ may be co-indexed with their antecedents.

[3]In many cases the *extant* parts of the subordinate clause are subclausal, as in the case of appositions.

The aim of this chapter is to determine whether or not there are reliable and explicit textual signs which indicate the occurrence of compound constituents (including compound clauses) and relative clauses (including those which modify complex$_{\overline{RF}}$ NPs) in English sentences. If so, the aim of the chapter is also to specify the forms and functions of these signs (**RQ-1**).

In this thesis, I use the term *signs of syntactic complexity* to denote words and punctuation marks that bound subordinate constituents and that coordinate the conjoins of compound constituents in sentences. The presence of these syntactic structures in a sentence is indicative of two commonly cited determinants of text processing difficulty: syntactic complexity and propositional density.

The syntactic functions of conjunctions, complementisers, relative adverbs, relative pronouns, and punctuation marks have been described in numerous linguistic studies of English (Chomsky, 1977; Quirk *et al.*, 1985; Nunberg *et al.*, 2002). For this reason, I posit a subset of words and punctuation marks of these categories as potential signs of syntactic complexity.

In this chapter, I present an annotation scheme to encode the linking functions of coordinators (conjunctions, punctuation marks, and pairs consisting of a punctuation mark followed by a conjunction)[4] and the bounding functions of subordinate clause boundaries (complementisers, wh-words, punctuation marks, and pairs consisting of a punctuation mark followed by a lexical sign).[5] The

[4]Restricted to a relatively unambiguous subset of coordinators to facilitate both the manual annotation task and the automatic tagging process (Chapter 5)

[5]With regard to punctuation, my research concerns the annotation of what Nunberg *et al.* (2002) refer to as *secondary boundary marks*. Due to practical resource limits, I focus on this subset, considering the annotation of other types of punctuation such as primary terminals, parentheses, dashes, punctuation involved in quotation, citation, and naming, capitalisation,

annotation scheme also encodes information about false signs that do not have coordinating or bounding functions (e.g. use of the word *that* as an anaphor or specifier). The chapter includes an analysis of the annotated corpus. It is expected that the encoding of this information can be gainfully exploited in the development of the sentence analysis and sentence simplification methods proposed in this thesis (Chapters 3–5).

## 2.1 Related Work

The main aim of the research described in this chapter is to produce annotated resources supporting development of a tool to automatically classify signs of syntactic complexity with specific information about their syntactic linking and bounding functions (Chapter 3). This sign tagger is a key component of the pipeline for automatic sentence simplification proposed in this thesis. Analysis of these annotated resources provides insights into **RQ-1**, concerning the form and characteristics of signs which may indicate the occurrence of compound clauses and complex$_{\overline{RF}}$ NPs in English sentences.

In view of these aims, the most relevant topics in previous work include the development of syntactically annotated resources, proposals to improve the quality of these resources, and the development of syntactic parsers that can automate the process.

There are currently a wide range of Treebanks available, providing access to syntactically annotated resources in many languages (Brants *et al.*, 2002; Simov

and word-level punctuation as issues to pursue in future research.

*et al.*, 2002; Hajič and Zemánek, 2004). In English, one of the most widely-used is the Penn Treebank (Marcus *et al.*, 1993) which has been exploited for the development of supervised syntactic parsers (Charniak and Johnson, 2005; Collins and Koo, 2005). Despite several criticisms of this resource, the Penn Treebank continues to be widely exploited in the field of supervised parsing because syntactically annotated data is scarce and expensive to produce. In addition, the Penn Treebank has been enhanced with other types of annotation, as described below.

Maier *et al.* (2012) observed that one shortcoming of the Penn Treebank is that punctuation symbols (commas and semicolons) are not tagged with information about their syntactic functions. If present, information of this type would facilitate the training of syntactic parsers that were better able to analyse sentences containing compound structures in which conjoins are linked in asyndetic coordination (Quirk *et al.*, 1985). To address this shortcoming, Maier *et al.* (2012) propose the addition of a second layer of annotation to disambiguate the role of punctuation in the Penn Treebank. They present a detailed scheme to ensure the consistent and reliable manual annotation of commas and semicolons with information to indicate their coordinating function.

An advantage of the approach described by Maier *et al.* (2012) is that the addition of an annotation layer is more cost-effective than the development of new annotated resources from scratch. By leveraging the original layer of annotation, minimal human effort and expertise is required. However, there are two main criticisms of this methodology. First, the scheme encodes only coarse-grained

information, with no discrimination between subclasses of coordinating and non-coordinating functions. Second, although production of the second annotation layer is inexpensive, application of the proposed scheme is costly as it depends on the availability of the original syntactic annotation layer. This limits the portability of the approach.

The annotation scheme developed in my research tags coordinators with more detailed information about their conjoins. It also encodes syntactic information about the extant constituents bounded by subordinate clause boundaries.[6] Resources produced using this scheme and the scheme proposed by Maier *et al.* (2012) can thus be regarded as complementary.

As noted earlier in this section, the Penn Treebank has been exploited in the development of supervised approaches to syntactic parsing. Given that this type of processing, if done with sufficient accuracy, could serve as the basis of any syntactic processing or sentence simplification system, there has been considerable research in improving the performance of syntactic parsing. Much of this involves techniques specifically designed to improve the parsing of coordinated structures (Charniak and Johnson, 2005; Ratnaparkhi *et al.*, 1994; Rus *et al.*, 2002; Kim and Lee, 2003; Nakov and Hearst, 2005; Hogan, 2007; Kawahara and Kurohashi, 2008; Kübler *et al.*, 2009). However, supervised methods trained on the Penn Treebank are likely to generate syntactic analyses subject to the shortcomings of that dataset. A better prospect is to exploit such traditional resources in

[6]Here, I use the term *extant constituent* to refer to the constituent that remains in the text when the rest of the clause has been elided. Examples of extant consituents would be noun phrases in non-finite nominal clauses and prepositional phrases in non-finite prepositional clauses.

combination with others, such as the annotation layer proposed by Maier *et al.* (2012).

The new scheme presented in this chapter is derived from the one proposed by Evans (2011), which aimed to improve performance in information extraction by simplifying sentences in input documents. In that scheme, members of a small set of textual markers of syntactic complexity were considered to belong to one of two broad classes: *coordinators* and *subordinators*. These groups were annotated with information on the syntactic projection level and grammatical category of conjoins linked and subordinate constituents bounded by those signs. The annotation of these markers, called *potential coordinators*, was exploited to develop an automatic classifier used in combination with a part-of-speech tagger and a set of rules to convert complex sentences into sequences of simpler sentences. Extrinsic evaluation showed that the simplification process evoked improvements in information extraction from clinical documents.

One weakness of the approach presented by Evans (2011) is that the classification scheme was derived by empirical analysis of rather homogeneous documents from a specialised source. Their consistency, together with the restricted range of linguistic phenomena manifested, imposes limits on the potential utility of the resources annotated. The scheme is incapable of encoding the full range of syntactic complexity encountered in documents of other registers.

## 2.2 The Annotated Corpus

This section presents the annotation scheme used to encode information on the syntactic function of a range of indicative signs of syntactic complexity (Section 2.2.1). The characteristics of three text collections annotated in accordance with this scheme are presented in Table 2.1.[7]

Rather than directly annotating the syntactic structure of each sentence, which is a time consuming and onerous process, we annotated a relatively small set of signs of syntactic complexity with information that will enable users to make inferences about the syntactic structure of sentences. The annotation provides information on the spans of subordinate syntactic constituents (their left and right boundaries, if explicit). It also provides information on the syntactic categories and relative sizes of both subordinate constituents and the conjoins of compound constituents. Under this annotation scheme, conjoins and subordinate constituents are not explicitly annotated. Only the signs themselves are tagged.

Texts of three registers (*news*, *health*, and *literature*), were collected from the METER corpus (Gaizauskas *et al.*, 2001), and the collections available at *patient.co.uk* and Project Gutenberg (*gutenberg.org*), respectively, to form the corpus. These texts were annotated with information about the syntactic functions of various signs of syntactic complexity. The characteristics of the texts are summarised in Table 2.1. The columns *Docs* and *Sents* display the total number of documents and sentences in the corpus. The next two columns provide

[7]The annotated resources and annotation guidelines are available at http://github.com/in6087/OB1/tree/master/corpora. Last accessed 6th January 2020.

information on the numbers of words and the average length of sentences in each collection. The final column provides information on the numbers of signs of syntactic complexity in each collection (*Total*) and the average number of signs per sentence (*Per Sent*).

Table 2.1: *Characteristics of the annotated corpus.*

| | | | **Words** | | **Signs** | |
|---|---|---|---|---|---|---|
| **Register (Source)** | **Docs** | **Sents** | **Total** | **Per Sent** | **Total** | **Per Sent** |
| Health (*patient.co.uk*) | 783 | 175 037 | 1 969 753 | 11.25 | 180 623 | 1.032 |
| Literature (Gutenberg) | 24 | 4 608 | 95 739 | 20.78 | 11 235 | 2.440 |
| News (METER) | 825 | 14 854 | 307 734 | 20.71 | 29 676 | 1.997 |

Inspection of Table 2.1 reveals that sentences in texts of the health register are approximately half as long as those found in texts of the other two registers. In line with intuition, sentences in these texts contain just over half as many signs of syntactic complexity as sentences in texts of the news register. From this table, we may infer that sentences in texts of the literary register are more syntactically complex than those of news or health (2.440 signs per sentence vs. 1.997 and 1.032, respectively). Syntactic information about the signs was manually annotated in texts of the three different registers: 10 756 signs in health, 11 204 in literature, and 12 718 in news. In the corpus, two main classes of signs were observed: *coordinators* and *subordination boundaries*.

### 2.2.1 Annotation Scheme

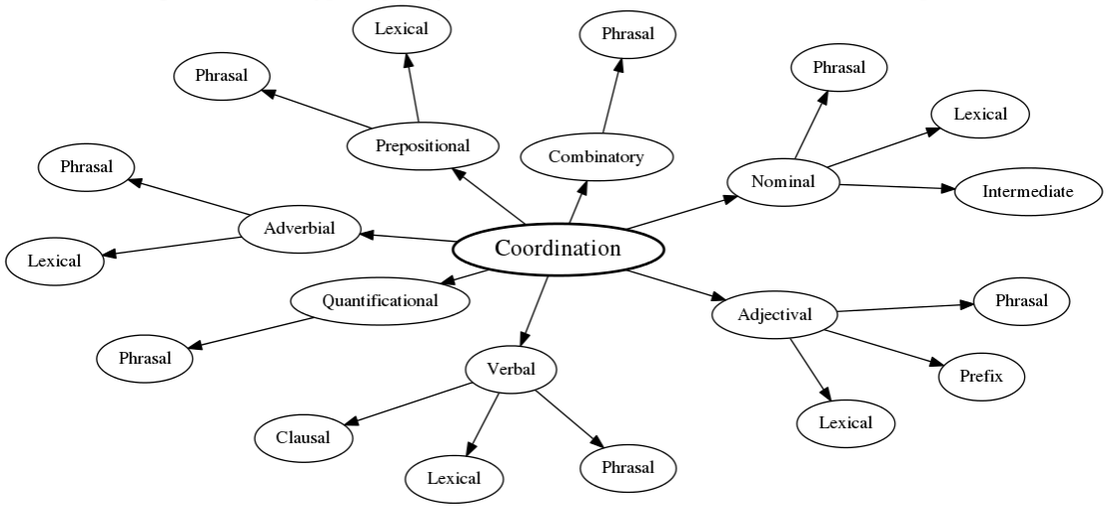
I updated the scheme that I had developed in previous work to encode information on the linking and bounding functions of several explicit signs of syntactic complexity (Evans, 2011). Development of the scheme was informed by corpus analysis and reference to sources covering a range of issues related to syntactic complexity, coordination, and subordination (Quirk *et al.*, 1985) and the function and distribution of punctuation marks (Nunberg *et al.*, 2002).

There are two major differences between the two schemes. First, the set of *subordinators* posited in my previous work (Evans, 2011) was expanded, re-designated, and sub-categorised to form one set of *left boundaries of subordinate clauses* and a second set of *right boundaries of subordinate clauses*. The sets of coordinators and subordinators referred to in my previous work were extended to include a larger number of signs in the new scheme. Second, a wider range of conjoins and relative clauses can be distinguished in the new scheme than was possible in that used in my previous approach.

The new annotation scheme is intended to encode information on the linking and bounding functions of different signs of syntactic complexity. Figures 2.1 and 2.2 present the scheme graphically. They depict each type of complexity, with the central nodes representing core functions of each type of sign. Coordinating functions are depicted in Fig. 2.1 and subordinating functions in Fig. 2.2.

As noted by Quirk *et al.* (1985), and supported by corpus evidence (see Section 2.2.3), coordinators usually link constituents of the same syntactic category.

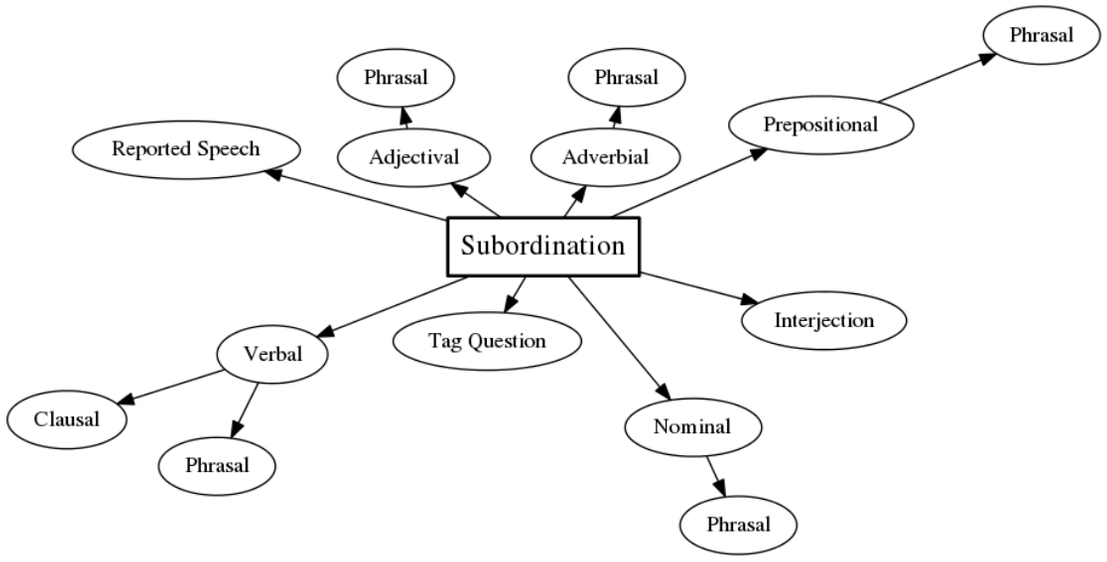
Figure 2.1: *Typology of coordinators annotated in the corpus*




Nodes in Fig. 2.1 directly connected to the central one representing coordination denote different syntactic categories of coordination. The leaf nodes represent the different levels of syntactic projection that coordinated conjoins may take (Chomsky, 1986). In terms of syntactic complexity, conjoins were observed in a range of projection levels: zero (morphemic and lexical), intermediate, maximal (phrasal), and extended (clausal).

In Fig. 2.2, nodes directly connected to the central one representing subordination denote different types of subordinate clause. These include five syntactic categories, reported speech, and two other units typical of spontaneous or colloquial language (tag questions and interjections). Given that non-finite subordinate clauses can be considered to involve ellipsis of the relative pronoun and the copula, the leaf nodes in the diagram represent the different syntactic projection levels of the extant portions of the clause. In non-finite (adjectival, adverbial, nominal, prepositional, and verbal) clauses, the extant constituents are all max-

Figure 2.2: *Typology of subordination boundaries annotated in the corpus*




imal projections. Finite subordinate clauses are extended projections of verbs. Interjections, reported speech, and tag questions can occur at numerous levels of projection, from zero to extended.

There is a third general type of linguistic/syntactic complexity that is not included in Fig. 2.1 or Fig. 2.2, which is used to denote special uses of potential signs of syntactic complexity. These include signs with anaphoric and specifying functions and coordinators linking several additional types of conjoin.

To summarise, the annotation scheme described in this chapter is used to develop resources in which coordinators are annotated with information about the specific type of coordination that they embody and therefore the specific types of conjoins that they link. Subordination boundaries are annotated with information about the specific type of subordinate constituent that they bound. As discussed in the next two sections, there is a limited set of signs which poten-

tially indicate syntactic complexity. For this reason, they can be automatically detected with great reliability by a specially developed annotation tool and presented to human annotators for classification. One advantage of this approach is that the annotation task does not require the level of expertise in syntax that would be required in the development of a treebank. The annotation presented in this chapter involves a much less detailed analysis of the syntactic structure of each sentence and does not depend on strict adherence to a specific linguistic theory. As a result, the task is less onerous and can be performed more rapidly. The remainder of this chapter discusses each category of sign in more detail. Appendix B provides additional examples of each of the classes of sign presented in Sections 2.2.1.1–2.2.1.3.

#### 2.2.1.1 Coordinators

There are three major types of coordinator: *conjunctions* ([*and*], [*but*], and [*or*]), which have an exclusively coordinating function; *punctuation* marks ([,] and [;]), which may occur as coordinators in some contexts and as subordinate clause boundaries in others; and *punctuation-conjunction pairs* ([, *and*], [; *and*], [, *but*], [; *but*], [, *or*], and [; *or*]), which are similarly ambiguous.

Signs that have been identified as coordinators are classified by assigning them to one of the subclasses (leaf nodes connected to coordination) displayed in Fig. 2.1. The class labels used in the annotation scheme are acronyms that indicate the function of each annotated coordinator.

- The first part of the acronym indicates the coordinating function (C).

- The second part indicates the syntactic projection level of the conjoins linked by the coordinator. These include constituents at the morphemic (P), lexical (L), intermediate (I), maximal (M), and extended (E) levels of syntactic projection.

- The third part of the acronym indicates the syntactic category of the co-ordinated conjoins. These may be verbal (V), nominal (N), adjectival (A), adverbial (Adv), prepositional (P), or quantificational (Q).

- The fourth part of the acronym is optional and takes a numerical value. It is used to distinguish between the coordination of different types of nominal and verbal maximal projection. These sub-types are:

    1. default maximal projections (CMA1, CMN1, CMV1);
    2. maximal projections in which the head of the second conjoin has been elided (CMN2, CMV2);
    3. maximal projections in which the complement of the head of the first conjoin has been elided (CMV3);
    4. maximal projections in which the head of the first conjoin has been elided (CMN4).

To illustrate, the class label CLA indicates that the sign is a coordinator of two lexical projections of an adjective (9) whereas CMP indicates that the sign is a coordinator of two maximal projections of a preposition (10).[8]

[8]In these examples, underlining is used to indicate the spans of compound constituents for the reader. These spans were not annotated.

(9) "He had a stable [$_{CLA}$ and] loving family."

(10) "But the melancholy experience of the courtrooms [$_{CMP}$ and] of life is that people have a good character in some respects and not in others."

Several additional class labels may be assigned to coordinators. COMBINATORY is used to tag markables indicating combinatory coordination (11), typically used in fixed phrases, proverbs, and aphoristic sentences. These coordinations are usually atomic and require a different type of processing when simplifying the sentences in which they occur.

(11) Withdrawal symptoms that may occur include: dizziness, anxiety and agitation, sleep disturbance, flu-like symptoms, diarrhoea, abdominal cramps, pins [$_{COMBINATORY}$ and] needles, mood swings, feeling sick, and low mood.

CXE is used to denote coordinators linking conjoins with unusual patterns of ellipsis (12) while CMX is assigned to coordinators linking syntactically ill-assorted conjoins[9] (13).

(12) The 38-page judgment stated that Mrs Coughlan, a tetraplegic, was entitled to free nursing care because her primary need for accommodation was$_i$ a health need [$_{CXE}$ and] her nursing needs $\phi_i$ not 'incidental'.

[9]This term denotes pairs of conjoins that do not match in terms of grammatical category (Quirk *et al.*, 1985).

(13) "Name something that is currently on BBC1 that gets people excited $[_{CMX}$ and] talking about it.

#### 2.2.1.2 Subordination Boundaries

There are six major types of subordination boundary. Of these, two involve lexical signs (complementisers ([*that*]) and wh-words ([*what*], [*when*], [*where*], [*which*], [*while*], and [*who*])), while four involve the use of punctuation either in isolation ([,], [;], and [:]) or in a pair, followed by any other sign of syntactic complexity. *Complementisers* and *wh-words* exclusively serve to bound subordinate clauses. As noted in Section 2.2.1.1, signs of syntactic complexity involving commas and semicolons may serve as coordinators in some contexts and as subordination boundaries in others.

Signs that have been identified as subordination boundaries are classified by assigning them to one of the subclasses (leaf nodes connected to subordination) displayed in Fig. 2.2. The class labels used in the annotation scheme are acronyms that indicate the function of each annotated subordination boundary.

- The first part of the acronym can indicate the left boundary of a subordinate clause (i.e. the start of the subordinate clause; SS) or the right boundary of a subordinate clause (i.e. the end of the subordinate clause; ES).

- The second part indicates the syntactic projection level of the extant consituent in the bounded clause. These include constituents at the maximal (M) and extended (E) levels of syntactic projection.

- The third part of the acronym indicates the syntactic category of the extant constituent in the bounded clause. These may be verbal (V), nominal (N), adjectival (A), adverbial (Adv), or prepositional (P).

The scheme includes class labels for annotation of the boundaries of the following additional types of subordinate clause:

- Interjections (SSMI/ESMI),
- direct quotes (SSCM/ESCM),
- tag questions (STQ),[10]
- constituents of ambiguous syntactic category (SSMX/ESMX).[11]

To illustrate, the class label ESMV indicates that the sign is the right boundary (i.e. end) of a non-finite subordinate clause whose extant constituent is the maximal projection of a verb (14).

(14) "Being put into a psychiatric ward with people with long-term mental illnesses who are shaking with the drugs they are taking[$_{ESMV}$ ,] there's no way you can feel normal and be OK with yourself," she told BBC TV's *That's Esther* programme with Esther Rantzen.

[10]It should be noted that the right boundary of a tag question is usually a sentence boundary (question mark). In the research supporting this thesis, sentence boundaries were not considered markable. As a result, no signs of syntactic complexity serving as the right boundaries of tag questions were encountered in the corpus presented in Section 2.2.3.

[11]There are just 25 instances of this class in the 34 678 signs annotated so far.

#### 2.2.1.3 False Signs

Finally, the annotation scheme includes the class label SPECIAL to denote false signs of syntactic complexity such as use of the word *that* with a specifying (15) or referential (16) function.

(15) "I'm quite happy to abandon [$_{SPECIAL}$ that] specific point" he said.

(16) 'Because of your involvement in the past with trying to stop all [$_{SPECIAL}$ that] in your work, you more than anybody else should have known the misery of people who had become addicted.'

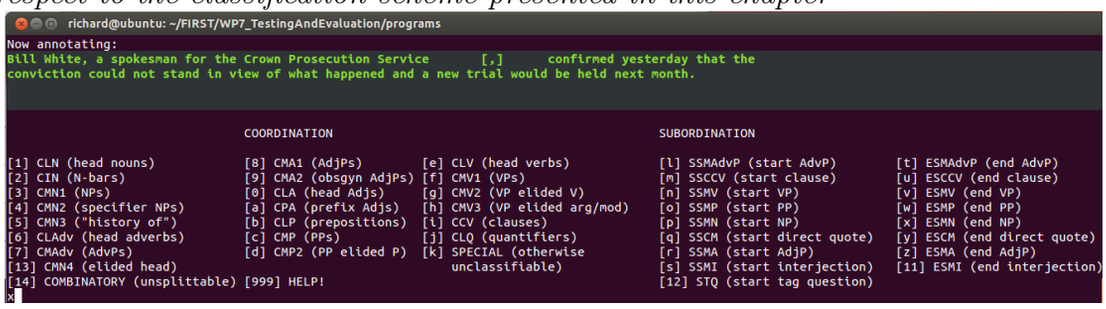
### 2.2.2 Corpus Annotation

The resources described in Section 2.2.3 were produced by manual annotation of plain text. I developed a purpose-built annotation tool to facilitate this process. Figure 2.3 shows a screenshot of the tool. The text to be annotated is first tokenised to enable automatic detection of sentence, word, and punctuation tokens. The tool automatically detects and highlights the signs of syntactic complexity listed in Sections 2.2.1.1– 2.2.1.3 of this chapter. For each one, it displays a version of the sentence containing the sign in which the location of the sign is clearly highlighted by square bracketing and white space, together with the list of classes to which the sign may belong. The classes are presented as a table of triples consisting of an index key, the class code, and a brief description of the class. Annotators are then prompted for selection of the appropriate one. Selection of the class is made by typing its index key followed by ENTER. Given that

a single sentence may contain multiple signs of syntactic complexity, one characteristic of the annotation process is that the same sentence may be presented several times to the annotator, with the location of a different sign of syntactic complexity highlighted in each case. For example, Sentence (17) is displayed seven times, once for each of the signs it contains.

(17) As researcher for the programme, Ms Price had arranged for the guests to appear on the show and the article went on to allege that not only were they fakes but that Ms Price had known this and had deliberately deceived her employers and viewers.

Figure 2.3: *Screenshot of the tool to annotate signs of syntactic complexity with respect to the classification scheme presented in this chapter*




The automatic detection of each sign by the tool enables rapid annotation, since this is one of the most time consuming and unreliable aspects of manual annotation. Selection of the appropriate class from a choice of 38 is potentially difficult and time consuming. In an attempt to alleviate this problem, as shown in Fig. 2.3, the annotation tool was designed to present the set of classes so

that closely related ones are in close proximity. Class labels for coordinators are displayed in the first three columns of the interface with subordination boundaries occupying the fourth and fifth columns. Column four lists left boundaries while column five lists their complementary right boundaries. Classes involving conjoins of the same syntactic category are also in close proximity. For example, column three lists classes denoting verbal conjoins.

Sentence (17) would be annotated in the following way.[12]

The first comma in the sentence follows a non-finite subordinate prepositional clause, and serves as its right boundary:

(17). a. As researcher for the programme[$_{ESMP}$ ,] Ms Price had arranged for the guests to appear on the show and the article went on to allege that not only were they fakes but that Ms Price had known this and had deliberately deceived her employers and viewers.

The first conjunction links two clauses in coordination:

(17). b. As researcher for the programme, Ms Price had arranged for the guests to appear on the show [$_{CEV}$ and] the article went on to allege that not only were they fakes but that Ms Price had known this and had deliberately deceived her employers and viewers.

[12]In this example, the labels assigned to different signs of syntactic complexity appear as subscripts added to the bracketing that indicates their location. As mentioned in Footnote 8, I added underlining to provide readers of my thesis with an indication of the underlying sentence structure. Annotators did *not* annotate this structure.

The first complementiser appears at the start of a finite subordinate clause, and serves as its left boundary:

(17). c. As researcher for the programme, Ms Price had arranged for the guests to appear on the show and the article went on to allege [$_{SSEV}$ that] not only were they fakes but that Ms Price had known this and had deliberately deceived her employers and viewers.

The second conjunction links two finite subordinate clauses in coordination:

(17). d. As researcher for the programme, Ms Price had arranged for the guests to appear on the show and the article went on to allege that not only were they fakes [$_{CEV}$ but] that Ms Price had known this and had deliberately deceived her employers and viewers.

The second complementiser appears at the start of a finite subordinate clause, and serves as its left boundary:

(17). e. As researcher for the programme, Ms Price had arranged for the guests to appear on the show and the article went on to allege that not only were they fakes but [$_{SSEV}$ that] Ms Price had known this and had deliberately deceived her employers and viewers.

The third conjunction links two predications (verb phrases) in coordination:

(17). f. As researcher for the programme, Ms Price had arranged for the guests to appear on the show and the article went on to allege that not only were they fakes but that Ms Price had known this [$_{CMV1}$ and] had deliberately deceived her employers and viewers.

The fourth conjunction links two nouns in coordination:

(17). g. As researcher for the programme, Ms Price had arranged for the guests to appear on the show and the article went on to allege that not only were they fakes but that Ms Price had known this and had deliberately deceived her employers [$_{CLN}$ and] viewers.

The fully annotated sentence is:

(17). g. As researcher for the programme[$_{ESMP}$ ,] Ms Price had arranged for the guests to appear on the show [$_{CEV}$ and] the article went on to allege [$_{SSEV}$ that] not only were they fakes [$_{CEV}$ but] [$_{SSEV}$ that] Ms Price had known this [$_{CMV1}$ and] had deliberately deceived her employers [$_{CLN}$ and] viewers.

### 2.2.3 Corpus Analysis

This section provides an evaluation of the annotated resources developed in this research. The descriptions include analysis of the distribution of different signs of syntactic complexity and the classes to which they belong. To provide insight

into reliability and consistency of annotation, this section includes an assessment of inter-annotator agreement using the Kappa statistic ($\kappa$) (Cohen, 1960).

A sample of the signs of syntactic complexity occurring in the three collections (Table 2.1) were manually classified in accordance with the annotation scheme presented in Section 2.2.1. During the annotation process, annotators were provided with access to annotation guidelines and to Quirk et al.'s (1985) Comprehensive Grammar of the English Language.

#### 2.2.3.1 Sign and Class Distribution

In documents of all registers, the comma, the conjunction [*and*], and the complementiser [*that*] were the most frequently occurring signs of syntactic complexity. Use of the sign [*, and*] was more characteristic of the register of literature (15.95%) than news (2.49%) or health (3.30%). In comparison with documents of the other registers, the conjunction [*or*] was most frequent in those of health. Use of the semicolon was relatively frequent in nineteenth/twentieth century literature.

In the three registers, the classes to which different signs of syntactic complexity most frequently belonged were left boundaries of finite subordinate clauses (SSEV), coordinators of verb phrases (CMV1), and coordinators of noun phrases (CMN1). The comma is used with a wide range of coordinating and bounding functions in all three registers, with its function as boundary of a subordinate clause being considerably more frequent than its function as a coordinator.[13] In the literary register, the comma is used slightly more often as the right boundary

[13]In this paragraph, *subordinate clause* denotes constituents left bounded by signs of any class in the set {SSEV, SSMV, SSMP, SSMV, SSMA, SSMAdv}.

Table 2.2: *Relative frequency of indicative signs and classes in the three collections of annotated documents*

| **Relative frequency** | **Collection 1 (News)** | **Collection 2 (Health)** | **Collection 3 (Literature)** |
|---|---|---|---|
| High | Sign [*:*] of class SSCM<br>Sign [*who*] of class SSEV | Sign [*or*] of classes CLN, CMN1, and CLA<br>Signs of class ESMAdv<br>Signs of class ESMN | Sign [*;*] of classes CEV, SSEV, and ESEV<br>Signs of class SSCM |
| Low | | Signs of class CEV | Signs of classes SSMN and ESMN |

of finite subordinate clauses (56% of its occurrences), while in patient healthcare documents it is used this way considerably more often (81% of its occurrences). In news articles, it is used slightly more frequently as the left boundary of finite subordinate clauses (52% of its occurrences).

As noted in Section 2.2 (page 35), 10 756 signs of syntactic complexity were annotated in texts of the health register, 11 204 in texts of the literary register, and 12 718 in texts of the news register. Due to the large number of classes and signs used in the annotation, it is difficult to visualise their distribution in an practical way.[14] Table 2.3 displays the frequency distribution of the twenty most common signs and the twelve most common tags assigned in the three text registers. Space restrictions prevent display of the full distribution of 35 annotated signs and 38 assigned tags. The row *Total* provides information on the total number of signs of each class in the corpus.

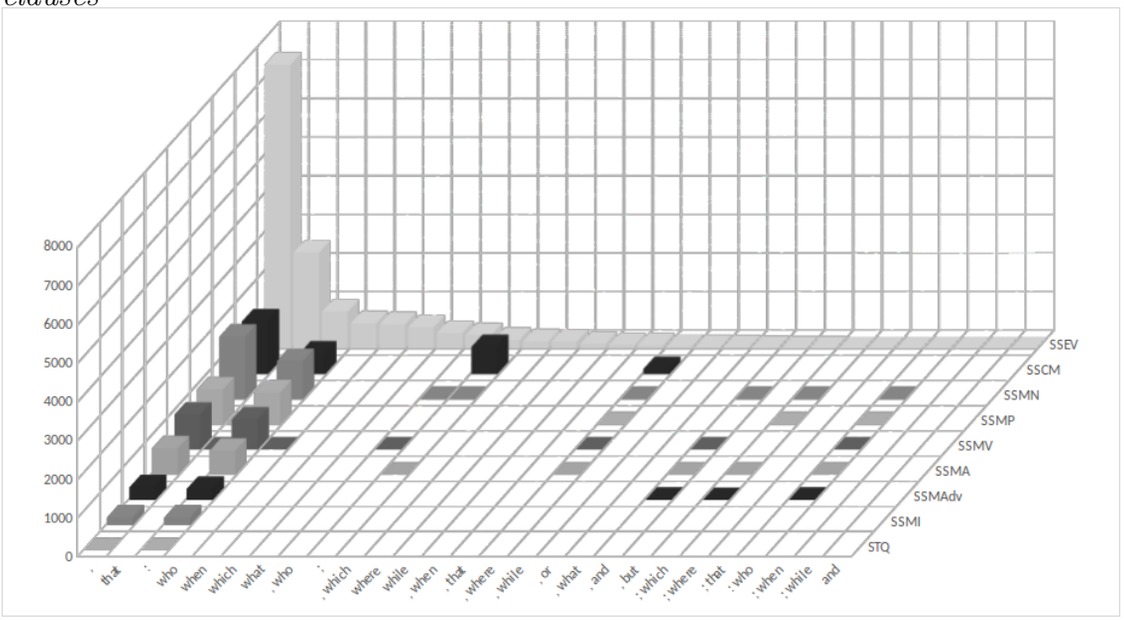
Figures 2.4–2.6, respectively, display the frequency distributions of signs and tags used to annotate the left boundaries of subordinate clauses, the right bound-

[14]The raw data is accessible at http://github.com/in6087/OB1/tree/master/corpora. Last accessed 6th January 2020.

Table 2.3: *Sign frequency distribution of the twenty most frequent signs and twelve most frequent tags in the dataset*

| **Sign/ Tag** | **SSEV** | **CEV** | **CMV1** | **CMN1** | **CLN** | **SSMN** | **ESEV** | **SSCM** | **ESMP** | **ESMN** | **ESCM** | **SSMP** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| , that | 89 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 6 | 2 | 0 | 0 |
| while | 124 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| where | 190 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ; and | 0 | 187 | 2 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| , which | 194 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| , or | 7 | 63 | 65 | 86 | 18 | 8 | 1 | 0 | 0 | 0 | 0 | 0 |
| , who | 353 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 0 | 0 |
| what | 406 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| but | 0 | 237 | 137 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ; | 57 | 190 | 6 | 42 | 7 | 3 | 4 | 133 | 2 | 1 | 2 | 4 |
| , but | 3 | 374 | 98 | 11 | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 2 |
| which | 561 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| when | 640 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| who | 665 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| : | 233 | 130 | 0 | 0 | 0 | 10 | 0 | 702 | 1 | 0 | 2 | 0 |
| or | 0 | 46 | 101 | 353 | 372 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| , and | 53 | 1 005 | 823 | 315 | 53 | 5 | 6 | 0 | 2 | 8 | 1 | 2 |
| that | 2 487 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| and | 0 | 712 | 1 543 | 1 253 | 873 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| , | 967 | 235 | 219 | 745 | 559 | 986 | 1 375 | 531 | 1 187 | 495 | 1 040 | 828 |
| **Total** | **7 357** | **3 298** | **2 995** | **2 820** | **1 882** | **1 627** | **1 588** | **1 366** | **1 275** | **1 162** | **1 046** | **909** |

Figure 2.4: *Sign/Tag frequency distribution of left boundaries of subordinate clauses*




aries of subordinate clauses, and coordinators, in texts of all three registers.

Table 2.2 presents information on the relative frequency of signs and classes characteristic of all three document collections. These features are indicative of various linguistic properties of each one:

- Collection 1 (News)

    1. Frequent use of the colon [*:*] to bound reported speech (SSCM).
    2. Frequent provision of additional explanatory information about the people mentioned in news articles.

- Collection 2 (Health)

    1. Frequent presentation of lists of alternative possibilities for treatment options, symptoms, anatomical locations, and medical procedures.

Figure 2.5: *Sign/Tag frequency distribution of right boundaries of subordinate clauses*

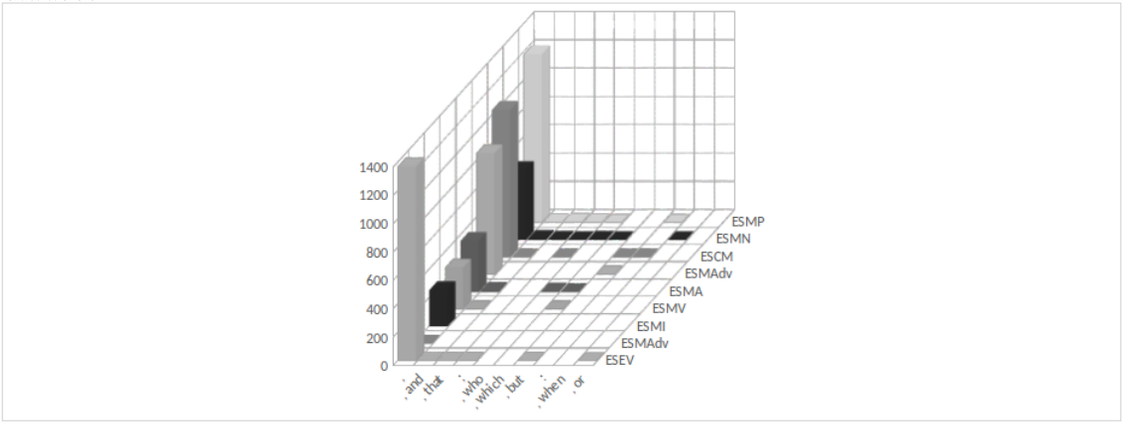
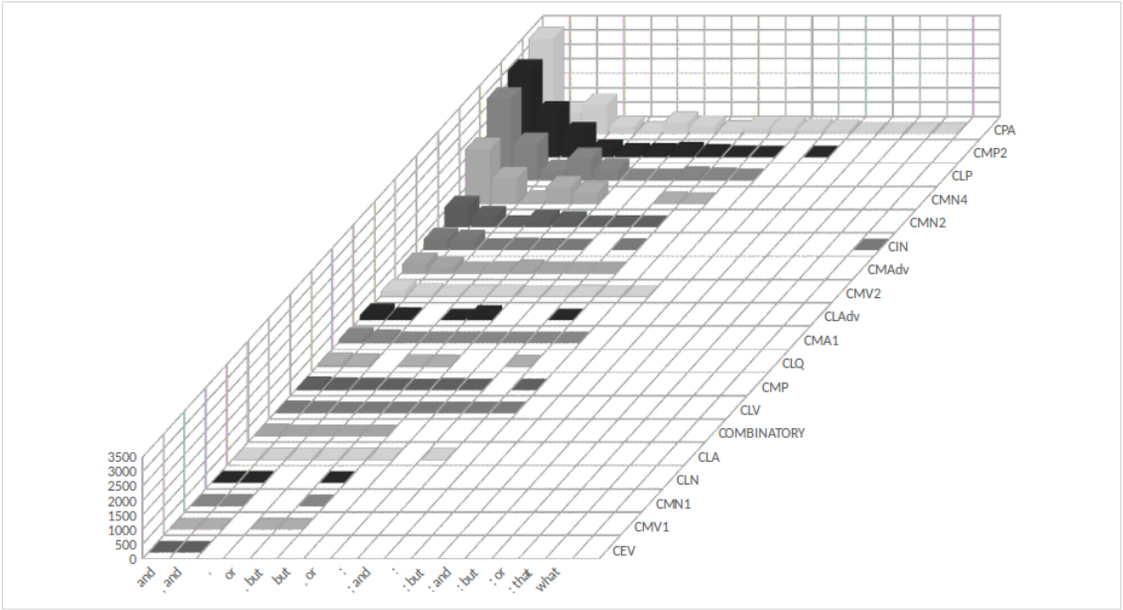



Figure 2.6: *Sign/Tag frequency distribution of coordinators*



2. Frequent occurrence of sentence-initial adverbials to contextualise symptoms or treatment options.

3. Frequent use of appositions to provide explanatory definitions of terms.

4. Sentences are of limited overall complexity, making the documents accessible by a wide a range of readers.

- Collection 3 (Literature)

1. Terms used in literary documents are rarely defined via explanatory subordinate noun phrases (SSMN and ESMN).

2. There is greater variation in the occurrence of direct speech and reporting clauses than is the case for news articles.

#### 2.2.3.2 Consistency/Reliability of Annotation

A subset of 1000 annotations of each of the three collections was cross-annotated and a confusion matrix plotted. The values of Kappa obtained for the annotations were 0.80 for signs annotated in news articles, 0.74 for those in documents conveying patient healthcare information, and 0.76 for those in literary documents. These levels imply a minimum of "substantial agreement" between annotators (Viera and Garrett, 2005).

In the annotation of news articles, the most common disagreement (8.67% of the cases) occurred for signs that one annotator considered to indicate clause coordination (CEV) and the other considered to indicate verb phrase coordination (CMV1), especially in cases involving imperative clauses and complex VP

conjoins with clausal arguments and modifiers.

Of the most frequent types of disagreement in the registers of health and literature, many were of the same type as those in news articles, though some register-specific types were also in evidence.

In the register of health, hyphenated items ((18), (19)) were a cause of disagreement, with annotators disagreeing on the projection level of nominal conjoins. Disagreement on the annotation of signs of syntactic complexity in bibliographic references (20) was also evident in this register. This latter type of disagreement can be accounted for by the lack of guidance on this potential usage of signs in the instructions given to annotators.

(18) One may be used instead of an ACE inhibitor if you have problems [or] side-effects with taking an ACE inhibitor (such as a persistent cough).

(19) Magin P, Pond D, Smith W, et al; A systematic review of the evidence for 'myths and misconceptions' in acne management: diet[,] face-washing and sunlight.; Fam Pract.

(20) Magnussen RA, Dunn WR, Thomson AB[;] Nonoperative treatment of midportion Achilles tendinopathy: a systematic review.

In the register of literature, many disagreements involved the coordination of verb phrases. It was observed that in cases where the second conjoin of a coordinator has an adverbial pre-modifier consisting of a *wh*-complementiser and its clausal complement, the coordinator is misclassified as linking two clauses

(21).

(21) The king comforted her and said: 'Leave your bedroom door open this night, and my servants shall stand outside[, and] when he has fallen asleep shall go in, bind him, and take him on board a ship which shall carry him into the wide world.

Another type of disagreement arising in the annotation of signs in literary text involves signs used in colloquial constructions such as interjections. At times, interjections are nested and there is disagreement about whether a sign is serving as the right boundary of an interjection or the left boundary of one that is subordinate (22).

(22) "Ah[,] well! We did our best, the dear knows."

While it is difficult to motivate any particular choice of class label (between SSMI and ESMI) for such signs, it is expected that consistency of annotation can be improved by adding to the guidelines an instruction for annotators to select SSMI (for example), when encountering signs in such contexts.

Approximately 12 months after the initial annotation, a set of 500 signs of syntactic complexity was re-annotated by one annotator. Of the signs involved, 99.97% were assigned the same class label ($\kappa = 0.9997$). This result indicates that the annotation task is well defined. Despite the protracted period between the annotation sessions, in nearly all cases, the second session led to the same

syntactic functions being assigned to signs of syntactic complexity. This implies that fidelity of annotation is based on annotators' comprehension of the task rather than guesswork or chance.

## 2.3 Contribution to Research Question RQ-1

The research described in this chapter addresses **RQ-1**:

> *Are there reliable and explicit textual signs which can indicate the occurrence of compound constituents and complex$_{\overline{RF}}$ NPs in English sentences? What are these signs and what are their functions?*

in several ways.

There are two aspects to be considered regarding the first question. Analysis of the syntactic linking and bounding functions exhibited by the set of signs introduced in Sections 2.2.1.1 and 2.2.1.2 provided partial confirmation of the *reliability* of the posited signs. The frequency distribution of class labels assigned in the annotated data reveals that, as a proportion of all signs, 38.89% coordinate the conjoins of compound constituents (all types), while 10.65% coordinate the conjoins of compound clauses. 26.96% of the signs bound restrictive or non-restrictive finite relative clauses, a subset of which modify complex$_{\overline{RF}}$ NPs, while 32.40% bound independent subordinate clauses or non-finite clauses. Thus, the signs of syntactic complexity are fairly reliable indicators of the phenomena relevant to the automatic sentence simplification method proposed in this thesis (Chapter 5).

The second aspect of the first question in **RQ-1** relates to *explicitness*. The occurrence of many compounds and relative clauses in English sentences are not marked by signs of syntactic complexity. However, analysis of 3737 syntactically annotated sentences in the Penn Treebank (Marcus *et al.*, 1993), containing a total of 9651 nested clauses of various types (including conjoins of compound clauses and finite relative clauses),[15] revealed that 56.74% had at least one adjacent sign of syntactic complexity.

The signs thus enable detection of just over half of the compound clauses and finite relative clauses in the sample. Due to the complexity and ambiguity of the annotation of clauses in the Penn Treebank, it has not been possible to determine whether the signs are better indicators of one or other of the different types of embedded clause.

The occurrence of signs of syntactic complexity in a text is indicative of more than half of the compound clauses and finite relative clauses in it. Systems using automatic detection of these signs as a basis for processing compound clauses and $\text{complex}_{\overline{RF}}$ NPs therefore have the potential to achieve reasonable coverage.

[15] With S, SBAR, SBARQ, SINV, and SQ clause level bracket labels (Bies *et al.*, 1995)

# Chapter 3

# Automatic Classification of Signs of Syntactic Complexity

The research described in this chapter addresses research question **RQ-2**, which is concerned with the feasibility of automatically and reliably detecting and classifying explicit textual markers of syntactic complexity with respect to their specific syntactic functions. In addressing this research question, the current chapter describes an automatic method developed in collaborative work for this purpose (Section 3.2). It also presents a quantitative evaluation of our approach (Section 3.3). The chapter ends with an analysis of my contribution to research question **RQ-2**.

The availability of an automatic tool to detect and classify explicit signs of syntactic complexity in accordance with the annotation scheme presented in Section 2.2.1 has the potential to facilitate the process of automatic sentence simplification. In this context, it can provide a shallow yet relatively detailed syntactic analysis of input sentences, avoiding the need for computationally intensive analysis using a full syntactic parser. Information about the explicit signs of syntactic complexity in a sentence can provide a basis for automatic detection of the syntactic constituents to which transformation operations are applied in sentence simplification.

## 3.1 Background Information

State of the art PoS taggers provide little information on the syntactic functions of conjunctions, complementisers, wh-words, and punctuation marks, and are of limited use in automatic sentence simplification. Detailed information can be derived from syntactic parsers but this information may be inaccurate for long complex sentences. Further, derivation of such information may be non-trivial, requiring the development of rules to handle the detailed processing of syntactic tree structures.

Van Delden and Gomez (2002) developed a tool to provide information on the linking and bounding functions of commas. Their method operates in two phases. In the first, 38 finite state automata are applied to PoS tagged data to derive an initial tagging of commas. In the second, information from a tag co-occurrence matrix derived from hand annotated training data is used to improve the initial tagging. Their system achieves accuracy of 0.91-0.95 in identifying the syntactic functions of commas in a collection of encyclopaedia and news articles. The inability of the method to process other signs limits its usefulness in automatic sentence simplification.

Srikumar *et al.* (2008) developed a machine learning approach to disambiguate the roles of commas indicating four different entailed semantic relations: *substitute* (hyponymy), *attributive*, *locative*, and *listing* (of elements in a group). A fifth class was reserved for *other* roles indicated by commas. In their approach, identification of the semantic relations is based on identifying structures in full

syntactic analyses of input sentences. In the context of my work, this is undesirable as the primary motivation for development of a tool to determine the syntactic functions of commas and other signs of syntactic complexity is to avoid the need for full syntactic analysis of long complex sentences. Srikumar *et al.* (2008) do not evaluate the classification of commas made by their system in isolation, but instead evaluate its ability to identify syntactic constituents over which the semantic relations indicated by the commas hold.[1] As a result, it is difficult to compare the evaluation figures that they report with those obtained by the system that I present in this chapter.

Evans (2011) presented a ML approach to tagging potential coordinators (analogous to signs of syntactic complexity) with information about the syntactic linking and bounding functions that they serve in clinical texts.[2] In that work, training data was developed in which potential coordinators were manually tagged and represented as feature vectors encoding information about their form and linguistic context. Memory-based learning was then used to derive a classification model for each type of potential coordinator ([*and*], [,], [*, and*], etc.) that could then be applied to tag previously unseen potential coordinators represented in the same way. The method had an overall accuracy of 0.832 in assigning one of 30 class labels to 7 types of potential coordinator. This approach was developed for use in a restricted, relatively homogeneous domain (patient notes), demonstrating only a limited range of syntactic constructions. Further, it is only

[1] For this task of identifying syntactic constituents, $F_1 = 0.776$ for their system.

[2] Differences between the two approaches to shallow syntactic analysis are discussed in Chapter 1 (pages 7–9).

capable of classifying a subset (namely, conjunctions, commas, and complementisers) of the signs of syntactic complexity annotated in the corpus described in Chapter 2 of this thesis. The classification scheme employed in my previous work (Evans, 2011) encodes only limited information about the syntactic functions of those signs. Thus, this approach is inadequate for tagging signs in the range of texts considered in this thesis.

## 3.2 A Machine Learning Method for Sign Tagging

The absence of automatic tools to identify the full set of syntactic functions of the full set of signs motivated me, in collaboration with Drs. Dornescu and Orasan, to develop a new sign tagger, exploiting the corpus described in Chapter 2. The method proposed is based on a ML algorithm which is able to classify each sign according to the classes annotated in our corpus. The algorithm relies mainly on the intrasentential context of the sign to determine its class. We conducted experiments to optimize the performance of our sign tagger by evaluating it with alternate settings of four parameters: *algorithm type*, *tagging mode*, *features* used to represent instances in the training data, and the selection of *training data*. The evaluation was carried out over the annotated part of the corpus presented in Chapter 2 (Table 2.1) and is expressed using the evaluation metrics typically used in NLP: precision, recall, $F_1$-score, and accuracy.[3] In all of the experiments, 10-fold cross validation was employed.

[3]In this thesis, these measures will be presented as decimals, not percentages.

### 3.2.1 Algorithm

With regard to algorithm type, we found that sequence based CRF tagging models (Lafferty *et al.*, 2001; Sutton and McCallum, 2011) provided better performance in the automatic tagging of signs than methods in which each sign is tagged independently of other signs in the same sentence. Table 3.1 displays performance of the tagging model when alternative algorithms are applied and used to process text of the news register.

Table 3.1: *Performance of machine learning algorithms when sign tagging in texts of the news register*

| | **Correct** | **Accuracy** |
|---|---|---|
| CRF-extended | 10 248 | 0.8058 |
| CRF-core | 9 979 | 0.7846 |
| SMO | 7 213 | 0.5671 |
| NB | 6 712 | 0.5278 |
| J48 | 6 742 | 0.5301 |
| IB7 | 6 662 | 0.5238 |

### 3.2.2 Tagging Mode

Our approach contrasts with my previous work (Evans, 2011), in which signs are classified independently, using a memory-based learning algorithm. In the current system, texts are treated as sets of token sequences, with each sequence corresponding to a sentence in the text. A prediction is made of the tag of every token in the text, not just the subset of tokens that are signs of syntactic

complexity.[4] The tags to be assigned are treated as variables that depend both on other observed variables and on the probabilities of the potential tags of other tokens occurring in the same sentence.

When applying the CRF tagger, two tagging modes were evaluated. In the first (*simple*), signs of syntactic complexity in the training data were tagged with the classes specified in Chapter 2 (Section 2.2) while non-signs were tagged *NA* to indicate that they are not signs of syntactic complexity. 90% of the tokens being tagged in this setting are non-signs and we were concerned that the derived tagging models would prioritize accurate tagging of non-signs at the expense of the task we are really interested in, which is the tagging of signs. In this chapter, to avoid misleading results, evaluation scores are reported in the context only of sign tagging, not token tagging. In the *simple* tagging mode, the model operates at acceptable levels of accuracy when sign tagging ($0.7846 < acc < 0.8323$). In the second tagging mode (*BIO*), signs of syntactic complexity in the training data were tagged with the class labels specified in Chapter 2 while non-signs were tagged with a class label matching that of the closest preceding sign.[5] Table 3.2 displays a sample of the annotations used in each of the two tagging modes. The sign tagger has slightly better accuracy when operating in the *BIO* tagging mode ($0.7991 < acc < 0.8383$).[6]

[4] In this context, signs comprising a punctuation mark followed by a word are treated as single tokens.

[5] We misapplied the term *BIO* in our previous work (Dornescu *et al.*, 2013). I continue to use it in this chapter for consistency. A more appropriate term would be *forward fill* (or *ffill*) tagging mode.

[6] The difference is marginal, but the *simple* tagging mode achieves superior performance to the *BIO* mode when applied to texts of the health register ($F = 0.8358$ vs. $0.8300$).

Table 3.2: *Training sample for the Simple and BIO tagging modes*

| **Token** | **PoS** | **Simple** | **BIO** |
|---|---|---|---|
| There | EX | NA | NA |
| are | VBP | NA | NA |
| a | DT | NA | NA |
| couple | NN | NA | NA |
| of | IN | NA | NA |
| scenes | NNS | NA | NA |
| **that** | WDT | SSEV | SSEV |
| involve | VBP | NA | SSEV |
| sex | NN | NA | SSEV |
| in | IN | NA | SSEV |
| **that** | DT | SPECIAL | SPECIAL |
| show | NN | NA | SPECIAL |
| **but** | CC | CEV | CEV |
| they | PRP | NA | CEV |
| focus | VBP | NA | CEV |
| on | IN | NA | CEV |
| the | DT | NA | CEV |
| faces | NNS | NA | CEV |
| . | . | NA | CEV |

### 3.2.3 Feature Representation of Tokens

Two types of representation of training instances were tested. In the first (*core*), tokens were represented by evaluating three sets of feature templates. That is, by automatically determining or extracting, for each token, the values of these feature templates:

1. Unigrams consisting of:

    - the orthographic form of the token being tagged,
    - the orthographic form and the PoS, in combination, of the token being tagged,

2. Bigrams consisting of:

    - the PoS of the token being tagged and the following token,

3. Trigrams consisting of:

    - the PoS of the preceding token, the token being tagged, and the following token,
    - the PoS of the token being tagged, and the following two tokens.

The *CRF++* package was used to derive the sequence tagging model (Kudo, 2005). Tokens in the training data were represented using a set of feature templates which encode an evaluation of the external observed variables. We built the *core* feature set by first evaluating a baseline sequence tagging model, derived using *CRF++*, in which tokens were represented by a single feature template specifying the orthographic form of the token being tagged. Models in which tokens were represented by a candidate feature template in isolation were then derived and evaluated. Those with superior performance to the baseline were included in the *core* feature set. This *core* set was supplemented with unigram feature templates evaluating the features I had proposed in previous work (Evans, 2011) to create an *extended* feature set. In evaluations exploiting the CRF model to tag signs in texts of the news register, use of the *extended* feature set was found to be more accurate than use of the *core* feature set (*acc* of 0.8058 vs. 0.7846).

### 3.2.4 Training Data

We were also interested in variation in the performance of the sign tagger as a result of a mismatch between the register of the text being tagged and the register of the text from which training data was derived. In every case, there was a considerable reduction in accuracy when training data of one register was used to build models tagging signs in text of a different register. Table 3.3 presents the evaluation results of this experiment. The main diagonal displays results obtained using ten-fold cross validation. We conducted a comparative evaluation of sequence taggers exploiting training data of a register matching that of the testing data with taggers exploiting training data derived by combining instances belonging to all three registers (ensuring complementarity with test instances). This experiment showed that training a single tagging model on the entire multi-register dataset yields slightly better performance ($acc = 0.8250$) than models trained on data derived from texts matching the register of the input ($acc = 0.8196$).

Table 3.3: *Cross-register $F_1$-score performance of the tagging models (BIO tagging mode)*

| | **Test register** | | |
|---|---|---|---|
| **Train register** | **news** | **health** | **literature** |
| news | **0.7991** | 0.6129 | 0.7148 |
| health | 0.4875 | **0.8244** | 0.5195 |
| literature | 0.6403 | 0.5644 | **0.8383** |

The sign tagger exploited by the approaches to sentence simplification presented in this thesis uses a CRF sequence tagging model, running in the *BIO* tagging mode, using the *extended* feature set to represent instances, and exploiting training data derived from texts of all three registers. A detailed evaluation of the sign tagger and its suitability for use in the sentence transformation task is presented in Sections 3.3 and 5.2.4.

## 3.3 Evaluation of the Sign Tagger

Table 3.4 shows the results of testing the performance of our sign tagger in texts of all three registers, using ten-fold cross-validation. *Register* is the register of the text data being tagged. Columns $P$, $R$, and $F_1$ display the precision, recall, and $F_1$-score statistics obtained by the tagger. *Signs* is the total number of signs of syntactic complexity in the test data, *Corr* is the number of signs tagged correctly while *Incorr* is the number tagged incorrectly. Accuracy (*Acc*) is the ratio of *Corr* to the sum of *Corr* and *Incorr*. Column *Bsln* displays the accuracy of a baseline classifier which tags signs with the class labels most frequently observed for signs of each type in the annotated corpus presented in Section 2.2.3.1.

Table 3.4: *Evaluation results of the sign tagger for text of three registers*

| **Register** | **P** | **R** | $F_1$ | **Signs** | **Corr** | **Incorr** | **Acc** | **Bsln** |
|---|---|---|---|---|---|---|---|---|
| Health | 0.841 | 0.824 | 0.830 | 10 796 | 8900 | 1896 | 0.824 | 0.422 |
| Literature | 0.860 | 0.838 | 0.847 | 11 204 | 9392 | 1812 | 0.838 | 0.387 |
| News | 0.816 | 0.799 | 0.805 | 12 718 | 10 163 | 2555 | 0.799 | 0.393 |

My analysis focuses on performance statistics obtained when tagging signs in

texts of the news register but these findings also apply to the sign tagging of texts of other registers (health and literature). Table 3.5 shows the *per tag* performance of the sign tagger. The column *Support* displays the number of signs assigned each tag by human annotators in the training data. The column *Cumulative Frequency* displays the percentage of training instances assigned the current tag and the tags preceding it in the table. To illustrate, 42.81% of the training data consists of signs tagged SSEV, CMV1, and CMN1. Statistics are displayed for the 14 most frequently occurring tags in the test data. The lower part of the table displays statistics micro-averaged over the fourteen most frequently occurring tags (*Top 14*), the 26 least frequently occurring tags (*Bottom 26*), and all the tags (*All*) in the test data. Inspection of the micro-averaged statistics reveals that the predictions have a good balance between precision and recall. There is more variance when looking at performance over specific tags or signs. For example, sign tagging is accurate for some tags (*e.g.* SSEV, SSCM, SSMA and ESCM), with $F_1 > 0.9$. Most of these tags mark the left boundaries of subordinate clauses. Other tags, despite occurring with comparable frequency, are more difficult to assign (*e.g.* CMN1, ESEV, ESMP, ESMN, and ESMA) and the tagger is substantially less accurate in tagging them ($F_1 < 0.7$). Signs with these tags serve as the right boundaries of subordinate clauses, suggesting that identification of the ends of clauses is more difficult than identification of their beginnings. This is especially true of the right boundaries of multiply-embedded clauses, where one sign serves as the right boundary of several clauses. This influences the accuracy of the sentence transformation process (see Section 5.2.4).

In my annotation scheme, signs can be labelled with only one tag. As a result, the human annotators were instructed to tag these signs as the right boundaries or coordinators of the most superordinate right-bounded constituent.

Table 3.5: *Evaluation of the sign tagger over individual tags in the register of news*

| **Tag** | **P** | **R** | $F_1$ | **Support** | **Cumulative Frequency (%)** |
|---|---|---|---|---|---|
| SSEV | 0.96422 | 0.92977 | 0.94668 | 3275 | 25.75 |
| CMV1 | 0.86180 | 0.80828 | 0.83418 | 1111 | 34.48 |
| CMN1 | 0.73812 | 0.66006 | 0.69691 | 1059 | 42.81 |
| SSMN | 0.88650 | 0.83842 | 0.86179 | 885 | 49.94 |
| CEV | 0.80708 | 0.77949 | 0.79305 | 907 | 56.90 |
| SSCM | 0.96587 | 0.97586 | 0.97084 | 580 | 61.51 |
| ESEV | 0.63830 | 0.56314 | 0.59837 | 586 | 66.07 |
| SSMA | 0.93032 | 0.95736 | 0.94365 | 516 | 70.12 |
| ESMP | 0.58577 | 0.56112 | 0.57318 | 499 | 74.05 |
| SSMP | 0.84691 | 0.81667 | 0.83152 | 420 | 77.70 |
| CLN | 0.75352 | 0.69181 | 0.72135 | 464 | 81.00 |
| ESMN | 0.59719 | 0.61005 | 0.60355 | 418 | 84.29 |
| SSMV | 0.84179 | 0.81034 | 0.82577 | 348 | 87.03 |
| ESCM | 0.92073 | 0.93789 | 0.92923 | 322 | 89.56 |
| **Micro average:** | | | | | |
| Top 14 | 0.8504 | 0.8133 | 0.8315 | 11390 | 89.56 |
| Bottom 26 | 0.4926 | 0.6769 | 0.5702 | 1328 | 10.44 |
| All | 0.7991 | 0.7991 | 0.7991 | 12718 | 100.0 |

Table 3.6 is a confusion matrix that includes statistics on the eight most frequent types of confusion made by our tagger in news texts. The tags listed in column 1 are those assigned by human annotators, ranked by frequency of confusion, while tags listed in the column headers are those assigned by our

tagger. The *TOTAL* column displays the number of signs of each class tagged by our annotators in the test data.

Two types of confusion are of direct relevance to the sentence simplification process. First, the sign tagger confuses signs coordinating noun phrases (CMN1) with signs coordinating clauses (CEV). This results in the system applying transformation rules in sentences to which they should not apply. These errors may occur because the constituents adjacent to clause coordinators are often noun phrases in the object position of the first clause and the subject position of the second clause. Second, the sign tagger frequently confuses the relative pronoun *that* as a determiner or anaphor (SPEC) rather than as the left boundary of a finite subordinate clause (SSEV). As a result, some sentences containing complex$_{\overline{RF}}$ NPs may not be simplified by the system. The sign tagger also frequently mistakes the right boundaries of finite subordinate clauses for the right boundaries of non-finite clauses, most notably prepositional (ESMP) and nominal (ESMN) clauses. As a result, the sentence simplification method is likely to make errors when identifying the right boundaries of finite relative clauses, and simplifying sentences that contain complex$_{\overline{RF}}$ NPs. As with the confusions when tagging NP coordinators as clause coordinators, there are surface similarities in the contexts of use of the signs involved in these latter types of confusion.

We presented *per sign* evaluation of the tagger in Dornescu *et al.* (2013), though the results are omitted from this thesis for brevity. When testing the system on texts of the *news* register, excellent performance was achieved when tagging the complementiser [*that*] and *wh*-words such as [*who*], [*when*] or [*which*]

Table 3.6: *Confusion matrix of the sign tagger for texts of the news register*

| K/R | CEV | CMN1 | CMV1 | ESEV | ESMN | ESMP | SPEC | SSEV | Errors |
|---|---|---|---|---|---|---|---|---|---|
| CMN1 | 0.352 | - | 0.179 | 0.029 | 0.045 | 0.029 | 0.047 | 0.012 | 351 |
| ESEV | 0.193 | 0.008 | 0.074 | - | 0.232 | 0.243 | 0.047 | 0.193 | 296 |
| ESMP | 0.023 | 0.027 | 0.018 | 0.265 | 0.173 | - | 0.031 | 0.134 | 221 |
| ESMN | 0.017 | 0.008 | 0.037 | 0.167 | - | 0.234 | 0.071 | 0.023 | 208 |
| SSEV | 0.057 | 0.016 | 0.006 | 0.110 | 0.064 | 0.084 | 0.323 | - | 202 |
| SSMN | 0.028 | 0.191 | 0.012 | 0.053 | 0.050 | 0.184 | 0.055 | 0.094 | 199 |
| CMV1 | 0.102 | 0.113 | - | 0.053 | 0.023 | 0.004 | 0.126 | 0.000 | 192 |
| CEV | - | 0.152 | 0.167 | 0.090 | 0.014 | 0.021 | 0.118 | 0.064 | 176 |
| CLN | 0.028 | 0.245 | 0.025 | 0.004 | 0.000 | 0.004 | 0.008 | 0.000 | 131 |

($F_1 > 0.95$). Due to the skewed distribution of signs, more than 83% of tagging errors were linked to the two most frequently occurring: [,] and [*and*] ($F_1 = 0.7377$ and 0.7562, respectively).

In Section 2.2.3.2, I presented agreement scores obtained by human annotators marking the class labels of signs of syntactic complexity in texts of the news register ($\kappa = 0.8$).[7] When evaluating our sign tagger in each of the ten folds, mean $\kappa = 0.7533$ (95% CI 0.7519, 0.7548). In light of this, and given the similarity of the sign tagger to human annotators in terms of classification accuracy, I would not expect that the availability of additional training data would evoke significantly improved performance from the sign tagger. I am also doubtful that performance would be dramatically improved through the use of more recent neural ML approaches or the use of more recent preprocessing tools.

With a level of accuracy similar to that of human annotators, I believe that the output of the sign tagger will be useful in the analysis step of my approach to sentence simplification. I also evaluated the accuracy of the tagger when classifying signs of specific classes which are elements in the handcrafted rule activation

[7] $\kappa = 0.7667$ for annotation of texts of all three registers.

patterns used in my approach to sentence simplification. This evaluation is presented in Section 5.2.4 of the thesis.

## 3.4 Contribution to Research Question RQ-2

The development of the sign tagger and its evaluation, described in this chapter, address research question **RQ-2**:

> *Can signs of syntactic complexity be automatically and reliably classified according to their specific syntactic functions?*

The method for sign tagging presented in Section 3.2 is fully automatic. Evaluation of the method showed that it operates with micro-averaged $F_1$-score $> 0.79$ (Table 3.5, Section 3.3). In Section 5.2.4, I present an evaluation of its accuracy when tagging signs of classes directly exploited in the handcrafted rule activation patterns used in my approach to sentence simplification. The sign tagger's $F_1$-score $= 0.9075$ when tagging signs of these classes. The level of performance achieved by our automatic approach to sign tagging thus provides an affirmative response to **RQ-2**.

# Chapter 4

## Automatic Identification of Compound Clauses and Complex Constituents

The research described in this chapter forms part of my response to research question **RQ-4**, which is concerned with a comparative evaluation between two approaches to sentence simplification: one based on handcrafted patterns and one based on machine-learned patterns to detect the spans of compound clauses and complex NPs modified by non-restrictive finite relative clauses (complex$_{\overline{RF}}$ NPs) in input texts. In this chapter, I present a new approach based on machine learning to identify the spans of compound clauses and complex constituents, including complex$_{\overline{RF}}$ NPs. Automatic identification of these units, in combination with the sign tagger presented in Chapter 3, provides a new shallow approach for sentence analysis enabling automatic detection of the coordinated conjoins of compound clauses and the relative clauses modifying complex$_{\overline{RF}}$ NPs. This analysis is the basis for implementing the rule activation patterns used in my approach to sentence simplification (Chapter 5). Evaluation of a sentence simplification method which integrates this shallow sentence analysis step (Chapter 6) completes my response to research question **RQ-4**.

The method for sentence simplification that I present in Section 5.2 of this thesis is based on a set of sentence transformation schemes triggered by automatic

detection in input sentences of various rule activation patterns. One of the main challenges in this approach is to detect the different elements of the rule activation patterns in input sentences, a process that requires some type of syntactic analysis. When simplifying Type 1 and Type 2 sentences, identification of the different elements is trivial if the locations of clause coordinators, the boundaries of finite relative clauses, and the spans of compound clauses and complex$_{\overline{RF}}$ NPs have already been determined. In this chapter, I describe a machine learning method to automatically tag compound clauses and various types of complex constituents, including complex$_{\overline{RF}}$ NPs, in input sentences.

My decision to explore this shallow data-driven approach to the identification of these constituents is motivated in two ways. First, the fact that the approach is shallow affords the advantages already discussed in relation to the sign tagger when compared with methods for sentence analysis based on full syntactic parsing. Second, if syntactic parsing is discounted as being impractical for the purpose of analysing long complex sentences (See Chapter 1, page 15), one common alternative to sentence analysis is the use of handcrafted patterns expressed in terms of tokens and parts of speech (I present a method of this type in Section 5.2.2). The data-driven approach that I present in the current chapter is attractive because it avoids the need for onerous handcrafting of patterns to determine the spans of syntactic constituents. Also, while the ongoing addition of new handcrafted activation patterns can be used to improve the recall of that approach, this can be hard to implement, it may be unreliable, and it may introduce unforeseen errors. By contrast, use of a data-driven machine learning approach means that

new tagging models can be built rapidly, with each model optimally exploiting information from the annotated data available at training time. Expansion of the manually annotated training data is more likely to produce increasingly reliable tagging models than is the expansion of the sets of handcrafted patterns. Another advantage is that annotated training data is easy to share in the research community, providing other researchers with the opportunity to exploit it for their own purposes and for the development of improved tagging methods.[1]

In this chapter, I discuss previous work related to the task of automatically identifying various types of multi-token elements in text (Section 4.1). In Section 4.2, I present a new corpus annotated with information about the spans of compound clauses and complex constituents occurring in English sentences. This includes a description of the annotation schemes and some discussion of the reliability of the annotation. Section 4.3 presents my machine learning method for identifying and categorising compound clauses and complex constituents in English sentences. Section 4.4 presents an evaluation of the tagging models, with a focus on their intrinsic accuracy. The chapter concludes with a discussion of the contribution made toward addressing research question **RQ-4** of the thesis (Section 4.5).

[1]The annotated token sequences used in the method described in Section 4.3 are available at https://github.com/in6087/STARS/tree/master/sequences. Last accessed 6th January 2020.

## 4.1 Previous Related Work

This chapter is concerned with the development of methods to automatically tag the spans of compound clauses and complex constituents in English sentences. In this context, previous relevant work has been conducted in the subfield of syntactic parsing (Goldberg, 1999). The automatic parsing of compound constituents was found to be significantly less accurate than the parsing of other types of constituent (Hogan, 2007). As a result, researchers have sought to develop methods to improve the analysis of compounds (Delisle and Szpakowicz, 1995). Numerous methods have also been proposed to identify their conjoins. Many of these are based on estimations of the orthographic (Shimbo and Hara, 2007) and semantic (Agarwal and Boggess, 1992; Cederberg and Widdows, 2003; Resnik, 1999; Rus *et al.*, 2002; Chantree *et al.*, 2005) similarities between potential conjoins in NP compounds. In this thesis, I focus on the intuitively more complex task of detecting the spans of compound clauses, potentially containing more than two conjoins.

Processing texts in the medical domain, Shimbo and Hara (2007) described a method based on identifying candidate conjoins of compound NPs and assessing the word-level edit distance between them (the number of operations required to transform one into the other). They refer to the sets of edits required to transform one conjoin into another as *alignments*. In their method, alignments between candidate conjoins are represented as paths in an edit graph and statistical approaches are used to identify potential conjoins between which there is a

low cost edit path. Their approach enables identification of compounds consisting of more than two conjoins. This is an aspect differentiating their method from other previous approaches such as that proposed by Agarwal and Boggess (1992). Due to the iterative nature of the sentence simplification algorithm I present in this thesis (Chapter 5), which is driven by sign tagging (Chapter 3), my approach is also able to process compounds consisting of more than two conjoins.

In their work, Shimbo and Hara (2007) also implemented a CRF tagging approach to identify compound NPs and their conjoins as a baseline against which to compare their method exploiting edit paths using path-based and box-based algorithms. They exploited the syntactically annotated GENIA corpus of Medline abstracts to train the CRF tagger, noting that the annotation of compounds in the Penn Treebank is insufficiently detailed for this purpose.[2] Its design for use in a domain-specific task and the fact that the model is derived from annotated data from the medical domain limits the applicability of their baseline system in other use cases. The method that I propose in this chapter is designed to process compound clauses and complex constituents in texts of multiple domains/registers.

Gómez-Rodríguez and Vilares (2018) presented a method for syntactic constituent analysis which also frames the problem as a sequence tagging task. In this context, tokens in the text are labelled with information about the nodes that dominate them in the syntactic tree structure being derived. In the tagging task,

[2]Specifically, structural information on the conjoins of compounds is often omitted in that corpus, with a "flat" analysis being provided instead.

multi-token sequences are labelled with information about the lowest common dominating node in the syntactic tree. In their experiments, Gómez-Rodríguez and Vilares (2018) train systems exploiting CRF tagging and bidirectional LSTM (BiLSTM) methods to derive parsing models that are accurate and relatively fast when compared with other syntactic analysers. They found that models derived using BiLSTMs had superior accuracy to those derived using conditional random fields. The authors report $F_1 \approx 90\%$ for their parser, but it is not clear how this varies with respect to the complexity of the constituents being tagged.

## 4.2 Compound Clauses and Complex Constituents in English Sentences

In this section, I present two datasets annotated with information about the compound clauses and complex constituents that they contain. These are used to train the machine learning method to automatically tag compound clauses and complex constituents in input texts (presented in Section 4.3).

### 4.2.1 Corpus Description

My method to identify the spans of compound clauses and complex constituents in input texts relies on the output of the sign tagger presented in Chapter 3. In the context of my approach to sentence simplification, all the stages of analysis are required to be fully automatic. Hence, for the development of annotated data to evaluate the new taggers of compound clauses and complex constituents, it was appropriate to sample new sentences from the same sources as those annotated

with information about signs of syntactic complexity (Section 2.2). For the human annotation of compound clauses and complex constituents, I used the sign tagger presented in Chapter 3 to automatically identify clause coordinators and the left boundaries of subordinate clauses in these new sentences. Certain features of the tagging models presented in Section 4.3.1 were also obtained using the automatic sign tagger. The characteristics of the sampled sentences are presented in Table 4.1.

Table 4.1: *Characteristics of the corpus annotated with information about compound clauses and complex constituents*

| **Register** | **Tokens** | **Clause Coordinators** | **Left Boundaries of Subordinate Clauses** |
|---|---|---|---|
| Health | 71 258 | 1 866 | 2 627 |
| Literature | 40 953 | 1 134 | 1 290 |
| News | 68 728 | 1 105 | 2 359 |
| Total | 180 939 | 4 105 | 6 276 |

The training and validation datasets discussed in Sections 4.2.4 and 4.4 were derived from this annotated corpus.

### 4.2.2 Corpus Analysis

Two datasets were constructed by random selection of 4256 sentences containing automatically detected clause coordinators and 5302 sentences containing automatically detected left boundaries of subordinate clauses from the development dataset. The dataset containing Type 1 sentences comprised 2163 sentences for training and 2093 for validation, drawn from texts of each of three registers (health, literature, and news). The dataset containing sentences with complex

constituents (including Type 2 sentences) comprised 2674 sentences for training and 2628 for validation, drawn from each of the three registers. Our evaluation of the sign tagger when processing texts of the news register (Table 3.5, Chapter 3) showed that it identifies clause coordinators quite accurately ($F_1 = 0.7930$) and the left boundaries of subordinate clauses even more accurately ($F_1 = 0.9467$). As a result, the automatic selection of sentences for annotation is rapid and, with the exception observed in the *Errors* row of Table 4.4 (Section 4.2.4), relatively reliable.

### 4.2.3 Description of the Annotation

When annotating Type 1 sentences, annotators were required to indicate the first and last words in the compound clause. In example (23), the first occurrence of the word *you* would be annotated as the first word and *irritable* would be annotated as the last word in the compound clause.

(23) For example, you may not sleep well or you may become irritable because you have frequent hot flushes, and not directly because of a low oestrogen level.

Manual annotation of Type 1 sentences thus yields information about the automatically identified clause coordinator and the manually annotated span of the compound clause. From this markup, the annotation tool implements a variant of inside, outside (IO) encoding in which every token in the sequence receives a tag to indicate whether it occurs before the compound clause, within the compound

clause but before the sign, within the compound clause,[3] within the compound clause but after the sign, or after the compound clause. The annotators were also asked to indicate when the sign of syntactic complexity triggering the annotation had been incorrectly tagged and to indicate when they were unable to decide on the class to which the sequence belongs. As a result, the total number of classes used in this scheme is seven:

1. BEFORE_COMPOUND_CLAUSE,
2. IN_COMPOUND_CLAUSE_BEFORESIGN,
3. IN_COMPOUND_CLAUSE,
4. IN_COMPOUND_CLAUSE_AFTERSIGN,
5. AFTER_COMPOUND_CLAUSE,
6. NOT_CLAUSE_COORDINATOR, and
7. UNDECIDED.

This annotation produced a data file suitable for use by automatic sequence tagging methods such as *CRF++* (Kudo, 2005). Table 4.2 presents the annotation prompted by the occurrence of a clause coordinator in Sentence (23).

When describing the manual annotation of sentences containing subordinate clauses in the second dataset, it should be noted that English sentences may include various types of complex constituent (e.g. adjectival phrases, adverbial

[3] i.e. The token is the clause coordinator.

Table 4.2: *Annotated sentence containing a compound clause.*

| **Position in sequence** | **Token** | **PoS** | **Token number** | **Class label** |
|---|---|---|---|---|
| 1 | For | IN | 582_1 | BEFORE_COMPOUND |
| 2 | example | NN | 582_2 | BEFORE_COMPOUND |
| 3 | , | ESMP | 582_3 | BEFORE_COMPOUND |
| 4 | you | PRP | 582_4 | IN_COMPOUND_BEFORESIGN |
| 5 | may | MD | 582_5 | IN_COMPOUND_BEFORESIGN |
| 6 | not | RB | 582_6 | IN_COMPOUND_BEFORESIGN |
| 7 | sleep | VB | 582_7 | IN_COMPOUND_BEFORESIGN |
| 8 | well | RB | 582_8 | IN_COMPOUND_BEFORESIGN |
| 9 | [or] | CEV | 582_9 | IN_COMPOUND |
| 10 | you | PRP | 582_10 | IN_COMPOUND_AFTERSIGN |
| 11 | may | MD | 582_11 | IN_COMPOUND_AFTERSIGN |
| 12 | become | VB | 582_12 | IN_COMPOUND_AFTERSIGN |
| 13 | irritable | JJ | 582_13 | IN_COMPOUND_AFTERSIGN |
| 14 | because | IN | 582_14 | AFTER_COMPOUND |
| 15 | you | PRP | 582_15 | AFTER_COMPOUND |
| 16 | have | VBP | 582_16 | AFTER_COMPOUND |
| 17 | frequent | JJ | 582_17 | AFTER_COMPOUND |
| 18 | hot | JJ | 582_18 | AFTER_COMPOUND |
| 19 | flushes | NNS | 582_19 | AFTER_COMPOUND |
| 20 | ,_and | CMP | 582_20 | AFTER_COMPOUND |
| 21 | not | RB | 582_21 | AFTER_COMPOUND |
| 22 | directly | RB | 582_22 | AFTER_COMPOUND |
| 23 | because | IN | 582_23 | AFTER_COMPOUND |
| 24 | of | IN | 582_24 | AFTER_COMPOUND |
| 25 | a | DT | 582_25 | AFTER_COMPOUND |
| 26 | low | JJ | 582_26 | AFTER_COMPOUND |
| 27 | oestrogen | NN | 582_27 | AFTER_COMPOUND |
| 28 | level | NN | 582_28 | AFTER_COMPOUND |
| 29 | . | . | 582_29 | AFTER_COMPOUND |

phrases, noun phrases, and verb phrases), often in the same sentence.[4] Furthermore, even when considering complex NPs in isolation, these may be modified by

[4]In the examples provided here, the spans of complex constituents, including complex NPs and compound clauses are marked using square brackets. In some examples, constituents may be multiply embedded. Subordinate clauses and the conjoins of compound clauses are underlined. In examples (24)–(28), only one complex noun phrase in each sentence is bracketed.

various types of finite and non-finite clause (e.g. (24)–(28)).

(24) The County Court in Nottingham heard that [Roger Gedge, 30], had his leg amputated following the incident outside a rock festival in Wollaton Park, Nottingham, five years ago.

(25) [Tomkins, who married at 20], told police that her husband was a domineering man who drank and was careless with money.

(26) They were asked by [David Price, solicitor advocate for Mr Burstein], to award damages of between £20,000 and £50,000.

(27) When [the mum of two, from Cheltenham, Glos], heard a rear window break, she feared for her life and accelerated at 40 mph, trapping Gedge.

(28) "The evidence is so thin, it is effectively invisible," said [Gareth Peirce, representing Eidarous].

One of the aims of my sentence simplification method (Section 5.2.1) is to simplify Type 2 sentences containing complex$_{\overline{RF}}$ NPs, such as (25)). While the sign tagger that we developed (Section 3.2) is accurate, it provides no additional information on the particular functions of the subordinate clauses detected in a sentence. For this reason, human annotators were required to encode information on the syntactic functions of each of the identified subordinate clauses. They were instructed to indicate the first and last words in the complex constituent that the

clause modifies and also to label its function. For this purpose, the annotation scheme includes ten different functions which are available for selection:

- **Adjectival**: The clause modifies an adjectival phrase (29).

  (29) "I am [very pleased that we don't]."

- **Adverbial**: The clause is adverbial or modifies an adverbial (30).

  (30) They were walking home from a party [when he was attacked by Aaron Lee Martin, 25].

- **Cleft**: the clause is used in a cleft, pseudo-cleft,[5] or inverted pseudo-cleft construction[6] (31).

  (31) On the other hand, altering it would be costly - £220m according to the RCN but rising rapidly as more old people enter the homes over the next two decades - and [it is the better off who would benefit, by saving their homes to hand on to their children].

- **Independent**: The clause is independent (32).

  (32) [Means-testing rules are tightly enforced: anyone who transfers

[5] E.g. [That people can love one another was the message he wanted to express].
[6] E.g. [This is the first time in the city's history that it has achieved a figure this high].

ownership of their house to try to escape paying will still be made to pay if they need to go into a home within seven years].

- **Intensifying**: The clause is used in an intensifying construction (33).

(33) Some are [so aroused by the feisty character, depicted wearing tight-fitting clothing, that they have put nude versions of her on the internet].

- **Reporting**: The clause is a reporting clause. (34)

(34) A consultant gynaecologist nibbled at a patient's ear then told her it was all part of the treatment, a Manchester court was told yesterday.

- **Restrictive**: The clause is restrictive and modifies an otherwise generic or non-specific noun phrase (35).

(35) "It has been incredibly traumatic and is [something that I will never forget]."

- **Verbal**: The clause is the obligatory argument of a clause complement verb (36).

(36) Penelope Tomkins, 49, who [claimed that she could not end the affair but was unable to leave her husband for fear of losing the love of her two grown-up children], was jailed for 3 years after admitting soliciting to murder between June 28 and October 28 last year.

- **Wh-clause**: The clause modifies another type of *wh*-phrase, but not nominally bound *which* and not an adverbial (37).

(37) "Obviously, no one would have wished what happened to Mr Gedge but perhaps now in hindsight he will realise if he hadn't done something as obscene in the first place, it would never have happened."

- **Non-nominal**: The clause is bound but not nominally bound. i.e. the superordinate phrase is not nominal and the clause does not belong to any of the classes previously listed (38).

(38) In Graves' disease the thyroid gland [usually enlarges, which causes a swelling (goitre) in the neck].

The annotation of sentences containing complex constituents was made in a similar way to the annotation of sentences containing compound clauses. The annotators were required to identify the first and last word in each complex constituent or subordinate clause in the corpus to be annotated. To illustrate, they were required to identify the first and last word in each of the square bracketed

sequences in examples (29), (31), (33), (35)–(36), and (38). In examples (30), (32), (34), and (37), they were required to annotate the first and last word of each of the underlined sequences. In both cases, they were also required to indicate the function of the subordinate clause. The annotators were also asked to indicate cases in which the sign of syntactic complexity triggering the annotation had been incorrectly tagged and to indicate when they were unable to decide on the class to which the sequence belongs.

Manual annotation of these sentences thus yields information about the automatically identified left boundary of the subordinate clause, the manually annotated span of the complex constituent, and the manually annotated function of the clause. As in the case when annotating Type 1 sentences, the annotation tool implements a variant of IO encoding in which every token in the sequence receives a tag to indicate whether it occurs before the complex constituent, within the complex constituent but before the sign, within the complex constituent (and is the sign), within the complex constituent but after the sign, or after the complex constituent.

As a result, the total number of classes used in this scheme is 54, with examples including BEFORE_ADJECTIVAL, IN_COGNITIVE_COMMUNICATIVE_VP_BEFORESIGN, IN_INTENSIFYING_CLAUSE, IN_RESTRICTIVE_CLAUSE_AFTERSIGN, and AFTER_ADVERBIAL. To illustrate, Table 4.3 displays the annotation of Sentence (39).

(39) The inquiry, which continues, will recall Dr Wisheart and Dr Roylance

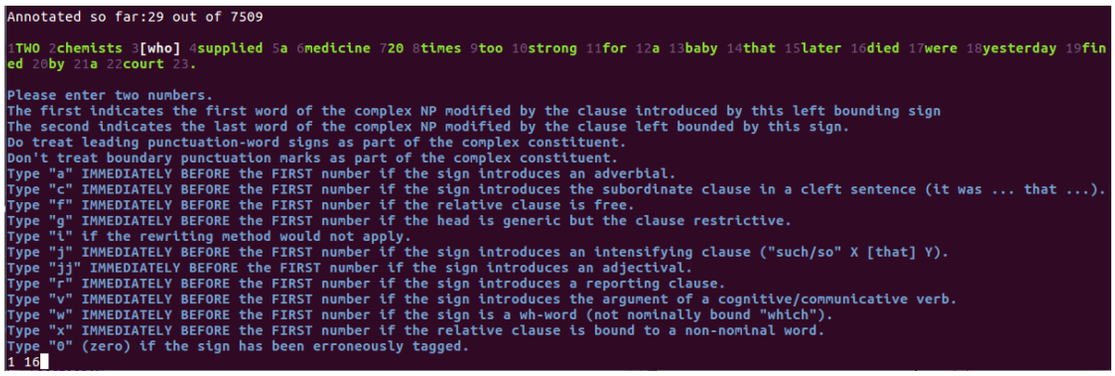



Figure 4.1: *Interface used for manual annotation of complex constituents*

next month for further questioning.

Both types of annotation were made using an annotation tool which displays the input sentence with the triggering sign of syntactic complexity highlighted in square brackets, the token numbers of words, and brief instructions and information about the annotation scheme. The highlighting of the triggering sign facilitates accurate annotation of sentences containing multiple clause coordinators or multiple left boundaries of subordinate clauses. In the latter case, for example, annotators were instructed only to annotate the complex constituent modified by the specific subordinate clause introduced by the highlighted sign. Figure 4.1 displays the interface of this tool.[7]

### 4.2.4 Analysis of the Annotation

Table 4.4 provides information on the characteristics of Type 1 sentences annotated in the corpus. This table also displays the numbers of sentences an-

[7]The interface used to annotate compound clauses is essentially identical, except that it provides a far smaller number of class labels.

Table 4.3: *Annotated sentence containing a complex noun phrase.*

| **Position in sequence** | **Token** | **PoS/Sign Tag** | **Token number** | **Class label** |
|---|---|---|---|---|
| 1 | The | DT | 2_1 | IN_COMPLEX_NP_BEFORESIGN |
| 2 | inquiry | NN | 2_2 | IN_COMPLEX_NP_BEFORESIGN |
| 3 | [,_which] | SSEV | 2_3 | IN_COMPLEX_NP |
| 4 | continues | VBZ | 2_4 | IN_COMPLEX_NP_AFTERSIGN |
| 5 | , | ESEV | 2_5 | AFTER_COMPLEX_NP |
| 6 | will | MD | 2_6 | AFTER_COMPLEX_NP |
| 7 | recall | VB | 2_7 | AFTER_COMPLEX_NP |
| 8 | Dr | NNP | 2_8 | AFTER_COMPLEX_NP |
| 9 | Wisheart | NNP | 2_9 | AFTER_COMPLEX_NP |
| 10 | and | CMN1 | 2_10 | AFTER_COMPLEX_NP |
| 11 | Dr | NNP | 2_11 | AFTER_COMPLEX_NP |
| 12 | Roylance | NNP | 2_12 | AFTER_COMPLEX_NP |
| 13 | next | JJ | 2_13 | AFTER_COMPLEX_NP |
| 14 | month | NN | 2_14 | AFTER_COMPLEX_NP |
| 15 | for | IN | 2_15 | AFTER_COMPLEX_NP |
| 16 | further | RB | 2_16 | AFTER_COMPLEX_NP |
| 17 | questioning | VBG | 2_17 | AFTER_COMPLEX_NP |
| 18 | . | . | 2_18 | AFTER_COMPLEX_NP |

notated (*Sequences*) and compound clauses identified (*Compound clause*). For annotation by two annotators of 200 compound clauses in the register of news, $\kappa = 0.8313$. This indicates 'almost perfect' agreement between the annotators (Viera and Garrett, 2005).

Table 4.4: *Compound clauses in the training data*

| **Type** | **Health** | **Literature** | **News** |
|---|---|---|---|
| Compound clause | 475 | 501 | 485 |
| Unknown | 19 | 15 | 1 |
| Errors | 508 | 98 | 61 |
| Sequences | 1 002 | 614 | 547 |
| Tokens | 23 999 | 25 310 | 16 236 |
| Avg. sequence length (tokens) | 24 | 42 | 30 |

The *Errors* row of Table 4.4, indicates that a relatively large number of se-

quences (508) from texts of the health register that were presented to annotators did not contain compound clauses. Investigation of the performance of the sign tagger revealed that it is slightly less accurate when identifying clause coordinators in texts of this register ($F_1 = 0.7573$). This lack of accuracy does not affect the quality of the human annotation. However, it does mean that annotators are presented with a larger proportion of sequences containing no compound clauses when annotating texts of the register of health than is the case for texts of the other registers. Inclusion within the annotation scheme of the NOT_CLAUSE_COORDINATOR class is intended to facilitate detection of erroneous sign tagging by data-driven approaches exploiting this dataset. Inspection of sentences tagged with this label in the health register revealed that the majority contain signs that are the right boundaries of adverbial clauses (tagged ESMAdv), often conditionals, which had been misclassified as clause coordinators. A large number of other types of confusion were also in evidence.

Table 4.5 provides information on the characteristics of sentences containing complex constituents in the second dataset. It also displays the numbers of sentences annotated (*Sequences*) and the frequency distribution of the different functions of subordinate clauses in complex constituents. The *Errors* row displays the number of automatically detected signs triggering annotation that were not left boundaries of subordinate clauses. For annotation by two annotators of 200 sentences containing complex constituents, $\kappa = 0.8255$. Again, this indicates 'almost perfect' agreement between annotators.[8]

[8]The assessments of inter-annotator agreement presented in this section were made over all

Table 4.5: *Complex constituents in the training data*

| **Type** | **Health** | **Literature** | **News** |
|---|---|---|---|
| Complex$_{\overline{RF}}$ NP | 179 | 144 | 355 |
| Adjectival | 3 | 9 | 2 |
| Adverbial | 130 | 143 | 138 |
| Cleft | 16 | 17 | 15 |
| Independent | 8 | 24 | 2 |
| Intensifying | 1 | 25 | 8 |
| Reporting | 0 | 5 | 48 |
| Restrictive | 179 | 30 | 100 |
| Clause Comp. Verb | 77 | 102 | 252 |
| *Wh*-phrase | 45 | 46 | 37 |
| Non-nominal | 2 | 0 | 1 |
| Unknown | 18 | 8 | 7 |
| Errors | 343 | 114 | 39 |
| Sequences | 1 002 | 667 | 1 005 |
| Tokens | 23 208 | 25 518 | 31 618 |
| Avg. sequence length (tokens) | 23 | 38 | 31 |

Inspection of the data annotated in accordance with the scheme described in Section 4.2.3 (pages 85–90) provides insights into the relative frequencies of different functions of subordinate clauses in texts of different registers. In the register of health, there is a relatively large proportion of restrictive relative clauses (17.86%) and clauses with an adverbial function (12.97%). Clauses with an adverbial function are also frequent in literary texts (21.4%). In texts of the news register, the most frequent class of subordinate clause is that of non-restrictive nominally bound relative clauses modifying complex$_{\overline{RF}}$ NPs (35.32%). A large proportion comprise the obligatory arguments of clause-complement verbs (25.07%). Texts of the health and literary registers also contain relatively large numbers of non-

tokens occurring in each sequence. It is possible that agreement scores would differ if only the tokens at constituent boundaries were considered.

restrictive nominally bound relative clauses (17.86% and 21.59%, respectively). I would therefore expect a sentence simplification method designed to reduce the number of complex$_{\overline{RF}}$ NPs in input sentences to be most impactful when processing texts of the register of news, followed by literature, and health.

## 4.3 A Machine Learning Method to Tag Compound Clauses and Complex Constituents

The approach to sentence simplification that I present in this thesis is based on the use of sentence transformation rules triggered by matching input sentences against various rule activation patterns (Section 5.2.1). Matching of the patterns to natural language text requires accurate detection of various elements, including the spans of compound clauses and complex$_{\overline{RF}}$ NPs. Identification of the latter also depends on discrimination between complex$_{\overline{RF}}$ NPs and various other types of complex constituent. In this section, I describe a new approach based on sequence tagging to automatically tag compound clauses and complex constituents in English sentences.

The purpose of the methods described in this chapter is to extend the automatic shallow syntactic analyses produced by the sign tagger presented in Chapter 3. While the sign tagger detects and categorises tokens and token bigrams (signs of syntactic complexity), the two methods presented in the current chapter detect and categorise token ngrams of arbitrary size: compound clauses and complex constituents. More specifically, one method is intended, for each sign tagged as a clause coordinator (tagged CEV), to identify the neighbouring tokens

comprising the compound clause in which the sign occurs. The other method is intended, for each sign tagged as the left boundary of a subordinate clause (tagged SSEV), to identify and categorise the neighbouring tokens comprising the complex constituent in which the sign occurs. These methods can then be applied to extend the shallow syntactic analysis performed by the sign tagger by detecting the spans of compound clauses and complex$_{\overline{RF}}$ NPs for the purpose of sentence simplification.

Sequence tagging models exploiting conditional random fields (CRF) have been used successfully in previous work for a variety of tagging and chunking tasks in NLP, including named entity recognition (McCallum and Li, 2003), shallow parsing (Sha and Pereira, 2003), and clause boundary identification (Lakshmi *et al.*, 2012). For this reason, I developed a method using the *CRF++* package (Kudo, 2005) to identify the spans of compound clauses and complex constituents in English sentences. This type of sentence analysis will greatly facilitate automatic detection of the elements of the rule activation patterns used in my approach to sentence simplification (Section 5.2.1). Output of the tagger that I developed resembles that displayed in columns *Token* and *Class label* in Tables 4.2 (p. 84) and 4.3 (p. 91) of the current chapter.

Sentence structure is hierarchical and may include compounds with complex conjoins, complex constituents modified by compound clauses, and multiply embedded complex constituents. By contrast, sequence labelling approaches are most suitable for tagging non-hierarchical contiguous chunks of text. The models that I derived based on sequence tagging generate a flat representation (i.e.

a maximum of one compound clause or one complex constituent identified in each input sentence). As a result, when integrated into a sentence simplification method processing complex hierarchical structures, these models must be applied repeatedly to tag input token sequences in an iterative sentence simplification algorithm (Section 5.2.1).

I used *CRF++* to derive two sequence labelling models, one for each of the tasks of tagging compound clauses and complex constituents in input sentences. For this purpose, each set of training data was represented as a set of token sequences, with each sequence corresponding to a sentence in the text. I developed a feature extraction tool to represent tokens in the training and validation datasets as vectors encoding various types of linguistic information.

### 4.3.1 Token Features

The feature extraction tool derives the values of 39 features of tokens occurring in input sequences corresponding to sentences. I designed the initial pool of features to encode information about the intrasentential linguistic context of each token. This included features intrinsic to the token such as its orthographic form and part of speech and information about its relationship to other tokens in the sequence. It was necessary to engineer features of this type due to the relatively limited size of my dataset, which restricted the ability of the machine learning method to derive even quite limited information about the contexts of tokens and the relationships holding between tokens of different types. For brevity, I do not list the 39 features here, but the full feature set is presented in Appendix C.

In addition to the training data described in Section 4.2.4, validation datasets were also developed for optimisation of the machine learning methods. For the models to tag sentences containing compound clauses, the validation set comprised 2093 sequences while, for models to tag complex constituents, the validation set comprised 2628 sequences. In both cases, the token sequences were from texts of the registers of health, literature, and news. Optimisation was performed using naïve hill climbing and grid search methods to assess the suitability of features in the pool and other parameters for use in the CRF sequence labelling models. When selecting features for the tagging of complex constituents, evaluation was based on the $F_1$-score obtained for classification of sequences involving complex$_{\overline{RF}}$ NPs (as opposed to other types of complex constituent).

Table 4.6 indicates the set of features selected for classification of tokens both in sequences containing compound clauses and in sequences containing complex constituents. This is the set of features exploited when learning the most accurate models for tagging input sequences in accordance with the annotation schemes detailed in Section 4.2.3. In the evaluations performed for feature selection, the CRF tagger was trained using data from all three text registers (health, literature, and news) at once and validated on data from these three registers.

Tables 4.8 and 4.9 list additional features from the initial pool that were selected for inclusion in the models to classify tokens in sequences containing compound clauses and complex constituents, respectively. For each of the two tagging tasks, the features listed in Tables 4.8 and 4.9 bring additional gains in the accuracy of the models when added to the set of features listed in Table 4.6.

Table 4.6: *Features selected for tagging of both compound clauses and complex constituents*

| | |
|---|---|
| Boolean | Token has a part of speech matching that of the first token following the next sign of syntactic complexity |
| | Token is the word *when* |
| | Token is a colon |
| | Token is a final/illative conjunction (see Table 4.7 for an indicative list of such conjunctions) |
| Ternary | Position of the token in the sentence: `FIRST_THIRD`, `SECOND_THIRD`, or `THIRD_THIRD` |
| Numeric | Number of words between token and the next word with part of speech tag IN |
| | Number of words between token and the next word with part of speech tag VBD |
| | Number of words between token and the next sign of syntactic complexity |
| | Number of verbs that precede the token in the sentence |
| Symbolic | The token |
| | Part of speech of the token or class label, if the token is a sign of syntactic complexity |
| | Part of speech of the first word in the sequence |

Table 4.7: *Final/illative conjunctions*

| | |
|---|---|
| hence | in consequence |
| of course | so that |
| so then | therefore |
| thus | |

Table 4.8: *Additional features selected for tagging of compound clauses*

| | |
|---|---|
| Boolean | Part of speech of token matches that of the first word in the sequence |
| | Token matches the first lexical word in the sequence |
| | Token is verbal (part of speech is in the set {VB, VBG, VBN, or RB}) |
| | Token is the word *some* |
| Ternary | Token is a coordinator: YES (*and*, *but*, or *or*), MAYBE (a punctuation mark followed by *and*, *but*, or *or*), or NO (any other token) |
| Numeric | Position of the token in the document |
| Symbolic | Acoustic form of the token (in the token, consonant clusters are rendered `C`, single consonants `c`, vowel sequences as `V`, and single vowels as `v`. The word *consonant* is thus rendered as `cvCvcvC` |

Table 4.9: *Additional features selected for tagging of complex constituents*

| | |
|---|---|
| Boolean | Token is a relative pronoun (wh-word or *that*) |
| | Sentence in which the token appears also contains a clause complement word[9] (see Table 4.11 for an indicative list of such words) |
| | Token is the word *who* and subsequent tokens include a comma immediately followed by a past tense verb (PoS is VBD) |
| | Token is either *that* or *which* and subsequent tokens include a comma immediately followed by a determiner (PoS is DT) |
| | Token is an adversative conjunction (see Table 4.10 for an indicative list of such conjunctions) |
| Quinary | Token's relationship to the word *because*: INDEPENDENT, PRECEDES, FOLLOWS, BOTH_PRECEDES_AND_FOLLOWS, or IS the word *because* |
| Numeric | Number of commas in the same sentence as the token |
| | Number of signs of syntactic complexity in the same sentence as the token |

Table 4.10: *Adversative conjunctions*

| | | | | | |
|---|---|---|---|---|---|
| although | contrariwise | conversely | despite | however | instead |
| nevertheless | nonetheless | though | whereas | while | yet |

When deriving the models, tokens were represented using the three sets of feature templates presented in Section 3.2.3.[10] For the model used to tag compound clauses, templates were included for all of the features listed in Tables 4.6 and 4.8. For the model used to tag complex constituents, templates were included for all of the features listed in Tables 4.6 and 4.9. These templates were 5-grams, used to condition the tagging of each token on the basis of information about the value of the feature in the two preceding tokens, the token being tagged, and the two following tokens.

[9]This includes morphological variants such as the past, present, and -ing forms of clause complement verbs. This footnote pertains to the first portion of Table 4.9.

[10]In CRF++, feature selection is implemented via the content of the feature template file. Only features associated with a template are exploited by the derived tagging models.

Table 4.11: *Clause complement words.*

| | | **Verbs** | | |
|---|---|---|---|---|
| accept | acknowledge | add | admit | agree |
| allege | announce | answer | appreciate | argue |
| ask | aware | believe | certain | claim |
| clear | complain | concern | conclude | confirm |
| convince | decide | demonstrate | deny | disappoint |
| disclose | discover | doubt | dread | emerge |
| emphasise | ensure | establish | expect | explain |
| fear | feel | find | given | guess |
| hear | hold | hope | illustrate | indicate |
| infer | insist | intimate | imply | know |
| learn | maintain | mean | note | order |
| plain | possible | promise | protest | prove |
| provide | record | realise | recognise | recommend |
| read | realise | record | relate | remain |
| report | retort | reveal | rule | satisfy |
| saw | say | see | show | state |
| suggest | suspect | tell | terrified | testify |
| think | warn | | | |
| | | **Nouns** | | |
| allegation | admission | belief | manner | scale |
| view | way | | | |
| | | **Adjectives** | | |
| disappointed | obvious | | | |

Identification of the sequences (sentences) to be tagged using these models depends on accurate detection of signs which coordinate clauses in compounds (tagged CEV) and which serve as the left boundaries of subordinate clauses (tagged SSEV). For this reason, the sign tagger described in Chapter 3 of this thesis is of central importance in this approach to tagging compound clauses and complex constituents.

Ablation of each selected feature from the derived CRF models indicated

that several features were particularly useful, with ablation negatively affecting accuracy by more than 1%. Table 4.12 lists these features and the effects of their ablation on the accuracy of the models.

Table 4.12: *Features for which ablation has the greatest adverse effect on accuracy of derived tagging models*

| **Feature** | $\Delta F_1$ **(negative)** |
|---|---|
| Tagging compound clauses | |
| Orthographic form | 0.0257 |
| Distance to sign | 0.0214 |
| Acoustic form | 0.0155 |
| Tagging complex constituents | |
| Orthographic form | 0.0376 |
| Distance to sign | 0.0201 |
| Sign is *when* | 0.0195 |
| Sign is a relative pronoun | 0.0147 |
| PoS/sign tag | 0.0101 |

Of the tagging models, the bigram model performed best. The feature encoding information from the sign tagger (*PoS/sign tag*) is ranked fifth in terms of its contribution to models tagging sentences which contain complex constituents and, although it is not listed in Table 4.12 because the negative change in $F_1 < 0.01$ (It is 0.0095), it is ranked fourth for models tagging Type 1 sentences. Other linguistic features brought minor improvements in performance, and were also included in the models. Table 4.13 displays micro-averaged $F_1$ scores obtained by the taggers using different combinations of features.

Experiments in which the classification of tokens in the training and validation datasets was extended, using variants of the BIO scheme, did not lead to the

Table 4.13: *Performance of the taggers when exploiting different combinations of features*

| | $F_1$ **(micro-averaged, all registers)** | |
|---|---|---|
| **Features** | **Compound Clauses** | **Complex Constituents** |
| Orthographic form | 0.4893 | 0.2577 |
| Orthographic form and PoS/sign tags | 0.5041 | 0.2716 |
| All but PoS/sign tags | 0.7186 | 0.5391 |
| All | 0.7281 | 0.5492 |

derivation of more accurate tagging models.

## 4.4 Evaluation of the Taggers

In this section, I present an empirical evaluation of the models developed to identify the spans of compound clauses (Section 4.4.1) and complex constituents (Section 4.4.2) in input texts.

### 4.4.1 Tagging Evaluation: Compound Clauses

Tagging of the spans of compound clauses was evaluated by comparison with a validation dataset, manually annotated using the scheme presented in Section 4.2.3 (pages 82–83). Characteristics of the validation dataset are presented in Table 4.14. The upper and middle row sections display the numbers of sequences of each type for each text register. As noted in Section 4.2.4, the *Errors* row provides statistics on the number of sequences erroneously presented to annotators because of sign tagging errors. These are not the numbers of errors made by annotators when tagging compound clauses.

In addition to the tagging method based on CRF, I implemented a base-

Table 4.14: *Characteristics of the validation dataset: token sequences containing compound clauses*

| **Type** | **Health** | **Literature** | **News** | **All** |
|---|---|---|---|---|
| Compound clause | 553 | 416 | 448 | 1 417 |
| Unknown | 3 | 4 | 6 | 13 |
| Errors | 446 | 100 | 117 | 663 |
| Sequences | 1 002 | 520 | 571 | 2 093 |
| Tokens | 24 086 | 15 409 | 17 066 | 56 561 |
| Avg. sequence length (tokens) | 24 | 30 | 30 | 28 |

line tagger which differed in terms of the machine learning algorithm used. This baseline exploited the TiMBL memory-based learning classifier (Daelemans *et al.*, 2010) and its performance was optimised using a combination of naïve manual hill-climbing and grid search for selection of features (via ablation), algorithm, feature metric, feature weighting, functions to weight neighbours with respect to their distance, and the number of neighbours considered. Optimal performance was achieved using the IB1 algorithm, the weighted overlap feature metric, information gain feature weighting, with inverse distance used as the function to weight neighbours with respect to their distance, and when making classifications on the basis of 19 nearest neighbours.

Table 4.16 presents the accuracy of the method presented in Section 4.3 when tagging the spans of compound clauses (Column $CRF$). The scores displayed in this table are averages of those obtained for the five subclasses of each class: those occurring before the compound clause, those occurring within the compound clause but before the clause coordinator, those that are the clause coordinator, those within the compound clause but occurring after the clause coordinator,

and those occurring in the same sentence but after the compound clause. The statistics displayed in Column $MBL$ are accuracy scores obtained by the baseline classifier implemented using memory-based learning. For texts of the registers of Literature and Health, the difference in overall accuracy between $CRF$ and $MBL$ is statistically significant ($p \ll 0.01$), with the approach based on CRF tagging being superior. The approach exploiting memory-based learning is superior to that using the CRF tagger when processing texts of the news register (again, $p \ll 0.01$). However, for the purpose of sentence simplification, the validity of the tag sequences output by the sentence analysis tool is as important as the overall accuracy of tagging at the token level. Specifically, I consider ten token tag sequences to be valid for the purpose of sentence analysis:

1. BEFORE_X BEFORE_X
2. BEFORE_X IN_X_BEFORESIGN
3. IN_X_BEFORESIGN IN_X_BEFORESIGN
4. IN_X_BEFORESIGN IN_X
5. IN_X IN_X_AFTERSIGN
6. IN_X_AFTERSIGN IN_X_AFTERSIGN
7. IN_X_AFTERSIGN AFTER_X
8. AFTER_X AFTER_X
9. ERROR ERROR

10. UNKNOWN UNKNOWN

Where X is a compound clause (for the task of tagging spans of compound clauses) or a type of complex constituent (for the task of tagging the spans of complex constituents). Other tag sequences, such as

BEFORE_X AFTER_X or

AFTER_X IN_X_BEFORESIGN

cannot be exploited for the purpose of syntactic analysis. Table 4.15 shows the contrasting output of the MBL and CRF tagging models when processing sentence (40).

(40) The letter created a lot of 'flak' and Mr Bolsin said he was called in to see a 'very angry' James Wisheart, the senior cardiac surgeon.

In this case, the tags assigned by the MBL method include invalid sequences (AFTER_COMPOUND immediately followed by IN_COMPOUND_AFTER-SIGN at *Token ID*s 21-22 and 24-25), which introduce ambiguity about the location of the right boundary of the compound clause. In this example, the CRF tagger introduces no such ambiguity.

The two machine learning methods are very different, with MBL classifying tokens independently of one another and CRF, as a sequence tagger, conditioned to exploit information about other tags in the sequence when classifying a given token. For this reason, it is more likely to generate valid sequences. Inspection

Table 4.15: *Output of the MBL and CRF sequence tagging methods for input sentence (40)*

| Token ID | Token | MBL | CRF |
|---|---|---|---|
| 1 | The | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 2 | letter | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 3 | created | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 4 | a | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 5 | lot | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 6 | of | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 7 | ‘ | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 8 | flak | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 9 | ’ | IN_COMPOUND_BEFORESIGN | IN_COMPOUND_BEFORESIGN |
| 10 | and | IN_COMPOUND | IN_COMPOUND |
| 11 | Mr | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 12 | Bolsin | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 13 | said | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 14 | he | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 15 | was | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 16 | called | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 17 | in | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 18 | to | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 19 | see | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 20 | a | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 21 | ‘ | AFTER_COMPOUND | IN_COMPOUND_AFTERSIGN |
| 22 | very | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 23 | angry | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 24 | ’ | AFTER_COMPOUND | IN_COMPOUND_AFTERSIGN |
| 25 | James | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 26 | Wisheart | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 27 | , | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 28 | the | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 29 | senior | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 30 | cardiac | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 31 | surgeon | IN_COMPOUND_AFTERSIGN | IN_COMPOUND_AFTERSIGN |
| 32 | . | AFTER_COMPOUND | AFTER_COMPOUND |

of Table 4.16 (*Valid tag sequences*), reveals that this is so, with the CRF tagging approach generating a far larger proportion of valid tag sequences than the approach using memory-based learning. For this reason, it is preferred for use in the analysis stage of sentence simplification.

Table 4.16: *Evaluation results ($F_1$-score) for the tagging of compound clauses in texts of the three registers*

| | **Health** | | **Literature** | | **News** | |
|---|---|---|---|---|---|---|
| **Type** | **CRF** | **MBL** | **CRF** | **MBL** | **CRF** | **MBL** |
| Compound clause | 0.7324 | 0.6114 | 0.8383 | 0.7823 | 0.7303 | 0.7637 |
| Before compound clauses | 0.4931 | 0.2322 | 0.5321 | 0.3488 | 0.5491 | 0.5131 |
| In compound clauses | 0.7613 | 0.7947 | 0.8682 | 0.9054 | 0.7554 | 0.8614 |
| After compound clauses | 0.6518 | 0.5398 | 0.7622 | 0.7732 | 0.6739 | 0.6987 |
| Unknown | 0 | 0 | 0 | 0 | 0 | 0 |
| Not clause coordinator (errors) | 0.7487 | 0.4537 | 0.6373 | 0.1893 | 0.3951 | 0.1976 |
| $F_1$ (Micro-averaged) | 0.7387 | 0.5512 | 0.7989 | 0.6898 | 0.6586 | 0.6718 |
| Valid tag sequences (%) | 98.27 | 63.00 | 99.21 | 79.81 | 98.24 | 78.45 |

Given that the tagging model built using the CRF approach will be used to facilitate the simplification of Type 1 sentences, the results obtained for identification of tokens in compound clauses is promising ($0.73 \leq F_1 \leq 0.84$, for all registers). They provide some cause for optimism about the success of a method integrating this tagging model for the analysis and simplification of sentences containing compound clauses.

### 4.4.2 Tagging Evaluation: Complex Constituents

Tagging of the spans of complex constituents was evaluated by comparison with a validation dataset, manually annotated using the scheme presented in Section 4.2.3 (pages 85–90). Characteristics of the validation dataset are presented in Table 4.17. The upper and middle row sections display the numbers of sequences of each type for each text register.

Table 4.17: *Characteristics of validation data: token sequences containing complex constituents*

| **Type** | **Health** | **Literature** | **News** | **All** |
|---:|---:|---:|---:|---:|
| Complex$_{\overline{RF}}$ NP | 117 | 87 | 324 | 528 |
| Adjectival | 2 | 12 | 6 | 20 |
| Adverbial | 121 | 155 | 131 | 407 |
| Cleft | 11 | 22 | 25 | 58 |
| Independent | 18 | 7 | 17 | 42 |
| Intensifying | 2 | 42 | 4 | 48 |
| Reporting | 0 | 11 | 77 | 88 |
| Restrictive | 226 | 65 | 40 | 331 |
| Clause Comp. Verb | 93 | 78 | 289 | 460 |
| *Wh*-phrase | 12 | 55 | 57 | 124 |
| Non-nominal | 14 | 7 | 0 | 21 |
| Unknown | 91 | 6 | 4 | 101 |
| Errors | 294 | 78 | 28 | 400 |
| Sequences | 1 001 | 625 | 1 002 | 2 628 |
| Tokens | 23 260 | 18 627 | 33 038 | 74 925 |
| Avg. sequence length (tokens) | 23 | 30 | 33 | 29 |

Inspection of these statistics indicates that a relatively large proportion of the subordinate clauses occurring in health texts are restrictive relative clauses. In the literary register, subordinate clauses with an adverbial function are more frequent. In validation data of the news register, complex$_{\overline{RF}}$ NPs are relatively

frequent, as are verb phrases with obligatory clause complements. Since the sentence simplification method proposed in this thesis is designed to simplify sentences containing compound clauses and complex$_{\overline{RF}}$ NPs, the prevalence of complex$_{\overline{RF}}$ NPs in texts of the news register suggests that the method will be capable of simplifying and changing the accessibility of these texts to a greater extent than those of the health or literary registers. In comparison with the training data (Table 4.5), validation data of the health register contains a larger proportion of restrictive relative clauses (22.58% vs. 17.86%) and a smaller proportion of non-restrictive nominally bound relative clauses modifying complex$_{\overline{RF}}$ NPs (11.69% vs. 17.86%). Validation data of the literary register also contains a smaller number of non-restrictive nominally bound relative clauses (13.92% vs. 21.59%).

In addition to the tagging method based on CRF, I implemented a baseline tagger which exploits memory-based learning and was optimised in the same way as the MBL baseline for tagging compound clauses. For the tagging of complex constituents, optimal baseline performance was achieved using the IB1 algorithm, the weighted overlap feature metric, gain ratio feature weighting, with equal weight given to all neighbours regardless of their distance, and when making classifications on the basis of eleven nearest neighbours.

Table 4.18 presents the accuracy of the methods based on CRF tagging (CRF) and memory-based learning (MBL) when tagging the spans of complex constituents. In this table, scores are averages obtained for all tokens in sequences containing complex constituents: those occurring before the complex constituent,

those occurring within the complex constituent but before the left boundary of the subordinate clause, those that are the left boundary of the subordinate clause, those within the complex constituent but occurring after the left boundary of the subordinate clause, and those occurring in the same sentence but after the complex constituent.

Table 4.18: *Evaluation results ($F_1$-score) for the tagging of complex constituents in texts of the three registers*

| | **Health** | | **Literature** | | **News** | |
|---|---|---|---|---|---|---|
| **Type** | **CRF** | **MBL** | **CRF** | **MBL** | **CRF** | **MBL** |
| Complex$_{\overline{RF}}$ NP | 0.4268 | 0.2339 | 0.4469 | 0.2622 | 0.6979 | 0.5198 |
| Adjectival | 0 | 0 | 0 | 0 | 0 | 0 |
| Adverbial | 0.4572 | 0.5646 | 0.5730 | 0.5330 | 0.7042 | 0.6249 |
| Cleft | 0.1723 | 0.0081 | 0 | 0.0086 | 0.0520 | 0 |
| Independent | 0 | 0 | 0.3565 | 0 | 0 | 0.0047 |
| Intensifying | 0 | 0 | 0.0756 | 0.0338 | 0 | 0 |
| Reporting | 0($NA$) | 0 | 0.1502 | 0.0488 | 0.7297 | 0.3251 |
| Restrictive | 0.4092 | 0.2715 | 0.2770 | 0.0940 | 0.1603 | 0.0674 |
| Clause Comp. Verb | 0.6044 | 0.7218 | 0.5324 | 0.3560 | 0.7339 | 0.6341 |
| *Wh*-phrase | 0.2959 | 0.0574 | 0.3926 | 0.0924 | 0.4381 | 0.0762 |
| Non-nominal | 0 | 0 | 0 | 0.1588 | 0 (NA) | 0 |
| Before complex constituents | 0.6012 | 0.4129 | 0.4736 | 0.3211 | 0.6412 | 0.4739 |
| In complex constituents | 0.5592 | 0.4245 | 0.4268 | 0.2892 | 0.6910 | 0.5722 |
| After complex constituents | 0.3802 | 0.2069 | 0.3459 | 0.2527 | 0.6336 | 0.4443 |
| Unknown | 0.0659 | 0.0554 | 0 | 0 | 0 | 0 |
| Errors (No complex constituent in sequence | 0.6269 | 0.5166 | 0.3195 | 0.2019 | 0.2343 | 0.0774 |
| $F_1$ (Micro-averaged) | 0.5442 | 0.4156 | 0.3994 | 0.2767 | 0.6525 | 0.5016 |
| Valid tag sequences (%) | 86.67 | 67.05 | 75.65 | 58.97 | 82.98 | 65.58 |

Inspection of Table 4.18 reveals that, when tagging sentences in texts of all

registers, the method based on CRF tagging is more accurate than the one exploiting memory-based learning. The results of paired-samples Student's t-tests indicate these differences in performance to be statistically significant ($p \ll 0.01$ in all cases). As in the evaluation of the tagging of compound clauses, row *Valid tag sequences* in Table 4.18 indicates that the tag sequences generated by the CRF method are more appropriate for use in the sentence simplification task than those generated using the $MBL$ baseline.[11]

Table 4.18 displays the accuracy of the tagger when applied to texts of the three registers. It is immediately apparent that the tagging of complex constituents, including complex$_{\overline{RF}}$ NPs, is less accurate than the tagging of compound clauses (Table 4.16). The method is much less accurate when tagging complex$_{\overline{RF}}$ NPs in texts of the registers of health ($F_1 = 0.4268$) and literature ($F_1 = 0.4469$) than it is in texts of the register of news ($F_1 = 0.6979$). For this reason, when simplifying sentences in the registers of health and literature, sentence analysis based on this automatic tagging of complex$_{\overline{RF}}$ NPs is unlikely to be useful. The relatively high accuracy of the method when detecting the spans of complex constituents in text of the news register provides some evidence to support integration of the tagging model into the method for simplification of Type 2 sentences in news texts. Inspection of Table 4.18 reveals that the tagging model detects the arguments of clause complement verbs with moderate success ($0.53 \leq F_1 \leq 0.73$) in texts of all registers. Detection of these constituents in

[11] In this context, the valid tag sequences are the same as those listed in Section 4.4.1, page 104.

input sentences can improve the performance of a sentence simplification method by helping to identify those constituents with obligatory subordinate clauses that cannot be simplified by splitting the subordinate clause and the matrix constituent over two shorter sentences.

## 4.5 Contribution to Research Question RQ-4

My development of tools to automatically tag the spans of compound clauses and complex constituents and my evaluation of these tools makes a partial contribution to research question **RQ-4**:

> *How does the accuracy of automatic sentence simplification compare when using a machine learning approach to detect the spans of compound clauses and complex$_{\overline{RF}}$ NPs and when using a method based on handcrafted patterns?*

The availability of accurate tools to tag the spans of compound clauses and complex constituents, in combination with the availability of an accurate sign tagger (Chapter 3) would make it trivial to detect the essential elements of the rule activation patterns exploited by my approach to sentence simplification: the conjoins of compound clauses and both the non-restrictive relative clause modifiers and the matrix constituents of complex$_{\overline{RF}}$ NPs. The methods presented in this chapter are not perfectly accurate, but despite their inaccuracy, it will be interesting to investigate the contribution that they may make in my approach to sentence simplification (Chapter 5). The relative success of the model to tag

the spans of compound clauses provides cause for optimism about the accuracy of a dependent sentence simplification method. This chapter provided the first in a three-part response to **RQ-4**.

# Chapter 5

# Automatic Sentence Simplification

This chapter describes a novel pipeline for automatic sentence simplification. The research described here comprises the first part of my response to research questions **RQ-3** and **RQ-5** and the second part of my response to **RQ-4**, supporting the main goal of the thesis.

Research question **RQ-3** seeks to establish the extent to which an iterative rule-based approach to sentence simplification can convert input sentences into a form containing fewer compound clauses and fewer complex$_{\overline{RF}}$ NPs. **RQ-3** focuses on an approach exploiting information about the signs of syntactic complexity occurring in input sentences and utilising handcrafted rules to simplify sentences. In this chapter, Section 5.2 details my sentence simplification algorithm and the transformation schemes that it applies. Evaluation of this sentence simplification method is presented in Chapter 6.

Research question **RQ-4** seeks to compare the accuracy of an approach to automatic sentence simplification exploiting handcrafted rule activation patterns with that of one exploiting machine-learned rule activation patterns. In this chapter, I present my overall approach to automatic sentence simplification, which forms the basis for two variant systems: one exploiting handcrafted rule activa-

tion patterns and one exploiting machine-learned rule activation patterns. The chapter includes details of the sentence transformation schemes used by the two systems and the handcrafted rule activation patterns exploited by the first system. This description forms the second part of my response to **RQ-4**. The intrinsic evaluation of the two systems, described in Sections 6.1 and 6.2, completes this response.

Research question **RQ-5** is concerned with determining whether my approach to sentence simplification facilitates subsequent text processing using NLP applications. The description in the current chapter of my overall approach to sentence simplification forms an initial response to **RQ-5**. The extrinsic evaluation of my approach, presented in Chapter 7, completes this response.

## 5.1 Previous Work in Sentence Simplification

In this section, I provide a survey of work related to the task of sentence simplification, with an emphasis on those methods which exploit information about the syntactic structure of input sentences for the transformation process. For brevity, as it is not the focus of my method, I do not cover previous work related to the task of lexical simplification here.

Automatic sentence simplification is one aspect of text simplification,[1] a topic that has been addressed in several lines of research since the 1990s. Numerous rule-based methods for sentence simplification have been developed (e.g. Chan-

[1]More comprehensive overviews of previous work in the general topic of text simplification are provided by Saggion (2017, 2018), and Siddharthan (2014).

drasekar *et al.*, 1996; Siddharthan, 2006; De Belder and Moens, 2010) and used to facilitate NLP tasks such as information extraction (Evans, 2011; Niklaus *et al.*, 2016) and semantic role labelling (Vickrey and Koller, 2008).

Previous work has addressed the task by exploitation of a range of language resources and NLP tools, including shallow preprocessing (e.g. Siddharthan, 2006) and syntactic parsing tools (e.g. Canning, 2002; Vickrey and Koller, 2008; Bott *et al.*, 2012a), sentence-aligned parallel corpora of texts in their original form and in a manually simplified form (e.g. Coster and Kauchak, 2011; Wubben *et al.*, 2012; Štajner, 2015), and syntactically-annotated versions of such corpora (e.g. Zhu *et al.*, 2010; Feblowitz and Kauchak, 2013; Siddharthan, 2014). In this section I present an overview of the most relevant research in sentence simplification.

### 5.1.1 Rule-Based Approaches

In many of the approaches exploiting shallow preprocessing, rules are triggered by pattern-matching applied to the output of text analysis tools such as partial parsers and part-of-speech (PoS) taggers. Siddharthan (2006) describes a method in which input text is analysed using a tokeniser, chunker, and PoS tagger. In this approach, handcrafted patterns are used to identify the grammatical roles of NPs, to resolve pronominal anaphora, and to "split" complex sentences containing relative clauses and compound constituents, including clausal and subclausal constituents. The handcrafted patterns are expressed in terms of prefix conjunctions (e.g. *though*, *when*, *if*) and infix conjunctions (e.g. *and*, *but*, *because*), and commas. The method is based on an iterative simplification method exploit-

ing rules which include operations for sentence ordering, for ensuring anaphoric cohesion, for preserving rhetorical relations, and for generating appropriate determiners when splitting sentences that contain relative clauses. In some respects, Siddharthan's (2006) method is similar to the one I propose in this thesis. However, the transformation rules used in his system are based on shallow information such as part of speech and chunk patterns. The rules used by my method for sentence simplification also exploit information about the coordinating and bounding functions of various lexical and punctuational markers of syntactic complexity. My approach integrates a ML-based classifier of these markers (Section 3.2) to provide a more detailed analysis of input sentences. The variant of my approach based on machine-learned rule activation patterns includes an additional layer of sentence analysis (Section 4.3).

Evans's (2011) approach, discussed in Chapter 1, is another example of a rule-based sentence simplification method.

#### 5.1.1.1 Methods Exploiting Syntactic Parsing

A large number of sentence simplification methods proposed in the past exploit automatic sentence analysis using syntactic parsers. These include techniques based on handcrafted transformation rules operating over the derived syntactic structure of input sentences and extraction of the syntactic relations or dependencies between words and the syntactic roles of constituents identified in those sentences. In many cases, the syntactic transformation rules employed in these methods are implemented using synchronous grammars (Shieber and Schabes,

1990), which specify transformation operations between syntactic trees, rather than surface-based text editing operations. They are typically used to simplify input sentences by re-ordering constituents or splitting sentences that contain compounds or complex constituents into simple sentences containing independent clauses (Angrosh and Siddharthan, 2014; Ferrés *et al.*, 2015; Mishra *et al.*, 2014; Rennes and Jönsson, 2015).

In previous work, several applications have been developed with the aim of improving text accessibility for human readers. Max (2000) described the use of a syntactic parser for sentence simplification to facilitate the reading comprehension of people with aphasia. In the PSET project, Canning (2002) implemented a system which exploits a parser in order to rewrite compound sentences as sequences of simple sentences and to convert passive sentences into active ones. Scarton *et al.* (2017) developed a multilingual syntactic simplification tool (MUSST) in the SIMPATICO project, which sought to improve the experience of citizens and companies in their daily interactions with public administration. The English sentence simplification tool includes components for sentence analysis, exploiting the Stanford dependency parser (de Marneffe *et al.*, 2006), to determine whether or not input sentences should be transformed, and to identify discourse markers and relative pronouns, which will be useful in the simplification of conjoint (compound) clauses and relative clauses. MUSST's syntactic simplification process implements the handcrafted rules proposed by Siddharthan (2004) and Siddharthan (2014) and applies them to the syntactic analyses generated for input sentences by the dependency parser. These include rules to split sentences con-

taining conjoint clauses, relative clauses, and appositive phrases and to convert passive sentences to active. After applying the simplification rules, MUSST also performs a generation step in which truecasing is applied to output sentences and discourse markers lost in the simplification process are re-inserted. When processing sentences in Public Administration texts, Scarton *et al.* (2017) report an accuracy of 76% for their system when simplifying English sentences, and taking into account a wider range of operations that those that are the focus of my thesis. In Sections 6.1 and 6.2, I compare the accuracy of MUSST with that of my approach, focusing only on the tasks of simplifying sentences containing compound clauses (Type 1) and complex$_{\overline{RF}}$ NPs (Type 2) in texts of several different registers.

Although I focus on simplification of English sentences in this thesis, rule-based methods have also been proposed for the processing of other languages (e.g. Dutch (Daelemans *et al.*, 2004), French (Brouwers *et al.*, 2014), and German (Suter *et al.*, 2016)). Several researchers have also developed methods to facilitate the process of acquiring sentence simplification rules from manually simplified corpora of languages such as Brazilian Portuguese (Aluisio *et al.*, 2008a,b) and Basque (Gonzalez-Dios *et al.*, 2018). Seretan (2012) presented a method to semi-automatically derive syntactic simplification rules for French sentences. Her method is based on a component to automatically identify sentences that require simplification and on manual analysis of the syntactic structures of complex and simple French sentences. The outputs of these two processes are then used to formulate rules to transform sentences with complex syntactic structures into

sentences with simpler syntactic structures.

In general, the weakness of simplification methods exploiting syntactic parsing is that they rely on high levels of accuracy and granularity of automatic syntactic analysis. Previous research has demonstrated that the accuracy of parsers is inversely proportional to the length and complexity of the sentences being analysed (Tomita, 1985; McDonald and Nivre, 2011). Rather than exploiting full syntactic parsing, the methods for sentence simplification that I present in this thesis exploit shallow and robust syntactic analysis steps (Sections 3.2 and 4.3).

### 5.1.2 Data-Driven Approaches

More recently, the availability of resources such as Simple Wikipedia has enabled text simplification to be included in the paradigm of statistical machine translation (Yatskar *et al.*, 2010; Coster and Kauchak, 2011). In this context, translation models are learned by aligning sentences in English Wikipedia (EW) with their corresponding versions in Simple English Wikipedia (SEW). Manifesting Basic English (Ogden, 1932), the extent to which SEW is accessible to people with reading difficulties has not yet been fully assessed. Effective SMT relies on the availability of representative pairs of texts in their original and converted forms. As a result, there are currently only a limited number of contexts in which SMT approaches are likely to be effective. Xu *et al.* (2015) are critical of the use of SEW to support SMT-based text simplification.

#### 5.1.2.1 Methods Exploiting Parallel Corpora

Despite these caveats, the availability of sentence-aligned parallel corpora of texts in their original and simplified forms provides additional opportunities to develop methods for sentence simplification. From a sentence-aligned collection of articles from EW and SEW, Coster and Kauchak (2011) derived the probabilities that various ngrams from EW occur in an edited form in SEW as a result of various transformation operations including phrase rewording, deleting, reordering, and splitting. They exploited the resultant phrase table in the development of a phrase-based statistical machine translation (PB-SMT) model to translate texts into a simplified form.

Wubben *et al.* (2012) applied this basic approach and also integrated a re-ranking metric to ensure that sentences generated by the model are sufficiently unlike the originals to constitute suitable transformations. This approach captures the intuition that generated sentences should be as fluent and informative as the originals, but sufficiently different from them. The models learned perform lexical substitution, phrase deletion, and phrase re-ordering operations.

Štajner *et al.* (2015) exploited a sentence-aligned parallel corpus of Spanish containing texts in their original versions and two simplified versions of decreasing complexity (manifesting "light" and "heavy" simplification). They applied methods from SMT to learn the simplification model and developed a language model for use by it from a set of sentences with a length of fifteen words or less. This was done to promote the simplicity of sentences generated by the system.

Zhang and Lapata (2017) applied methods from neural machine translation to develop DRESS, the deep reinforcement learning sentence simplification system. Trained on parallel corpora of unmodified and manually simplified English text, their method uses recurrent neural networks to implement an encoder-decoder architecture network to transform input word sequences into a simplified form. The system was trained in a reinforcement learning framework to ensure that the generated output satisfies constraints on simplicity, fluency, and meaning. The types of transformation operation learned by the model which affect sentence structure include those performed by other systems described in this section: addition, copying, deletion, and re-ordering of words and phrases.

Shardlow and Nawaz (2019) developed a text simplification system to facilitate communications from healthcare specialists to their patients. Their method is based on a neural text simplification method trained on texts from EW and SEW. They extended the model by adding a phrase table to provide simplifications of specialist medical terminology mined from a medical ontology (SNOMED-CT). Shardlow and Nawaz (2019) evaluated their method via text readability metrics, manual error analysis, and human evaluation based on crowd-sourced ranking of well-simplified sentences generated by four simplification methods. Of the four systems, on average, humans ranked sentences produced by theirs as being easiest to understand.

My approach to sentence simplification does not depend on the availability of parallel corpora of text in its original form and in a manually simplified form. It

does not apply text editing operations of the type used in phrase-based machine translation or neural machine translation. My approach is iterative and rule-based rather than exploiting empirically derived phrase tables.

#### 5.1.2.2 Methods Exploiting Syntactically Parsed Parallel Corpora

Several methods for sentence simplification have exploited syntactically annotated sentence-aligned parallel corpora of texts in their original and simplified forms. Zhu *et al.* (2010) developed an approach to sentence simplification using PB-SMT. Data from a sentence-aligned parallel corpus of EW/SEW articles was syntactically parsed. This syntactic information was exploited when computing the probabilities of transformation operations applied to sentences in EW generating the aligned sentences in SEW. The approach was able to derive these probabilities from information about syntactic structure such as constituent size and information about the occurrence in the constituent of relative pronouns. A PB-SMT approach was then used to learn syntactic transformation operations from this data. The types of transformations learned included phrase splitting, dropping and reordering, and substitution operations.

Feblowitz and Kauchak (2013) presented a method in which syntactic transformation rules are learned from a syntactically annotated parallel corpus of texts in their original and simplified forms. The rules were encoded in a synchronous tree substitution grammar (STSG) formalism, which models syntactic simplification. The authors improved the simplification model by incorporating additional syntactic information to better discriminate between input structures for which

transformations should be applied and those for which they should not.

Paetzold and Specia (2013) developed a system exploiting the *Tree Transducer Toolkit* to learn syntactic transformation rules from a syntactically parsed parallel corpus of texts in their original and simplified forms. The acquired rules are applied to input sentences parsed using the Stanford constituent parser.[2] Transformations learned in this approach include lexical and syntactic simplifications. The authors developed a set of heuristic filters to prevent the system from learning spurious rules. These filters ensure that, in order to be incorporated in the model, a candidate rule must either be general enough, must split one sentence into multiple sentences, must delete information, or must apply to structures which contain connectors such as *and* and *or*.

Angrosh *et al.* (2014) developed an approach incorporating one method for syntactic and lexical simplification and a second method for sentence compression. Lexicalised sentence transformation rules were learned from a syntactically parsed parallel corpus. These rules included both lexical and syntactic transformations. The sentence compression method employed techniques from integer linear programming and dynamic programming to select the best from among a large set of candidate node deletions to be applied to the syntactically analysed input sentences.

Siddharthan and Angrosh (2011) present a method exploiting techniques from PB-SMT to learn a synchronous grammar applying transformations to parse

[2]Implemented by John Bauer and available from https://nlp.stanford.edu/software/srparser.html. Last accessed 5th December 2019.

trees for text simplification. In their method, the synchronous grammar is semi-automatically acquired from a syntactically parsed sentence-aligned parallel corpus of articles from EW and SEW. These transformations include lexicalised rules for insertion, deletion, and re-ordering of syntactic constituents together with morphological transformation of verb forms to enable conversion of sentences from passive voice to active. A substantial part of the grammar consists of handcrafted rules to enable transformations that are more difficult to learn, such as the conversion of passive sentences to active, the splitting of complex sentences and compounds, and the standardisation of quotations. The authors applied a simple text generation component to ensure that sentences produced by the system are ordered in a way that matches that of the original text.

Narayan and Gardent (2014) present a method for sentence simplification in which a phrase-based SMT model learned from a parallel corpus of sentence-aligned EW and SEW articles is improved through the integration of deep semantic information. This is derived from *Boxer* (Bos, 2008), a tool which provides information on the discourse representation structure (Kamp and Reyle, 1993) of input sentences. Semantic information from this parser is used to improve the splitting of complex sentences by ensuring preservation of multiword units (entities and concepts) in the generated sentences and by avoiding the deletion of the obligatory arguments of verbs.

The field of text summarisation also includes approaches that exploit sentence simplification. For example, Cohn and Lapata (2009) present a syntactic tree-to-tree transduction method to filter non-essential information from syntac-

tically parsed sentences. This compression process often reduces the syntactic complexity of those sentences. An advantage of their method over the one that I present in this thesis is that it can identify elements for deletion in the absence of explicit signs of syntactic complexity. However, as with all methods exploiting full syntactic parsing, the approach is computationally expensive, with relatively long run times. One recent approach to sentence compression was presented by Klerke *et al.* (2015). Their method exploits LSTM learning in a joint-learning task to integrate information from combinatorial categorial grammar (Steedman, 1987) supertagging and eye tracking data for sentence simplification. The model is used to compress sentences by identifying non-essential words and phrases for deletion. The methods proposed by Cohn and Lapata (2009) and Klerke *et al.* (2015) both work by applying deletion operations. As with all such methods, they run the risk of omitting relevant information from their output.

In addition to the exploitation of handcrafted rules, several systems based on a syntactic analysis of input texts include a post-processing module to improve the quality of the sentences that they generate. Bott *et al.* (2012a) integrated a probabilistic component into their system to assess the suitability of applying transformation operations to input sentences. This approach to syntactic simplification was integrated into the *Simplex* text simplification system designed to convert texts into a more accessible form for people with Down's syndrome (Saggion *et al.*, 2015). Vickrey and Koller (2008) included a machine learning method in their sentence simplification tool to decide on the set of transformations to apply when processing input sentences. In addition to a syntactic dependency

analysis, Siddharthan (2011) integrated a generator in his system to sequence the application of handcrafted transformation rules and to ensure agreement between constituents in automatically generated sentences. Brouwers *et al.* (2014) developed an approach in which all possible transformations in their grammar are applied to input sentences and a method using integer linear programming and four assessment criteria is used to select the best of these.

The methods described in this section rely on two resources that are not exploited by the approach to sentence simplification that I present in this thesis: syntactic parsers and large syntactically analysed corpora. The former are ill-suited to the sentence simplification task, as noted in Sections 1.1 (page 15) and 5.1.1.1 (page 121), while the latter are relatively scarce resources that are not available for texts of all registers. These methods involve the learning of sentence transformation rules from syntactically parsed parallel corpora. The rules used by my sentence simplification system are not derived in this way. They are based on a shallow and robust analysis step in which explicit signs of syntactic complexity are classified with respect to their specific syntactic coordinating and subordinating functions. In this context, implicit syntactic structure is not tagged directly, but is inferred from the classification of explicit signs.

#### 5.1.2.3 Methods Exploiting Deep Parsing and Semantic Analysis

Several methods for sentence simplification exploit deep parsing and automatic methods for semantic analysis. Jonnalagadda *et al.* (2009) presented a method for syntactic simplification which includes a preprocessing step in which redun-

dant phrases are deleted,[3] the names of certain entities (genes) are normalised, and noun phrases are replaced. After preprocessing, input sentences containing multiple clauses are split into independent clauses using information about linkage relations identified by the Link Grammar parser[4] and about the distribution of commas in the sentences.

Miwa *et al.* (2010) applied a method based on deep parsing to preprocess sentences, removing unnecessary information to expedite a relation extraction task. They developed handcrafted rules to identify entity mentions, relative clauses, and copulas in sentences and to exploit the syntactic analysis to delete those parts of the sentence not mentioning entities. Transformation operations performed in this approach include the replacement of compound phrases by the final conjoin in the phrase that refers to an entity of interest and the deletion of matrix NPs whose appositions refer to an entity of interest.

Sheremetyeva (2014) presents an approach to sentence simplification in patent claims. Her method exploits a variety of advanced preprocessing steps including supertagging to identify semantic information about the words, terminology, and predicates in the text. Phrases are identified using a chunker based on phrase structure grammar rules and relations between predicates and their arguments are identified using an approach based on domain permutation graphs. These tools are used to identify the full set of predicate-argument dependencies in input sentences and to generate new simple sentences on the basis of each of them.

[3] Jonnalagadda *et al.* (2009) refer to these as "spurious phrases".

[4] Available at `https://www.abisource.com/projects/link-grammar`. Last accessed 7th January 2020.

My approach, which is used to process texts of multiple registers and domains, performs no deep semantic analysis of this kind. The approach presented by Sheremetyeva (2014), which exploits specific linguistic knowledge about patent claims and their structure and involves extraction of predicate-argument dependencies and sentence generation, may be difficult to adapt for simplification of texts that are not patent claims.

Sections 5.1.2.2 and 5.1.2.3 have included descriptions of several sentence compression methods. One disadvantage of such methods is that they are "destructive" in the sense that information is deleted rather than preserved as a result of compression. Although some information loss is inevitable in text simplification, my method is designed to minimise it. Currently, information conveyed by conjunctions and other signs of syntactic complexity is lost in my approach but information conveyed by other function words and content words is preserved.

To conclude this review of related work, I noted that many of the previous rule-based approaches to sentence simplification are based on a relatively coarse analysis of syntactic complexity and are often designed for use in a specific application area, such as domain-specific information extraction. I sought to develop an approach incorporating an analysis step detailed enough to support simplification of a variety of syntactically complex structures. Approaches based on the full syntactic analysis of input sentences have the potential to perform a larger variety of more precise transformation operations but they may be time consuming to run and unreliable when processing the types of sentence most in

need of simplification. Methods for sentence simplification based on statistical machine translation are efficient to run but depend on the availability of large collections of sentence-aligned parallel corpora for their training. This type of data is expensive to produce, especially in the case of systems designed to exploit syntactically parsed data. In general, methods for simplification based on sentence compression are unsuitable for my purpose because I seek to improve the accessibility of information in input sentences rather than deleting it. For these reasons, I developed methods for sentence simplification of English (Section 5.2) which are designed to be meaning-preserving and which integrate new components for syntactic analysis that are based not on syntactic parsing but on the automatic classification of various explicit textual signs of syntactic complexity (Chapter 3) and identification of the spans of compound clauses and complex$_{\overline{RF}}$ NPs (Chapter 4). Development of these taggers is based on the analyses of syntactic complexity in English sentences described in Chapter 2 and in Section 4.2.

## 5.2 Sentence Transformation

The overall approach to sentence simplification that I present in this thesis combines data-driven and rule-based methods. In the first stage, input sentences are tokenised and part-of-speech tagged using the TTT2 language processing package (Grover *et al.*, 2000).[5] After this, signs of syntactic complexity are identified and

[5]The experiments described in this thesis relied on TTT2 but the current version uses the implementation of the Brill tagger (Brill, 1994) distributed with GATE and used in the ANNIE application (Hepple, 2000). PoS tagging errors were not observed to have a great influence on the accuracy of the sentence simplification method which exploits handcrafted rule activation patterns. However, as implied by results presented in Table 6.7 of Section 6.1.3, this statement may not hold for the method based on machine-learned rule activation patterns which simplifies

classified using the machine learning method described in Chapter 3. One of the strengths of my approach is that it only requires these two shallow and reliable preprocessing steps.

My sentence simplification method is designed to process Type 1 sentences and Type 2 sentences. In this chapter, I present two variant systems implementing my approach. The two systems are based on a generic set of four sentence transformation schemes which are applied iteratively to simplify input sentences. The first system, OB1,[6] uses handcrafted rule activation patterns and a set of associated rules to implement the four transformation schemes. The second system, STARS,[7] uses the sequence tagging approach described in Chapter 4 to identify the spans of compound clauses and complex$_{\overline{RF}}$ NPs to derive more flexible machine-learned rule activation patterns. A small set of rules associated with these patterns is used in STARS to implement the four sentence transformation schemes.

### 5.2.1 The Algorithm

I observed in the annotated corpus presented in Chapter 2 that there is no upper limit on the number of signs of syntactic complexity that may occur in a sentence. For this reason, I developed an iterative approach to sentence simplification. A single transformation operation is applied in each iteration according to the class labels of the signs occurring in the sentence. Each application of a transformation

Type 1 sentences.

[6]Integrated within a system called *OpenBook*, which was developed in the EC-funded FIRST project to convert text into a more accessible form for autistic individuals (Orăsan *et al.*, 2018).

[7]Originally from a *Sequence Tagging Approach to Rewrite Sentences*.

operation converts an input sentence containing signs of syntactic complexity into two sentences, each containing fewer signs. These transformation operations apply exhaustively until the system is unable to detect any compound clauses or complex$_{\overline{RF}}$ NPs in the derived sentences.

**Input:** Sentence $s_0$, containing at least one sign of syntactic complexity of class $c$, where $c \in \{\text{CEV, SSEV}\}$.
**Output:** The set of sentences $A$ derived from $s_0$, that have reduced propositional density.

```
1  The empty stack W;
2  O ← ∅;
3  push(s_0, W);
4  while isNotEmpty(W) do
5  |   pop(s_i, W);
6  |   if s_i contains a sign of syntactic complexity of class c (specified in
   |    Input) then
7  |   |   s_i1, s_i2 ← transform_c(s_i);
8  |   |   push(s_i1, W);
9  |   |   push(s_i2, W);
10 |   else
11 |   |   O ← O ∪ {s_i}
12 |   end
13 end
```

**Algorithm 1:** Sentence simplification algorithm

My sentence simplification method implements Algorithm 1. Two iterative processes are used to transform the original sentence and each of the sentences generated in the working set. The first process applies rules to transform Type 1 sentences. It ends when no compound clauses can be detected in any of the sentences in the working set. The second process applies rules to transform Type 2 sentences. In a similar fashion, this process ends when no complex$_{\overline{RF}}$ NPs can

be detected in any of the sentences in the working set.[8]

Application of the sentence transformation schemes used in these processes (line 7 of Algorithm 1) is triggered by detection of tagged words and signs in the input sentence. Signs of class CEV indicate the occurrence of at least one compound clause in the input sentence. Signs of class SSEV following nouns in the sentence indicate the occurrence of at least one finite subordinate clause, potentially modifying a complex$_{\overline{RF}}$ NP. Detection of other types of signs has a role to play in the automatic simplification process as it can be used in my implementation of the sentence transformation schemes to identify clause boundaries.

The method presented in this thesis implements different transformation schemes for Type 1 sentences and for three subcategories of Type 2 sentences. Each scheme has an activation pattern consisting of various textual elements to be identified in an input sentence and an output sequence constructed from elements detected in the activation pattern. Elements in the activation patterns include signs of syntactic complexity, words with particular parts of speech, and word sequences. In the rules implementing the transformation schemes, sentence initial elements of the activation pattern that occur in non-sentence initial positions in the output sequence are downcased. Similarly, non-sentence initial elements of the activation pattern that occur in a sentence initial position in the output sequence are upcased. In the schemes presented in Sections 5.2.1.1 and 5.2.1.2, the sign of syntactic complexity of class C, triggering the transformation, is denoted **_C**.

[8]The sentences of input documents are processed one at a time, rather than all being enqueued in a single batch. The stack only holds the sentence being processed and its intermediate derivations.

Specific signs of class C are denoted $sign_{\mathbf{C}}$ (e.g. $and_{\mathbf{CEV}}$ or $that_{\mathbf{SSEV}}$). Word sequences are denoted as uppercase letters (A, B, etc). For word sequence A beginning with an indefinite article, $A_D$ is the same word sequence beginning with a definite article. Sequences of nominal words (e.g. determiners, adjectives, and nouns of various types) are denoted $A_n$. Words with part of speech P are denoted $w_P$. The spans of compound clauses and complex$_{\overline{RF}}$ NPs are marked using square brackets. In the example sentence transformations provided, elements within the compound clause or complex$_{\overline{RF}}$ NP are underlined and co-indexed with the corresponding elements in the transformation scheme. Elements outside of the compound clauses and complex$_{\overline{RF}}$ NPs are indicated using braces which are also co-indexed with the corresponding elements of the transformation scheme.

#### 5.2.1.1 Transformation Scheme to Simplify Type 1 Sentences

One transformation scheme is used to simplify Type 1 sentences. The rules in this scheme generate two new sentences that do not contain the sign of syntactic complexity which triggered its application. The transformation scheme used to simplify Type 1 sentences is

A [B $\_\mathbf{CEV}$ C] D. $\rightarrow$ A B D. A C D.

This would convert a sentence such as (41-a) to (41-b). In this example, the conjoins of the compound clause (B and C) are underlined while indexed braces are placed around elements A and D.

(41) a. {They were formally found not guilty by the recorder Michael Gibbon QC after}$_A$ [a witness, who cannot be identified, withdrew from giving evidence$_B$ and$_{\mathbf{CEV}}$ prosecutor Susan Ferrier offered no further evidence$_C$]{}$_D$.

b. (i) {They were formally found not guilty by the recorder Michael Gibbon QC after}$_A$ a witness, who cannot be identified, withdrew from giving evidence$_B$ {}$_D$.

(ii) {They were formally found not guilty by the recorder Michael Gibbon QC after}$_A$ prosecutor Susan Ferrier offered no further evidence$_C$ {}$_D$.

#### 5.2.1.2 Transformation Schemes to Simplify Type 2 Sentences

Three transformation schemes are used to simplify Type 2 sentences. The rules of these schemes are designed to generate two new sentences that do not contain the sign that triggered their application.

The first transformation scheme used to simplify Type 2 sentences is

A [B$_n$ $w_{IN}$ *which*$_{\mathbf{SSEV}}$ C] D. $\rightarrow$ A B D. C $w_{IN}$ B$_D$.

This scheme is used to simplify sentences containing a complex$_{\overline{R}F}$ NP with an object relativised clause, where the modified NP is the object of a preposition. The rule would convert a sentence such as (42-a) to (42-b).

(42) a. {Littlebury was dressed in}$_A$ [a dark overcoat$_B$ under$_{IN}$ which$_{\mathbf{SSEV}}$ he concealed a gun he had bought three days earlier for £300$_C$]{}$_D$.

b. (i) {Littlebury was dressed in}$_A$ a dark overcoat$_B$ {}$_D$.

(ii) He concealed a gun he had bought three days earlier for £300$_C$ under$_{IN}$ the dark overcoat$_B$.

The second transformation scheme used to simplify Type 2 sentences is

A [B$_n$ **_SSEV** C $w_v$ D] E. → A B E. C $w_v$ B D.

This scheme is used to simplify sentences containing a complex$_{\overline{RF}}$ NP with an object relativised clause, where the modified NP is the object of a verb. The rule would convert a sentence such as (43-a) to (43-b).

(43) a. {}$_A$ [The attention$_B$ that$_{\mathbf{SSEV}}$ the Beckhams have$_C$ brought$_v$ to Alderley Edge since they arrived two and a half years ago$_D$] {is clearly not welcomed by some villagers}$_E$.

b. (i) {}$_A$ The attention$_B$ {is clearly not welcomed by some villagers}$_E$.

(ii) The Beckhams have$_C$ brought$_v$ the attention$_B$ to Alderley Edge since they arrived two and a half years ago$_D$.

The third transformation scheme used to simplify Type 2 sentences is

A [B$_n$ **_SSEV** C] D → A B D. B C.

This rule template is used to simplify sentences containing a complex$_{\overline{RF}}$ NP with a subject relativised clause. The rule would convert a sentence such as (44-a) to (44-b).

(44) a. {The case against}$_A$ [the Grants$_B$, who$_{\mathbf{SSEV}}$ denied three charges of cruelty between 1994 to 1997$_C$], {collapsed when a witness withdrew evidence due to be given in court}$_D$.

b. (i) {The case against}$_A$ the Grants$_B$ {collapsed when a witness withdrew evidence due to be given in court}$_D$.

(ii) The Grants$_B$ denied three charges of cruelty between 1994 to 1997$_C$.

The transformation schemes are relatively simple but automatic detection of the different patterns is non-trivial. In Chapter 4, I presented a machine learning method to identify the spans of compound clauses and complex$_{\overline{RF}}$ NPs and the elements in the activation patterns in the transformation schemes. In Section 5.2.2, I present a second method exploiting handcrafted patterns to identify these elements.

### 5.2.2 Handcrafted Rule Activation Patterns

The activation patterns used in the sentence transformation schemes presented in Section 5.2.1 require detection of various word sequences to enable identification of the spans of compound clauses and complex$_{\overline{RF}}$ NPs. In this section, I present rules exploiting handcrafted patterns that are used to automatically identify these elements and these spans. The transformation scheme for simplification of Type 1 sentences was implemented by developing 28 rules, each of which was based on a different handcrafted pattern. The scheme for simplification of Type 2 sentences

was implemented by developing 125 rules of this kind.

The rule sets associated with each sign tag (SSEV and CEV) were developed incrementally by using the sentence simplification method to process the annotated corpus described in Section 2.2. The texts described earlier, annotated with information about signs of syntactic complexity, were used for this purpose but were not used for the evaluation. Each rule set was initialised as an empty set. When processing a sentence which contains at least one sign of the relevant class and which does not match any existing rule activation pattern, the sentence was printed and the program stopped. I then manually formulated a new pattern to match the compound clause (CEV) or complex$_{\overline{RF}}$ NP (SSEV) in the sentence together with an associated transformation operation and added the resulting rule to the relevant rule set. This process continued until I perceived that the addition of new rules to process previously unseen sentences introduced errors in the processing of sentences that had previously been processed successfully. When developing the rule sets, my focus was on the capacity of the activation patterns to correctly match the different elements of sentences containing compound clauses and complex$_{\overline{RF}}$ NPs. In some cases, especially for the simplification of Type 2 sentences, the implemented rules do not perfectly implement the transformation schemes (see rules SSEV-43 and SSEV-61 in Table 5.2). After inclusion in the rule set, transformation operations were edited manually on inspection of the resulting output sentences generated. During development, formal evaluation of the sentence transformation rules and the combined operation of the rule sets was not performed due to the absence at that time of gold standard evaluation

data. The handcrafted rule activation patterns were not evaluated in isolation because they are intrinsic to the transformation rules themselves. The accuracy of the rules is discussed in Section 6.1.3. The texts used for development of the rules were not included in the gold standards used to evaluate my system.

Tables 5.1 and 5.2 each display the three most frequently activated rules used to transform Type 1 and Type 2 sentences, respectively. In these examples, the handcrafted rule activation patterns are expressed in terms of elements defined in Table 5.3. The * operator is used to indicate non-greedy matching.[9] In these tables, square brackets denote sentence boundaries, not the boundaries of compound clauses and complex$_{\overline{RF}}$ NPs. Sentence simplification was facilitated by accurate identification of signs linking clauses (CEV), noun phrases (CMN1), and adjective phrases (CMA) in coordination and signs serving as the left or right boundaries of bound clauses, including finite relative (SSEV/ESEV), nominal/appositive (SSMN/ESMN), and adjective (SSMA/ESMA) clauses.

The transformation operations applied to Type 1 sentences generate pairs of sentences in which the sentence containing the first conjoin precedes the sentence containing the second. In the case of Type 2 sentences, the reduced sentence containing the matrix NP[10] precedes the sentence linking the matrix NP to the predication of the relative clause. The use of a stack data structure in Algorithm 1 means that the simplification occurs in a depth-first manner. In a sentence containing two clause conjoins, each of which contains one complex$_{\overline{RF}}$ NP, the

[9] Analogous to the *? operator in Perl regular expressions.

[10] This sentence is "reduced" because the transformation operations delete the nominally bound relative clause.

output is ordered so that the sentence containing the reduced first conjoin is followed first by the sentence linking the matrix NP of that conjoin to the predication of its bound relative clause, then by the sentence containing the reduced second conjoin, and finally by the sentence linking the matrix NP in the second conjoin to the predication of its bound relative clause. In this way, sentences containing formerly complex$_{\overline{RF}}$ NPs are immediately followed by the sentences that provide more information about those NPs.

Table 5.1: *Example rules used to transform Type 1 sentences ($transform_{CEV}(s_i)$)*

| **Rule** | **Pattern** | **Original sentence** | **Transformed sentence** |
|---|---|---|---|
| CEV-24 | [*A* _ *B*]<br>↓<br>[*A*.]<br>[*B*.] | Kattab of Eccles, Greater Manchester, was required to use diluted chloroform water in the remedy[, *but*] the pharmacy only kept concentrated chloroform, which is twenty times stronger. | Kattab, of Eccles, Greater Manchester, was required to use diluted chloroform water in the remedy. The pharmacy only kept concentrated chloroform, which is twenty times stronger. |
| CEV-12 | [*A that B* _ *C*]<br>↓<br>[*A that B*.]<br>[*A that C*.] | "He was trying to intimate that mum was poorly [*and*] we should have expected that she might die at any time." | "He was trying to intimate that mum was poorly." "He was trying to intimate that we should have expected that she might die at any time." |
| CEV-27 | [*A* $v_{EV}$ *B* "*C* _ *D*]<br>↓<br>[*A* $v_{EV}$ *B* "*C*.]<br>[*A* $v_{EV}$ *B* "*D*.] | He said to me, 'You're dodgy[,] you're bad news[,] you know you're bad news.' | He said to me, 'You're dodgy.' He said to me, 'you're bad news.' He said to me, 'you know you're bad news.' |

Although the patterns used in the rule sets only explicitly refer to a small

Table 5.2: *Example rules used to transform Type 2 sentences* ($transform_{SSEV}(s_i)$)

| Rule | Pattern | Original sentence | Transformed sentence |
|---|---|---|---|
| SSEV-1 | [*A* $w_n$* _ *B* $s_{ESEV}$ *C*] ↓ [*A* $w_n$ *C*.] [$w_n$ *B*.] | Drummond[, *who*] had pleaded not guilty, was jailed for three months concurrently on each of six charges of wilfully killing, taking and mistreating badgers. | Drummond was jailed for three months concurrently on each of six charges of wilfully killing, taking and mistreating badgers. Drummond had pleaded not guilty. |
| SSEV-43 | [A *a/an* $w_n$* $w_n$ _ $w_{NNP}$ $w_{VBD}$ *C*] ↓ [*A a/an* $w_n$* $w_n$.] [*It was the* $w_n$* $w_n$ $w_{NNP}$ $w_{VBD}$ *C*] | In February last year police raided a council house [*which*] Francis rented in St Ann's. | In February last year police raided a council house. It was the council house Francis rented in St Ann's. |
| SSEV-61 | [*A* $w_{IN}$ $w_{DT}$ $w_n$* _ $w_V$ *B*] ↓ [*A* $w_{IN}$ $w_{DT}$ $w_n$*.] [*That* $w_n$* $w_V$ *B*] | One's heart goes out to the parents of the boy [*who*] died so tragically and so young. | One's heart goes out to the parents of the boy. That boy died so tragically and so young. |

Table 5.3: *Elements used in sentence transformation patterns*

| Element | Denotation |
|---|---|
| $\overline{\text{c}}$ | The detected sign of class *c* |
| Upper case letters (A-D) | Sequences of zero or more characters matched in a non-greedy fashion |
| $w_{POST}$ | Word of PoS POST, from the Penn Treebank tagset (Marcus *et al.*, 1993) |
| $w_n$ | Nominal word |
| $w_v$ | Verbal word, including -*ed* verbs tagged as adjectives |
| $s_{TAG}$ | Sign of syntactic complexity with tag TAG |
| $v_{EV}$ | Clause complement verb (e.g. *accept*, *deny*, *mean*, *retort*, *said*, etc.) |
| *word* | Word *word* |

number of sign tags, it is necessary to discriminate between them accurately. For example, when simplifying a sentence such as (45),

(45) Helen[$_{SSEV}$ , who] has attended the Carol Godby Theatre Workshop in Bury[$_{SSMN}$ ,] Greater Manchester[$_{ESMN}$ ,] since she was five[$_{ESEV}$ ,] has also appeared in several television commercials.

it is necessary to discriminate between the two final commas to accurately identify the span of the complex$_{\overline{RF}}$ NP.

I developed an automatic sentence simplification system called OB1 which integrates the handcrafted rule activation patterns presented in this section and the sign tagger presented in Chapter 3 into the sentence simplification algorithm described in Section 5.2.1. I present intrinsic and extrinsic evaluations of this approach in Chapters 6 and 7 of the thesis.

### 5.2.3 Machine-Learned Rule Activation Patterns

This implementation of my approach to sentence simplification uses the methods described in Chapter 4 to automatically identify the spans of compound clauses and complex$_{\overline{RF}}$ NPs in input sentences. The sentence simplification process based on machine-learned rule activation patterns works by first using the sign tagger (Chapter 3) to identify and classify the signs of syntactic complexity occurring in input sentences. After this step, Algorithm 1 (Section 5.2.1) is applied. At line 7 of the algorithm, the methods described in Chapter 4 to identify the spans of compound clauses and complex constituents and to classify complex constituents are applied to input sentence $s_i$. This automatic tagging of compound clauses and complex$_{\overline{RF}}$ NPs in the input sentence directly identifies the square bracketed

multi-token elements of the sentence transformation schemes presented in Sections 5.2.1.1 and 5.2.1.2. As a result, this approach to shallow syntactic analysis, which includes sign tagging and tagging of compound clauses and complex$_{\overline{RF}}$ NPs provides all of the information needed to implement the sentence transformation schemes. Function $transform_C$ in Algorithm 1 is implemented using four rules exploiting information from the taggers. I developed an automatic sentence simplification system called STARS which integrates this approach. I present a comparative evaluation of the systems exploiting handcrafted and machine-learned rule activation patterns in Chapter 6.

### 5.2.4 Suitability of the Sign Tagger for Use in Sentence Simplification

The transformation scheme implementing the $transform_{CEV}$ function in Algorithm 1 (page 133) depends on accurate classification of signs of syntactic complexity. Information about the classes of signs is required to detect the different elements of the handcrafted rule activation patterns in input sentences (presented in Section 5.2.2). The method for sentence transformation exploiting machine-learned activation patterns also exploits information about the tags applied to signs of syntactic complexity. Accurate sign tagging serves as an important first step in applying the associated transformation schemes.

Matching of the handcrafted rule activation pattern used in the transformation scheme for $transform_{CEV}$ depends on accurate detection of two classes of signs:

1. clause coordinators (CEV) and

2. left boundaries of subordinate clauses ({SSEV, SSCM, SSMA, SSMAdv, SSMI, SSMN, SSMP, and SSMV})

Thus, when simplifying Type 1 sentences, confusions between tags of the set specified in 2 are irrelevant (e.g. confusion between SSEV and SSCM). By contrast, confusion between tags of *different* sets specified in 1 and 2 is relevant (e.g. confusion between CEV and SSMA). Table 5.4 displays the accuracy with which the sign tagger assigns the two sets of class labels relevant to simplifying Type 1 sentences. There, row $SSX$ pertains to signs tagged with any of the class labels in the set listed in 2. Considered over the full set of signs, the tagger assigns these class labels with a micro-averaged $F_1$-score of 0.9318.

Table 5.4: *Evaluation of the sign tagger over tags exploited in the simplification of Type 1 sentences*

| **Tag** | **P** | **R** | $F_1$ | **Support** | **True-Pos** | **False-Pos** | **False-Neg** |
|---|---|---|---|---|---|---|---|
| CEV | 0.7991 | 0.7991 | 0.7991 | 876 | 700 | 176 | 176 |
| SSX | 0.9794 | 0.9251 | 0.9515 | 6076 | 5621 | 118 | 455 |
| **Micro average:** | | | | | | | |
| All | 0.9556 | 0.9092 | 0.9318 | 6952 | 6321 | 294 | 631 |

The transformation scheme implementing the $transform_{SSEV}$ function in Algorithm 1 depends on accurate detection of four classes of signs. This information is required to detect, in input sentences, different elements of the handcrafted rule activation patterns. Once these elements have been identified, the rules implementing the associated transformations are easy to apply. The four classes of signs are:

1. noun phrase coordinators (CMN1),

2. right boundaries of finite relative clauses (ESEV),

3. right boundaries of direct quotes (ESCM), and

4. left boundaries of subordinate clauses ({SSEV, SSCM, SSMA, SSMAdv, SSMI, SSMN, SSMP, or SSMV}).

Thus, for sentence simplification, confusions between tags in the set specified in 4 are irrelevant (e.g. confusion between SSEV and SSCM). By contrast, confusion between tags in the sets specified in 1-4 are relevant (e.g. confusion between SSMA and ESEV or between CMN1 and SSMP). Table 5.5 displays the accuracy with which the sign tagger assigns these four sets of class labels. There, row $SSX$ pertains to signs tagged with any of the class labels in the set specified in 4. The sign tagger assigns class labels belonging to these four sets to signs with a micro-averaged $F_1$-score of 0.8862.

Table 5.5: *Evaluation of the sign tagger over tags exploited in the simplification of Type 2 sentences*

| **Tag** | **P** | **R** | $F_1$ | **Support** | **True-Pos** | **False-Pos** | **False-Neg** |
|---|---|---|---|---|---|---|---|
| CMN1 | 0.7286 | 0.6628 | 0.6942 | 1041 | 690 | 257 | 351 |
| ESEV | 0.5261 | 0.4789 | 0.5014 | 1041 | 272 | 245 | 296 |
| ESCM | 0.9207 | 0.9379 | 0.9292 | 322 | 302 | 26 | 20 |
| SSX | 0.9794 | 0.9251 | 0.9515 | 6076 | 5621 | 118 | 455 |
| Micro average: | | | | | | | |
| All | 0.9142 | 0.8598 | 0.8862 | 8480 | 6885 | 646 | 1122 |

Over all the tags exploited in the two types of sentence simplification, the tagger assigns class labels with a micro-averaged $F_1$-score of 0.9075.

For the approach to sentence simplification based on machine-learned rule activation patterns (Section 5.2.3 and Chapter 4), implementation of these patterns depends on accurate identification of clause coordinators (signs of class CEV) and the left boundaries of finite subordinate clauses (signs of class SSEV). As already observed, the sign tagger is able to identify these signs accurately ($F_1 = 0.7791$ for signs of class CEV while $F_1 = 0.9467$ for signs of class SSEV). Information on the tags of other signs occurring in the same sentences as these items was also found to be a useful feature in the tagging models for compound clauses and complex$_{\overline{RF}}$ NPs.

Considered over all signs of syntactic complexity, with a micro-averaged $F_1 = 0.7991$, I am optimistic that the sign tagger will be useful for matching both the machine-learned and handcrafted rule activation patterns in English text and implementing the sentence transformation schemes specified in Section 5.2.1.

## 5.3 Contribution to Research Questions RQ-3, RQ-4, and RQ-5

The work described in this chapter makes partial contributions to three of the research questions set out in Chapter 1.

In response to research question **RQ-3**:

> *To what extent can an iterative rule-based approach exploiting automatic sign classification and handcrafted patterns convert sentences*

*into a form containing fewer compound clauses and fewer complex$_{\overline{RF}}$ NPs?*

the chapter presents a generic sentence simplification algorithm which exploits an automatic sign classifier (described in Chapter 3), two sets of sentence transformation schemes (Sections 5.2.1.1 and 5.2.1.2), and a set of rules exploiting handcrafted activation patterns which implement those schemes (Section 5.2.2). The intrinsic evaluation of this approach presented in Chapter 6 will make a direct contribution to **RQ-3**.

In response to research question **RQ-4**:

*How does the accuracy of automatic sentence simplification compare when using a machine learning approach to detect the spans of compound clauses and complex$_{\overline{RF}}$ NPs and when using a method based on handcrafted patterns?*

this chapter includes a description of the development of a generic sentence simplification algorithm (Section 5.2.1) and a set of sentence transformation schemes (Sections 5.2.1.1 and 5.2.1.2) which can be implemented using machine-learned rule activation patterns.[11] This comprises the second part of my three-part response to **RQ-4**.

In response to research question **RQ-5**:

*Does the automatic sentence simplification method facilitate subsequent text processing?*

[11] Derived using the methods presented in Chapter 4.

this chapter presents a generic sentence simplification algorithm, the sentence transformation schemes that it applies, and rule sets exploiting handcrafted activation patterns which implement those schemes. This comprises the first part of my response to **RQ-5**. The extrinsic evaluation of this approach presented in Chapter 7 will complete my response to **RQ-5**.

# Chapter 6

# Intrinsic Evaluation

The evaluations described in this chapter address research questions **RQ-3** and **RQ-4**. **RQ-3** is concerned with evaluating the accuracy of a sentence simplification system exploiting automatic sign classification and handcrafted rule-activation patterns, while **RQ-4** is concerned with comparing the accuracy of a sentence simplification system of this type with that of a system exploiting machine-learned rule activation patterns.

In this chapter, I present my evaluation of the sentence analysis and sentence transformation methods developed in my research. This includes evaluation by comparison of the output of the methods with human simplified text (Section 6.1), by reference to automatic estimations of the readability of their output (Section 6.2), and by reference to readers' opinions on the grammaticality, comprehensibility, and meaning of their output (Section 6.3). Section 6.1 includes evaluation of a sentence simplification method that uses handcrafted patterns (presented in Section 5.2.2) and a method using machine learning to identify the spans of compound clauses and complex$_{\overline{RF}}$ NPs (Section 5.2.3). Analysis of the two addresses **RQ-4**.

## 6.1 Comparison with Human-Produced Simplifications

I evaluated automatically simplified sentences generated by two variants of my approach to sentence simplification in terms of accuracy, assessed by reference to gold standards produced by linguists. For this purpose, tools and resources were also developed to support automatic evaluation of the systems that can be replicated easily to facilitate their development. In the experiments, system performance was also compared with that of two baseline methods.

### 6.1.1 Gold Standards

Two datasets were developed which constitute gold standards for the sentence simplification tasks against which system output could be compared. These were developed by a linguist who was a native speaker of English and was well-versed in English grammar. She was presented with output generated by the sentence simplification system when it was used to automatically simplify Type 1 sentences (1009 sentences of three registers – 325 of Health, 419 of Literature, and 265 of News) and when used to simplify Type 2 sentences (885 sentences of the three registers – 137 of Health, 379 of literature, and 369 of news). The linguist produced the gold standards by manually correcting automatically transformed sentences generated by the OB1 system exploiting handcrafted rule activation patterns. She was asked to undo transformations involving the arguments of clause complement verbs and transformations triggered by the misclassification of signs without coordinating or bounding functions. She was also asked to cor-

rect grammatical errors in the output sentences. The goal of the task she was undertaking and the way in which the algorithm worked were verbally explained to the linguist and the sentence simplification tool was demonstrated before the post-editing task began.

The sentence simplification methods were applied to texts of all three registers. Table 6.1 contains information about the subset of data used to test the sentence simplification method when simplifying sentences which contain compound clauses (Type 1). The column *Signs* contains two subcolumns: *All*, which displays the number of signs of syntactic complexity in the data, and *CEV*, which displays the number of signs tagged CEV by human annotators (*Oracle*) and by the automatic tagger described in Chapter 3 (*OB1*). *Compound Clauses* displays the number of compound clauses in the dataset.[1] It comprises one column (*Gold*) which displays the number of compound clauses identified by linguists in the dataset (the gold standard) and another (*OB1*) which displays the number identified by the sentence transformation method described in Section 5.2. *Derived Sentences* is the number of sentences generated as a result of simplifying Type 1 sentences. Subcolumn *Gold* displays the number of sentences generated by the linguists in the gold standard while subcolumn *OB1* displays the number generated by the automatic sentence simplification tool. In the evaluation, I filtered sentences that did not contain signs manually tagged as being of class CEV.

Table 6.2 contains information about the subset of data used to test the sen-

[1]These may contain two or more conjoins, each of which is a clause.

Table 6.1: *Characteristics of the test data used to evaluate the method to simplify Type 1 sentences*

| | | Signs | | | Compound Clauses | | Derived Sentences | |
|---|---|---|---|---|---|---|---|---|
| | | | CEV | | | | | |
| **Register** | **Tokens** | **All** | **Oracle** | **OB1** | **Gold** | **OB1** | **Gold** | **OB1** |
| Health | 7 198 | 885 | 375 | 265 | 364 | 229 | 698 | 470 |
| Literature | 15 067 | 2 181 | 442 | 511 | 425 | 291 | 1 154 | 686 |
| News | 7 270 | 898 | 311 | 294 | 293 | 276 | 607 | 564 |

Table 6.2: *Characteristics of the test data used to evaluate the method to simplify Type 2 sentences*

| | | Signs | | | Complex Sentences | | Derived Sentences | |
|---|---|---|---|---|---|---|---|---|
| | | | SSEV | | | | | |
| **Register** | **Tokens** | **All** | **Oracle** | **OB1** | **Gold** | **OB1** | **Gold** | **OB1** |
| Health | 3 481 | 501 | 214 | 229 | 176 | 125 | 260 | 129 |
| Literature | 13 280 | 1 967 | 430 | 525 | 404 | 206 | 482 | 260 |
| News | 25 850 | 2 534 | 531 | 619 | 401 | 372 | 598 | 501 |

tence simplification method when simplifying Type 2 sentences. In many cases, the meanings of the column headings are the same as those provided about Table 6.1. In Table 6.2, subcolumn *SSEV* of *Signs* displays the number of left boundaries of finite subordinate clauses in the dataset. *Complex Sentences* displays the number of sentences in the dataset that contain one or more of these boundaries. *Derived Sentences* is the number of sentences generated as a result of simplifying Type 2 sentences in this dataset. I filtered sentences that did not contain signs manually tagged as being of class SSEV.

### 6.1.2 Evaluation Using Overlap Metrics

I used an existing implementation of the SARI metric (Xu *et al.*, 2016)[2] to evaluate the sentence simplification systems described in this thesis. Xu *et al.* (2016) note that SARI "principally compares [s]ystem output [a]gainst [r]eferences and against the [i]nput sentence." It is based on a comparison of each sentence generated by a simplification system in response to a given input sentence with both the original form of the input sentence and with the set of sentences generated by human simplification of the input sentence. This metric is preferred to BLEU for the evaluation of sentence simplification systems because it is noted to correspond better with human judgements of simplification quality (Xu *et al.*, 2016). SARI provides a measure of the similarity between a single sentence and its simplification. I adapted the implementation to compute an average score over all simplified sentences output by the systems (for each type of sentence and each text register).

In addition to the SARI evaluation metric, I calculated the $F_1$-score of the method as the harmonic mean of precision and recall, given by Algorithm 2. In this algorithm,

$$sim = 1 - (\frac{ld(h, r)}{max(length(h), length(r))})$$

where $h$ and $r$ are sentences occurring in the gold standard and in the system response, respectively; $ld$ is the Levenshtein distance between $h$ and $r$ (Levenshtein,

[2]Available at https://github.com/cocoxu/simplification/blob/master/SARI.py. Last accessed 7th January 2020.

1966);[3] and $length(x)$ is the length of $x$ in characters. The intuition for use of Algorithm 2 is to find, in a greedy manner, the best matches between sentences produced by the system and sentences in the gold standard while still allowing some small differences between them.

**Input:** $H$ – set of simplified sentences in the gold standard for a given input sentence $S$
$R$ – set of simplified sentences produced by the system for input sentence $S$.
$H_0 \leftarrow H$.
$R_0 \leftarrow R$.
**Output:** $Precision$, $Recall$
$matched\_pairs = 0$
**while** $|H| \neq 0$ *and* $|R| \neq 0$ **do**
  $h, r \leftarrow \arg\max_{h \in H, r \in R} (sim(h, r))$
  **if** $sim(h, r) > 0.95$ **then**
    $H = H \backslash \{h\}$
    $R = R \backslash \{r\}$
    $matched\_pairs \mathrel{+}= 1$
  **else**
    *break*
  **end**
**end**
$Precision = \frac{matched\_pairs}{|H_0|}$
$Recall = \frac{matched\_pairs}{|R_0|}$

**Algorithm 2:** Evaluation algorithm for sentence simplification

Table 6.3 displays evaluation statistics for methods to simplify Type 1 sentences obtained using the SARI and $F_1$ metrics. These include the simplification methods presented in Chapter 5 of this thesis. The *Bsln* subcolumn displays the

[3] I used the Perl implementation of Levenshtein distance posted at https://www.perlmonks.org/?node_id=245428. Last accessed 5th December 2019.

performance results of a baseline system exploiting the transformation schemes and handcrafted rule activation patterns presented in Section 5.2, but with each sign tagged using the majority class label observed for that sign in our annotated data. In this setting, with the exceptions of those listed in Table 6.4, all signs were tagged with class label SSEV (left boundaries of subordinate clauses). Comparison of these results with those in the OB1 column indicates the contribution made by the automatic sign tagger to the simplification task. The *MUSST* column presents evaluation results for a reduced version of the MUSST sentence simplification system (described in Section 5.1.1.1, page 119).[4] MUSST implements several types of syntactic simplification rule. In the table, I focused on performance of the one which splits sentences containing conjoint (compound) clauses, which is used to simplify Type 1 sentences. I deactivated the other transformation functions (simplifying relative clauses, appositive phrases and passive sentences). STARS is a method for automatic simplification of Type 1 sentences which implements the sentence transformation schemes specified in Section 5.2.1 of this thesis. To identify the spans of compound clauses in input sentences and to implement the rule activation patterns used in the sentence transformation schemes, STARS uses the sequence tagging approach described in Section 4.3. Thus, STARS exploits machine-learned rule activation patterns. It is a fully automatic system, exploiting machine learning methods for sign tagging (Chapter 3) and for identification of the spans of compound clauses. The *STARS* column

[4]Available at https://github.com/carolscarton/simpatico_ss. Last accessed 7th January 2020. Experiments conducted in my evaluations were based on a version downloaded and modified in January 2018. I am not aware of any subsequent change made to the system since then.

in Table 6.3 presents evaluation figures for this sentence simplification method. OB1 is also an implementation of the sentence simplification method presented in Chapter 5, which exploits the handcrafted rule activation patterns described in Section 5.2.2 of that chapter. In Table 6.3, the *OB1* column displays the performance of this system when operating in fully-automatic mode, exploiting the sign tagger described in Chapter 3. The *Orcl* column displays the performance of the OB1 sentence simplification method when it exploits error-free sign tagging (an oracle).

Table 6.3: *System performance when simplifying Type 1 sentences*

| **Register** | **Bsln** | **MUSST** | **STARS** | **OB1** | **Orcl** |
|---|---|---|---|---|---|
| | | | **SARI** | | |
| Health | 0.201 | 0.124 | 0.309 | 0.362 | 0.514 |
| Literature | 0.203 | 0.087 | 0.190 | 0.202 | 0.229 |
| News | 0.119 | 0.171 | 0.478 | 0.596 | 0.623 |
| | | | **$F_1$-score** | | |
| Health | 0.362 | 0.281 | 0.532 | 0.495 | 0.613 |
| Literature | 0.150 | 0.101 | 0.286 | 0.208 | 0.262 |
| News | 0.233 | 0.237 | 0.623 | 0.690 | 0.706 |

Table 6.4: *Tags most frequently assigned to the signs in our annotated corpus*

| **Majority Tag** | **Signs** |
|---|---|
| CEV | [; or], [: but], [: and], [; but], [; and], [, but], [, and] |
| CLN | [or] |
| CMN1 | [, or] |
| CMV1 | [and] |
| ESEV | [,] |
| SPECIAL | [: that] |
| SSCM | [:] |

According to the $F_1$ metric, when transforming Type 1 sentences in the registers of health and literature, the output of OB1 is more similar to the gold standard than the output of the baseline (*Bsln*) is. For both evaluation metrics, in this task, the performance of OB1 also compares favourably with that of the reduced version of MUSST, which exploits a syntactic dependency parser. Calculated by comparing per-sentence Levenshtein similarity between sets of simplified sentences, two tailed paired sample $t$-tests revealed that the observed differences in performance between OB1 and MUSST and OB1 and *Bsln* are statistically significant for both $F_1$ and $SARI$ metrics for texts of all registers ($p \ll 0.01$). The only exception was when comparing the $SARI$ scores obtained by the *Bsln* and *OB1* systems when processing texts of the literary register ($p = 0.0604$).

When transforming Type 1 sentences in the register of health, the $F_1$-score of STARS, which exploits machine-learned rule activation patterns is greater than that of OB1, which uses handcrafted rule activation patterns. Use of two tailed paired sample t-tests indicates that this difference is statistically significant ($p = 0.01119$). The reverse is true when simplifying sentences of the news register ($p = 0.0004$). There is no statistically significant difference in the accuracy of the two systems when simplifying Type 1 sentences in literary texts ($p = 0.1739$).

Table 6.5 presents the accuracy of the methods derived using the SARI and $F_1$ metrics when simplifying Type 2 sentences. In this evaluation, the columns and rows of the table are similar to those of Table 6.3, though the evaluated simplification methods are those which use transformation schemes and rule activation patterns to detect and simplify $\text{complex}_{\overline{RF}}$ NPs in input sentences. In the case

Table 6.5: *System performance when simplifying Type 2 sentences*

| **Register** | **Bsln** | **MUSST** | **STARS** | **OB1** | **Orcl** |
|---|---|---|---|---|---|
| | | | **SARI** | | |
| Health | 0.207 | 0.020 | 0.182 | 0.285 | 0.296 |
| Literature | 0.168 | 0.008 | 0.051 | 0.204 | 0.289 |
| News | 0.434 | 0.056 | 0.194 | 0.451 | 0.467 |
| | | | **$F_1$-score** | | |
| Health | 0.231 | 0.063 | 0.281 | 0.306 | 0.315 |
| Literature | 0.572 | 0.000 | 0.248 | 0.516 | 0.791 |
| News | 0.583 | 0.141 | 0.373 | 0.577 | 0.629 |

of the MUSST system, the activated simplification rule was the one used to split sentences containing relative clauses, which is used to simplify Type 2 sentences.

The SARI evaluation metric indicates few statistically significant differences in the accuracy of the OB1 and *Bsln* systems when simplifying Type 2 sentences (Table 6.5). A statistically significant difference in performance was only evident for sentences of the health register, where $p = 0.036$. By contrast, differences between the accuracy scores obtained by OB1 and *MUSST* are statistically significant, in favour of OB1, when simplifying Type 2 sentences in texts of all registers ($p \ll 0.01$).

In terms of $F_1$, when simplifying Type 2 sentences in texts of the registers of literature and news, the *Bsln* baseline is more accurate than my approach (OB1). The performance of OB1 was superior to that of *Bsln* when processing texts of the health register. Differences in the accuracy of the OB1 and *Bsln* systems are statistically significant for texts of the registers of health and literature ($p < 0.0005$). For the task of simplifying Type 2 sentences, performance of the OB1 system is far superior to that of the reduced version of MUSST. The system

exploiting handcrafted rule activation patterns (OB1) is more accurate than the one exploiting machine-learned patterns (STARS) when simplifying Type 2 sentences in texts of all three registers. The differences are statistically significant ($p \ll 0.0001$ in all cases).

When considered over all text registers, the difference in $F_1$-scores obtained by the OB1 and STARS systems is statistically significant when simplifying Type 1 sentences and Type 2 sentences. In the former case, the STARS system tends to be superior while in the latter, the OB1 system is superior.

### 6.1.3 Evaluation of Individual Rules and Error Analysis

In this section, I report on the accuracy of the individual sentence transformation rules exploited by the OB1 and STARS systems when simplifying Type 1 and Type 2 sentences. I also present an error analysis of the OB1 and STARS systems and of the MUSST baseline system.

In this context, the accuracy of the rules is the ratio of the number of applications of each rule that led to the derivation of correct output sentences to the total number of applications of the rules. When simplifying Type 1 sentences, the rules based on handcrafted activation patterns used in OB1 have an overall accuracy of 0.6990. The rules based on machine-learned activation patterns used by the STARS system have an accuracy of 0.6981.

I categorised and quantified errors made by the *OB1*, *STARS*, and *MUSST* systems when simplifying Type 1 sentences. Here, each error is an incorrect analysis or transformation operation applied by the system when simplifying a

given input sentence given that the final multisentence simplification is less than 95% similar to that generated by linguists simplifying the same sentence. In this context, similarity is measured using the *sim* function defined in Section 6.1.2. Output generated by the simplification system in response to a given input sentence may be the product of multiple errors. Across all registers, when transforming Type 1 sentences, information about the five most frequent categories of error made by OB1 and STARS is presented in Tables 6.6 and 6.7, respectively. Examples of errors made by MUSST are presented in Table 6.8.

In Table 6.6, the columns provide error category labels (*Error category*), examples of the simplification of a given input sentence by linguists (*Human simplified*), examples of the simplification of that sentence by my system (*OB1 simplified*), the similarity of the two simplifications (*Similarity*), and the frequency of errors of this type in the test data (*Freq*). This information is provided for the five most frequent categories of error.

*Sign tagging* errors are those caused when OB1 fails to simplify a sentence correctly due to a failure to correctly tag the clause coordinator. *Incorrect transformation* errors are those caused when the activated transformation rule fails to generate correct output for some other reason. *Missing pattern* errors are those caused when OB1 makes no transformation of the input sentence despite the fact that the relevant sign of syntactic complexity has been correctly tagged. Overcoming such errors requires the addition of new transformation rules and activation patterns into the set used by OB1. The *left conjoin too wide* and *left conjoin too narrow* errors are those made when the patterns used by the trans-

Table 6.6: *Example errors when simplifying Type 1 sentences (OB1)*

| **Error category** | **Human simplified** | **OB1 simplified** | **Similarity** | **Freq (%)** |
|---|---|---|---|---|
| Sign tagging | Bloodstained knives were found in the kitchen. An axe was discovered in the bedroom near the bodies of the two men. | Bloodstained knives were found in the kitchen and an axe was discovered in the bedroom near the bodies of the two men. | 0.48 | 405 (53.78) |
| Incorrect transformation | 'How they came to be committed is not clear,' he said. 'That they were committed and committed by you is abundantly clear,' he said. | "How they came to be committed is not clear. They were committed and committed by you is abundantly clear," he said. | 0.84 | 82 (10.89) |
| Missing pattern | "The organisation was no stranger to the imposition of serious violence against those who might seek to challenge them. Few could afford to trifle with their wishes." | "The organisation was no stranger to the imposition of serious violence against those who might seek to challenge them and few could afford to trifle with their wishes." | 0.35 | 77 (10.23) |
| Left conjoin too wide | I said 'Maybe' because I wanted to know what he was talking about. I said 'Maybe' because I wanted to know who he was talking with. | I said 'Maybe' because I wanted to know what he was talking about. I said 'who he was talking with. | 0.49 | 66 (8.76) |
| Left conjoin too narrow | Most are tablets or liquid that you swallow. You may need an injection, a suppository (see page 29) or an inhaler. | Most are tablets or liquid that you swallow. Most are tablets or liquid that you may need an injection, a suppository (see page 29) or an inhaler. | 0.84 | 48 (6.37) |

formation rules incorrectly identify the left boundaries of compound clauses.

In the error analysis, I was able to distinguish *sign tagging* from *missing pattern* errors by examining the tagged versions of input sentences. When the clause coordinator is tagged as being of a different class, the simplification is a sign tagging error. When the clause coordinator is correctly tagged, the simplification is a *missing pattern* error.

Table 6.7 provides examples of errors made by the STARS system when simplifying Type 1 sentences. When no rule transformation pattern can be matched in the original sentence, the simplified sentence output by STARS is an empty string with no similarity to the human simplified sentence. This contrasts with the *missing pattern* errors made by the OB1 system (Table 6.6) in which the originals are returned as the simplified versions of the input sentences. When the simplified sentence output by STARS is an empty string, the original sentence is printed in italics in the *STARS simplified* column of Table 6.7.

By far the most frequent are *sign tagging* errors, which occur when the sign tagger fails to identify clause coordinators in input sentences. After this, tokenisation and PoS tagging errors are relatively common. These often involve incorrect sentence boundary identification in sentences containing direct speech. To illustrate, in the example presented in the second row of Table 6.7, the tokeniser incorrectly identifies the first question mark in sentence (46) as a sentence boundary.

(46) “I asked her ‘Suppose it does not work?’ and she said ‘Then there is

Table 6.7: *Example errors when simplifying Type 1 sentences (STARS)*

| **Error category** | **Human simplified** | **STARS simplified** | **Similarity** | **Freq (%)** |
|---|---|---|---|---|
| Sign Tagging | Such a preparation had become a rarity, said Mr Hughes. The NHS now encouraged doctors to prescribe proprietary brands, said Mr Hughes. | *Such a preparation had become a rarity, said Mr Hughes, and the NHS now encouraged doctors to prescribe proprietary brands.* | 0 | 357 (46.73) |
| Tokenisation/PoS Tagging | "I asked her 'Suppose it does not work?' She said 'Then there is nothing left for me'." | *"I asked her 'Suppose it does not work?' and she said 'Then there is nothing left for me'."* | 0 | 87 (11.39) |
| No compound clause detected | "This condition in a child is known as simple obesity. In my experience obesity in children is rarely simple and not a matter of diet alone." | *"This condition in a child is known as simple obesity but in my experience obesity in children is rarely simple and not a matter of diet alone."* | 0 | 86 (11.26) |
| Left conjoin too narrow | Further inquiries led them to Roberts's cousin, Peter Roberts, the manager of a Thetford dairy. It emerged that the two men had been having a homosexual affair for five years. | Further inquiries led them to Roberts's cousin, Peter Roberts, the manager of a Thetford dairy. Further inquiries led it emerged that the two men had been having a homosexual affair for five years. | 0.88 | 73 (9.55) |
| Incorrect transformation | He said: "She explained if she could not have the operation she would kill herself". He said: "She explained if I was not successful she would kill herself". | He said: "She explained if she could not have the operation." He said: "I was not successful she would kill herself." | 0.59 | 52 (6.81) |

nothing left for me'."

As a result, the simplification method is not able to identify the complex sentence simplified by the linguist in the test data.[5] In the remaining cases, errors can be attributed directly to the tagger presented in Section 4.3 which is designed to identify the spans of compound clauses. Although the cause is uncertain, some of the *incorrect transformation* errors may also be due to the implementation of the sentence transformation scheme which is used to simplify sentences containing compound clauses.

From the test data presented in Section 6.1.1, I categorised the errors made by the *MUSST* baseline system when processing 100 sentences of each type. The two main categories of error were caused by inaccurate syntactic parsing. This led to failures in detecting compound clauses in input sentences (91.67% of errors) and inaccuracies when transformation rules are applied to incorrectly identified syntactic constituents (8.33% of errors). The first of these categories causes total failure in the system to perform any transformation of Type 1 input sentences. An example of erroneous output generated by *MUSST* when transformations are applied to incorrectly parsed Type 1 sentences (the second category of error) are provided in the first row of Table 6.8. For comparison, human simplifications of these sentences are provided in the *human simplified* column of this table, while column *Sim.* displays the similarity of the automatically simplified sentence to the human simplified one, as computed using the *sim* function described in Section

[5]Processing of sentences such as (46) will pose difficulties for many NLP tasks. In this case, the tokenisation error adversely affects the simplification process.

6.1.2.

Table 6.8: *Transformations applied to incorrectly parsed sentences (*MUSST*)*

| **Transformation type** | **Human simplified** | ***MUSST* simplified** | **Sim.** | **Freq. (%)** |
|---|---|---|---|---|
| Compound clauses | Elaine Trego never bonded with 16-month- old Jacob, a murder trial was told. He was often seen with bruises, a murder trial was told. | Elaine Trego never bonded with Jacob. And Elaine Trego he was often seen with bruises, a murder trial was told. | 0.38 | (8.33) |
| Nominally bound relative clauses | And last night police said fellow officers had reopened their files on three unsolved murders. These police saw Kevin Cotterell caged. | And last night police caged said fellow officers had reopened their files on three unsolved murders. Police saw Kevin Cotterell. | 0.73 | (2.86) |

Overall, the transformation rules based on handcrafted activation patterns exploited by OB1 to simplify Type 2 sentences have an accuracy of 0.5829. As in the case when processing Type 1 sentences, two primary sources of error were found in the OB1 system when simplifying Type 2 sentences: the specificity of the rules, which limits their coverage; and the inability of the method to discriminate between signs of class CEV which link bound relative clauses and those which link independent clauses. The transformation rules based on machine-learned activation patterns used by STARS to simplify Type 2 sentences have an accuracy

of 0.5240.

Across all registers, when transforming Type 2 sentences, information about the five most frequent categories of error made by OB1 is presented in Table 6.9. Information about the most frequent categories of error made by STARS is presented in Table 6.10.

In Table 6.9, *sign tagging* errors are those caused when OB1 fails to simplify a sentence correctly due to a failure to correctly classify the left boundary of the relative clause. *Matrix NP too narrow* errors are a subset of those made when the applied transformation rule fails to correctly identify the left boundary of the complex$_{\overline{RF}}$ NP that the relative clause modifies. *Relative clause too narrow* errors are a subset of those made when the applied transformation rule fails to correctly identify the right boundary of the complex$_{\overline{RF}}$ NP that the relative clause modifies. As in Table 6.6, *Incorrect transformation* errors in Table 6.9 are those caused when the activated transformation rule fails to generate correct output for some other reason. *Missing pattern* errors are those that occur when none of the implemented rule activation patterns can be matched in the input sentence. They are indicative of incomplete coverage of the rules implemented by OB1. In the case of simplifying Type 2 sentences, I observe that the most frequent sources of error are caused by poor coverage of the implemented rules and errors in sign tagging.

In Table 6.10, as in Table 6.7, when STARS generates an empty string as the simplified version of an input sentence, the original sentence is printed in italics in the *STARS simplified* column. The most frequently occurring are *incorrect*

Table 6.9: *Example errors when simplifying Type 2 sentences (OB1)*

| **Error category** | **Human simplified** | **OB1 simplified** | **Similarity** | **Freq (%)** |
|---|---|---|---|---|
| Missing pattern | "Instead of a bouncy healthy boy his parents had this thrust upon them. This must have been beyond their wildest nightmares." | "Instead of a bouncy healthy boy his parents had this thrust upon them which must have been beyond their wildest nightmares." | 0.28 | 102 (44.15) |
| Sign tagging | These cells react to light and send electrical signals down tiny nerve fibres to the brain. (These fibres collect into the optic nerve.) | These cells react to light and send electrical signals down tiny nerve fibres (which collect into the optic nerve) to the brain. | 0.35 | 78 (33.77) |
| Matrix NP too narrow | And the hearing was told that a 14-year-old boy was also flying on the same frequency. That 14-year-old boy cannot be named. | And the hearing was told that a 14-year-old boy was also flying on the same frequency. Boy cannot be named. | 0.55 | 17 (7.36) |
| Incorrect transformation | Connie rarely left the cottage, but Janice was regularly seen walking her whippets. She bred her whippets as coursing dogs and showed at Crufts. | Connie rarely left the cottage, but Janice was regularly seen walking her whippets. Her she bred as coursing dogs and showed at Crufts. | 0.85 | 16 (6.93) |
| Relative clause too wide | Staff complained that the subsequent inquiry was serviced by managers and that it failed to call key witnesses. These managers had already defended Francis. | Staff complained that the subsequent inquiry was serviced by managers. These managers had already defended Francis and that it failed to call key witnesses. | 0.47 | 11 (4.76) |

Table 6.10: *Example errors when simplifying Type 2 sentences (STARS)*

| **Error category** | **Human simplified** | **STARS simplified** | **Similarity** | **Freq (%)** |
|---|---|---|---|---|
| Incorrect transformation | There are other leaflets available. These leaflets discuss respiratory tract infections, conjunctivitis and cystitis. | There are other leaflets available. Other leaflets available discuss respiratory tract infections, conjunctivitis and cystitis. | 0.92 | 59 (26.34) |
| Span tagging: No complex constituent detected | (The aorta is the large artery.) (That artery takes oxygen-rich blood from the heart chambers to the body.) The main coronary arteries divide into smaller branches. These branches take blood to all parts of the heart muscle. | (The aorta is the large artery. The large artery takes oxygen-rich blood from the heart chambers to the body.) *The main coronary arteries divide into smaller branches which take blood to all parts of the heart muscle.* | 0.55 | 23 (10.27) |
| Matrix NP too narrow | In 1993 Francis was made head of the Crack Awareness Team set up by the Association for the Prevention of Addiction charity. That charity gets £3.5 million from the Home Office. | In 1993 Francis was made head of the Crack Awareness Team set up by the Association for the Prevention of Addiction charity. Addiction charity gets £3.5 million from the Home Office. | 0.91 | 18 (8.04) |
| Matrix NP too wide | The £12,000 CAT job - funded by taxpayers - allowed him to build up a database of addicts. His dealers could supply these addicts. | The £12,000 CAT job - funded by taxpayers - allowed him to build up a database of addicts. His dealers could supply the database of addicts. | 0.89 | 17 (7.59) |
| Restrictive relative clause misclassified as non-restrictive | Complications can sometimes occur. Your surgeon will advise on the possible complications. | Your surgeon will advise on the possible complications. The possible complications can sometimes occur. | 0.85 | 16 (7.14) |

*transformation* errors. This can partly be explained by the fact that for Type 2 sentences containing subject relativised clauses, I implemented just one simple transformation rule. As in the example shown in Table 6.10, this sometimes leads to the generation of ungrammatical output. These errors could be addressed by a relatively simple post-editing step which preposes a determiner (e.g. *These*) to plural subject NPs, indicating that they are anaphoric to an antecedent in the preceding sentence.

The most obvious difference between the frequency distributions of error types made when using both STARS and OB1 to simplify Type 1 sentences and when using STARS to simplify Type 2 sentences is the relative absence of sign tagging errors in the latter case. I note that when using OB1 to simplify Type 2 sentences, a significant number of sign tagging errors are still made (33.77% of errors). This may indicate that the STARS system, exploiting machine-learned rule activation patterns, is less sensitive to sign tagging errors than OB1 when identifying the spans of complex$_{\overline{RF}}$ NPs. This may be a result of the contribution of other features to the model tagging complex$_{\overline{RF}}$ NPs. Handcrafted and machine-learned patterns have similar propensities to underestimate the spans of complex$_{\overline{RF}}$ NPs, with both methods making a similar proportion of *Matrix NP too narrow* errors (7.36% for OB1 and 8.04% for STARS).

As in the case when processing Type 1 sentences, the two main categories of error which occur when the MUSST baseline system is used to simplify Type 2 sentences were caused by inaccurate syntactic parsing. This led to failures in detecting complex$_{\overline{RF}}$ NPs in input sentences (97.14% of errors) and inaccu-

racies when transformation rules are applied to incorrectly identified syntactic constituents (2.86% of errors). An example of erroneous output generated by *MUSST* when transformations are applied to incorrectly parsed Type 2 sentences (the second category of error) are provided in the second row of Table 6.8 (page 167).

## 6.2 Automatic Estimation of Readability

Six readability metrics were used to estimate the impact of four sentence simplification methods (*MUSST*, *STARS*, *OB1*, and *orcl*) on the propositional density, reading grade level, syntactic complexity, and various aspects of cohesion of input texts. The selected metrics were:

- Propositional idea density (Brown *et al.*, 2008),[6]
- Flesch-Kincaid Grade Level (Kincaid *et al.*, 1986), obtained via the *style* package (Cherry and Vesterman, 1981)),[7]
- Four metrics from the Coh-Metrix package (McNamara *et al.*, 2014):[8]
  - *Syntactic simplicity*
  - Three metrics providing information about text cohesion:

[6] Calculated using CPIDR, available for download via a link at http://ai1.ai.uga.edu/caspr/. Last accessed 7th January 2020.

[7] *Style* is a Linux command-line utility, part of the GNU software suite.

[8] Calculated using the Coh-Metrix Web Tool at http://tool.cohmetrix.com/. Last accessed 7th January 2020.

- *Referential cohesion*, which measures the extent to which words and ideas overlap across sentences and across the entire text, forming explicit threads that connect the text for the reader (Lei *et al.*, 2014),
- *Deep cohesion*, which uses frequency counts of causal and intentional connectives and the causal and logical relationships expressed within the text to estimate readability (when the text contains many relationships but few connectives, it is more difficult to process as readers must infer the relationships between ideas in the text), and
- *Temporality*, which uses information on the number of inflected tense morphemes, temporal adverbs, and other explicit cues to estimate the consistency of tense and aspect in the text to assess the ease with which it can be processed and understood.

Selection of these metrics was motivated in several ways. Due to their design, I expect that the implemented sentence transformation schemes (Section 5.2.1) will generate output texts that are more readable in terms of *propositional idea density* and *syntactic simplicity*. *Flesch-Kincaid Grade Level* was selected because it has been widely used in previous evaluations of sentence simplification systems (e.g. Woodsend and Lapata (2011); Wubben *et al.* (2012); Glavas and Stajner (2013); Vu *et al.* (2014); Shardlow and Nawaz (2019)). There is a risk that the conversion of complex sentences into sequences of simple sentences will adversely affect text

cohesion, for example, by changing the sequencing of subjects (centers) occurring in adjacent sentences in the text. In this evaluation, given that these metrics are available and their computation is not expensive, I used the text cohesion metrics to observe the effects of the simplification methods on these phenomena.

Accessible texts are expected to have small values for propositional density and Flesch-Kincaid Grade Level and large values for the other four metrics. Readability scores were obtained for texts in their original form and the form output by the simplification methods when processing Type 1 sentences (Table 6.11) and Type 2 sentences (Table 6.12). In the tables, the *orig* columns present values of each metric obtained for the original versions of the texts.

Inspection of Tables 6.11 and 6.12 reveals that all of the automatic systems generate texts that are more readable, in terms of propositional density and Flesch-Kincaid Reading Grade level, than the originals. These metrics also indicate that OB1 compares favourably with the *MUSST* system when simplifying sentences of Type 1 and Type 2. For all registers, with the exception of the MUSST system processing Type 2 sentences in literary texts, the automatic sentence simplification systems generated texts with reduced *propositional idea density*, making them more readable than the originals. When transforming Type 1 and Type 2 sentences, of the fully automatic systems, the greatest reduction in propositional idea density was made by the STARS system.

Inspection of Table 6.11 reveals that the original versions of the input texts, estimated by the *referential cohesion* and *deep cohesion* metrics, are more readable than those generated by the fully automatic systems that transform Type 1

Table 6.11: *Estimated readability of text output when transforming Type 1 sentences*

| Register | Orig | MUSST | STARS | OB1 | Orcl |
|---|---|---|---|---|---|
| | | | **Propositional Idea Density** | | |
| Health | 0.523 | 0.521 | 0.468 | 0.510 | 0.503 |
| Literature | 0.593 | 0.588 | 0.569 | 0.588 | 0.592 |
| News | 0.505 | 0.502 | 0.480 | 0.483 | 0.482 |
| | | | **Flesch-Kincaid Grade Level** | | |
| Health | 8.9 | 7.4 | 6.4 | 6.0 | 5.4 |
| Literature | 10.3 | 7.1 | 4.6 | 5.4 | 6.0 |
| News | 9.6 | 7.9 | 5.4 | 5.4 | 5.3 |
| | | | **Referential Cohesion** | | |
| Health | 41.68 | 37.45 | 42.47 | 26.43 | 23.89 |
| Literature | 90.49 | 50.00 | 52.39 | 65.17 | 72.24 |
| News | 40.90 | 34.83 | 40.90 | 35.20 | 51.99 |
| | | | **Deep Cohesion** | | |
| Health | 96.16 | 94.41 | 93.32 | 92.07 | 90.66 |
| Literature | 72.91 | 68.79 | 66.28 | 64.63 | 63.68 |
| News | 56.36 | 54.38 | 46.41 | 48.40 | 46.02 |
| | | | **Syntactic Simplicity** | | |
| Health | 83.89 | 91.62 | 94.18 | 96.78 | 98.26 |
| Literature | 10.93 | 58.32 | 64.43 | 69.50 | 55.17 |
| News | 46.81 | 66.64 | 82.29 | 89.07 | 85.77 |
| | | | **Temporality** | | |
| Health | 52.39 | 51.20 | 50.80 | 54.38 | 53.98 |
| Literature | 63.31 | 81.86 | 82.64 | 72.57 | 76.42 |
| News | 27.76 | 40.52 | 44.43 | 30.15 | 35.94 |

sentences. The effect on *referential cohesion* may be explained by the fact that the transformation operations increase the numbers of sentences in the texts, reducing the amount of word overlap between adjacent sentences. These findings can be taken as evidence that the transformation operations have a disruptive effect on the cohesion of a text. It was noted for texts of all registers. With respect to *referential cohesion*, when transforming Type 1 sentences, only the STARS and *orcl* systems are able to generate texts of the registers of health and

Table 6.12: *Estimated readability of text output when transforming Type 2 sentences*

| Register | Orig | MUSST | STARS | OB1 | Orcl |
|---|---|---|---|---|---|
| | | | **Propositional Idea Density** | | |
| Health | 0.500 | 0.493 | 0.470 | 0.499 | 0.500 |
| Literature | 0.597 | 0.599 | 0.564 | 0.592 | 0.594 |
| News | 0.489 | 0.486 | 0.474 | 0.478 | 0.480 |
| | | | **Flesch-Kincaid Grade Level** | | |
| Health | 8.4 | 8.3 | 7.1 | 8.3 | 8.3 |
| Literature | 9.9 | 9.4 | 5.6 | 6.8 | 6.9 |
| News | 10.3 | 9.6 | 8.3 | 7.7 | 7.9 |
| | | | **Referential Cohesion** | | |
| Health | 39.74 | 45.62 | 72.57 | 38.97 | 41.29 |
| Literature | 70.54 | 55.57 | 46.41 | 33.72 | 66.28 |
| News | 24.51 | 32.64 | 49.20 | 18.41 | 44.04 |
| | | | **Deep Cohesion** | | |
| Health | 87.70 | 87.90 | 81.86 | 87.49 | 87.49 |
| Literature | 81.59 | 59.48 | 77.34 | 77.94 | 77.04 |
| News | 63.31 | 65.17 | 55.96 | 61.79 | 59.87 |
| | | | **Syntactic Simplicity** | | |
| Health | 68.44 | 68.44 | 80.51 | 69.85 | 68.79 |
| Literature | 22.36 | 56.36 | 58.71 | 58.71 | 46.41 |
| News | 38.59 | 41.29 | 58.71 | 81.86 | 85.77 |
| | | | **Temporality** | | |
| Health | 66.28 | 62.17 | 62.55 | 65.17 | 64.06 |
| Literature | 65.17 | 28.10 | 80.23 | 58.71 | 64.06 |
| News | 28.10 | 30.50 | 31.92 | 27.76 | 30.15 |

news, respectively, that are more accessible than the originals. For text of the health register, use of *MUSST* harms readability considerably less than OB1, while the reverse is true when transforming literary texts. When transforming Type 2 sentences (Table 6.12), STARS and *MUSST* generated texts that were more referentially cohesive than the originals, in the registers of health and news. Use of the OB1 system adversely affects the referential cohesion of the input texts. From this, it can be inferred that the transformation, using handcrafted rule ac-

tivation patterns, of long sentences with many concept mentions into sequences of shorter sentences, each with fewer concept mentions, reduces cohesion by spacing out these mentions over multiple sentences and reducing their repetition in adjacent ones. The data in Tables 6.11 and 6.12 shows that, with respect to the *deep cohesion* metric, texts generated by STARS and OB1 are not as readable as the originals (sentences of both types) or those generated by *MUSST* (Type 1 sentences). One possible reason for this is that the transformations performed by STARS and OB1 generate texts containing fewer connectives while those performed by *MUSST* do not. When splitting a sentence containing a compound clause into two, *MUSST* preserves the conjunction as the first word of the second output sentence. This can improve the readability of the output text.

The statistics presented in Tables 6.11 and 6.12 indicate that for all registers, the sentence simplification systems generate texts with greater *syntactic simplicity* than the originals. Overall, the texts generated by STARS and OB1 are indicated to be much more syntactically simple than those generated by *MUSST*.

When transforming Type 1 sentences, the *temporality* columns in Table 6.11 indicate that texts generated by the automatic systems are more consistent in terms of tense and aspect than the originals. It can be observed that the *MUSST* system brings greater improvements in this metric than OB1, except when processing health texts, while the STARS system is superior to MUSST when processing literary and news texts. This implies that the transformation operations implemented using the handcrafted rule activation patterns (Sections 5.2.1–5.2.2) introduce more inconsistencies with respect to tense and aspect than those used

by *MUSST* or STARS. When transforming Type 2 sentences, statistics in Table 6.12 indicate that text readability, as estimated using the *temporality* metric can be improved through simplification of literary and news texts using the STARS system.

Analysis of the results presented in this section enables a ranking of the systems with respect to the readability of their output when simplifying Type 1 and Type 2 sentences. In both cases, STARS is the best performing system, followed by OB1 and MUSST. Overall, texts are least readable in their original forms. As a semi-automatic system based on an oracle, I exclude the *orcl* system from this ranking, noting that it is superior to OB1 when simplifying Type 1 sentences but inferior to STARS regardless of the type of sentence being simplified. This is a marked contrast to the performance figures obtained in intrinsic evaluation of the systems using $F_1$-score (Section 6.1.2).

## 6.3 Reader Opinions

Following human-centred evaluation methods used in previous work (e.g. Angrosh *et al.* (2014); Wubben *et al.* (2012); Feblowitz and Kauchak (2013); Shardlow and Nawaz (2019)), I used the output of OB1 to create items surveying the extent to which readers agreed with five statements about the grammaticality, comprehensibility, and meanings of sentences in their original and simplified forms.[9] Figure 6.1 displays one such survey item. Each participant in this assessment

[9]These criteria are analogous to *fluency*, *simplicity*, and *meaning preservation*, respectively, used by Angrosh *et al.* (2014).

task provided opinions for each of 150 sentences that had been transformed by the sentence simplification method. As a result, this aspect of the evaluation ignores potentially complex sentences that the system failed to transform. This failing is mitigated by the comparison of system output with human simplifications described in Section 6.1.

In the evaluation based on reader opinions, five participants each responded to eight items[10] in nineteen surveys. Four of the participants were fluent non-native speakers of English, while one was a native speaker.[11]

I converted participants' extent of agreement with the opinions to integer values ranging from 1 for strong disagreement to 5 for strong agreement. Overall, participants grudgingly agreed that sentences generated by OB1 are easy to understand (95% CI 3.789, 4.050) and collectively have the same meaning as the original sentences (95% CI 3.721, 4.017). Although derived from a smaller number of participants, this compares favorably with agreement scores obtained for various text simplification systems in experiments conducted by Saggion *et al.* (2015).

Participants rated OB1's output as most comprehensible for Type 1 sentences of the news register ($\mu = 4.053$). They rated it as least comprehensible for Type 2 sentences ($\mu < 3.9$), especially for texts of the literary register ($\mu = 3.683$). Participants perceived that sentence transformations made by OB1 preserved meaning better for Type 1 sentences ($\mu = 3.9853$) than Type 2 sentences ($\mu =$

[10]Survey 19 contained six items
[11]I was the native speaker.

Figure 6.1: *Opinion survey item*

Although Fraser told police that he had his hands around his wife's throat for only five to eight seconds, medical experts said it was likely that the attack lasted longer. Mrs Fraser was advised to seek an exclusion order against her husband. Mrs Fraser had bruising and burst blood vessels in her eyelids. Since his wife's disappearance, Fraser has co-operated with the police and made appeals for information.

Mrs Fraser, who had bruising and burst blood vessels in her eyelids, was advised to seek an exclusion order against her husband.

| | Strongly disagree | Disagree | Neither agree nor disagree | Agree | Strongly agree |
|---|---|---|---|---|---|
| As a group, the blue sentences are grammatically correct. | ○ | ○ | ○ | ○ | ○ |
| The green sentence is grammatically correct. | ○ | ○ | ○ | ○ | ○ |
| As a group, the blue sentences are easy to understand. | ○ | ○ | ○ | ○ | ○ |
| The green sentence is easy to understand. | ○ | ○ | ○ | ○ | ○ |
| As a group, the blue sentences have the same meaning as the green sentence. | ○ | ○ | ○ | ○ | ○ |

3.7511). Overall, transformations were felt to best preserve meaning when applied to Type 1 sentences of the news register ($\mu = 4.053$). Our participants were most undecided ($\mu = 3.1111$) about the preservation of meaning in output sentences derived from Type 2 sentences in the health register.

Participants broadly agreed that sentences output by OB1 are grammatical (95% CI 4.031, 4.250) but that the original sentences were already easy to understand (95% CI 4.318, 4.443). They also strongly agreed that the original versions of the sentences were grammatical (95% CI 4.686, 4.769). Opinions expressed in the surveys indicate that participants found the original sentences significantly more comprehensible than those generated using OB1 ($p \ll 0.05$). I noted many cases where participants agreed equally strongly that sentences were easy to understand in both their original and simplified forms, despite the fact that, objectively, the latter contained fewer complex constituents. One possible explanation for this is that survey participants were not first provided with example sentences demonstrating different levels of complexity or comprehensibility. Access to such examples may have elicited better informed judgments about the relative complexities of the presented sentences.

I examined correlations (Pearson's correlation coefficient) between different variables in the opinion survey. The three most closely correlated ($0.6840 \leq r \leq 0.8168$) were between:

1. the perceived comprehensibility of sentences generated by OB1 and the perceived grammatical correctness of those sentences,

2. the perceived comprehensibility of sentences generated by OB1 and the perceived extent to which those sentences preserved the meanings of the originals, and

3. the perceived extent to which automatically generated sentences were grammatical and the perceived extent to which they preserved the meanings of the originals.

I found no linear relationship between the similarities of system-generated simplifications to gold standard simplifications[12] and either the perceived accessibility or the perceived grammaticality of those simplifications (Pearson's $r = 0.1716$ and $r = 0.0625$, respectively). There is a small linear relationship between the similarities of system-generated simplifications to gold standard simplifications and the extent to which readers perceived that the system-generated simplifications preserve the meanings of the original sentences ($r = 0.3039$). This correlation was slightly closer for simplifications of Type 2 sentences ($r = 0.4705$).

My observation from the reader opinion survey is that, overall, participants found the output of the OB1 system to be usable. It was generally agreed to be grammatical, to be comprehensible, and to preserve the meanings of the original sentences. The results of the opinion surveys tend to reinforce the findings of my comparison of system output with human-produced text simplifications (Section 6.1).

[12]Measured using the *sim* function presented in Section 6.1.2.

## 6.4 Contribution to Research Questions RQ-3 and RQ-4

The work described in this chapter makes contributions to two of the research questions set out in Chapter 1 of the thesis. With respect to **RQ-3**:

> *To what extent can an iterative rule-based approach exploiting automatic sign classification and handcrafted patterns convert sentences into a form containing fewer compound clauses and fewer complex$_{\overline{RF}}$ NPs?*

it completes the response begun in Chapter 5, which presented an automatic method for sentence simplification based on automatic sign classification (Chapter 3) and handcrafted rule activation patterns (Section 5.2.2). The current chapter presented an evaluation of this method (Sections 6.1–6.3).

This evaluation revealed that when simplifying sentences to reduce the number of compound clauses that they contain, performance of the method exploiting handcrafted rules is register-dependent, with superior precision and recall[13] when processing texts of the news and health registers, but poor performance when simplifying sentences in literary texts. For texts of all three registers, the method is far less reliable when simplifying sentences which contain complex$_{\overline{RF}}$ NPs (Type 2 sentences) than it is when simplifying Type 1 sentences. In the former setting,

[13]Obtained for calculation of the $F_1$-scores listed in Section 6.1.2 but not reported there. Here, precision reperesents the correctness of simplifications generated by the system while recall represents the degree to which human simplifications are correctly generated by the system.

recall is slightly better than 0.5 when processing news articles and significantly worse than 0.5 for texts of the other two registers.

I used the automatic sign tagger to estimate the numbers of compound clauses and relative clauses removed by the different variants of the $transform$ function used in the sentence simplification algorithm (Algorithm 1, Section 5.2). This indicated that use of $transform_{CEV}$ led to a reduction in the number of compound clauses in the input sentences to just 55.58% of their original number. However, use of this function to simplify sentences containing compound *relative* clauses led to the generation of additional finite relative clauses (8% of the total estimated number of finite relative clauses in the output).[14]

The function $transform_{SSEV}$ was less effective than $transform_{CEV}$. Output texts were indicated to contain as many as 90.12% of the relative clauses identified in the input. However, it should be noted that these figures include all finite subordinate clauses and not just the subset of these which are non-restrictive and modify $\text{complex}_{\overline{RF}}$ NPs.

To improve the accuracy of these estimations, I used the taggers described in Chapters 3 and 4 to count the numbers of automatically tagged clause coordinators and left boundaries of subordinate clauses occurring in automatically tagged compound clauses and $\text{complex}_{\overline{RF}}$ NPs in input texts and the corresponding output texts produced by OB1. When simplifying Type 1 sentences, the output of OB1 contains just 25.39% of the clause coordinators occurring in compound

[14]Simplification of sentences containing compound relative clauses using only $transform_{CEV}$ generates simplified sentences containing one relative clause for each conjoin of the original compound. As a result, there is an increase in the number of different relative clauses occurring in the text.

clauses in the input texts. When simplifying Type 2 sentences, the output of OB1 still contains 72.82% of the left boundaries of subordinate clauses modifying complex$_{\overline{RF}}$ NPs in the input texts.

My response to **RQ-3** is that the approach based on handcrafted rule activation patterns can reduce the numbers of compound clauses in input texts to a moderate extent (by around 45%) but can only reduce the number of complex$_{\overline{RF}}$ NPs to a limited extent (by around 27%).

This chapter completes my three-part response to **RQ-4**:

> *How does the accuracy of automatic sentence simplification compare when using a machine learning approach to detect the spans of compound clauses and complex$_{\overline{RF}}$ NPs and when using a method based on handcrafted patterns?*

It presented evaluation statistics for the OB1 system which exploits handcrafted rule activation patterns and the STARS system which exploits machine-learned rule activation patterns. These statistics show that, while the two approaches have similar levels of accuracy, there are differences between them. The method based on handcrafted rule activation patterns is significantly more accurate than the one based on machine-learned patterns when simplifying Type 2 sentences. One reason for this is that the tagging approach used to identify the spans of complex$_{\overline{RF}}$ NPs (Chapter 4) has relatively poor accuracy ($0.4268 \leq F_1 \leq 0.6979$), which adversely affects the sentence simplification approach that exploits it. It should be noted that in the data used to train this tagger, only

25.35% of the annotated sequences (678 of 2674 sequences) contain complex$_{\overline{RF}}$ NPs. The derivation of tagging models from larger training sets may lead to improvements in performance of both the tagger and the dependent sentence simplification system.

My findings from the evaluation of systems simplifying Type 1 sentences are less clear cut. For this task, the STARS system exploited a relatively accurate method to identify the spans of compound clauses ($0.7303 \leq F_1 \leq 0.8383$). In the sentence simplification task, STARS was significantly more accurate than OB1 when processing texts in the register of health. When processing literary texts, there was no statistically significant difference in accuracy between STARS and OB1. STARS was significantly less accurate than OB1 when processing texts of the news register.

In terms of readability, when simplifying Type 1 sentences, STARS's output has a lower propositional density than OB1's but the two variant systems generate output at similar reading grade levels (output from OB1 being slightly more readable when processing text of the health register and output from STARS being slightly more readable when processing literary text). Overall, STARS's output is more cohesive than OB1's (measured using both *referential cohesion* and *deep cohesion metrics*) and easier to process and understand with respect to the temporal information that it conveys when applied to literary and news texts. The reverse is true when considering texts of the health register. For texts of all registers, the output generated by OB1 is syntactically simpler than that generated by STARS.

When simplifying Type 2 sentences, in terms of cohesiveness, OB1's output is more cohesive when measured using the *deep cohesion* metric than the output of STARS but less cohesive when measured using *referential cohesion*. The output generated by STARS is also of a lower reading grade level when processing texts of all registers except news and is easier to process and understand with respect to the temporal information that it conveys (measured using the *temporality* metric) than the output of OB1. Output generated by STARS when simplifying both types of sentences is propositionally less dense than that generated using OB1. Overall, and by a narrow margin with respect to these readability metrics, the STARS system generates more readable output than OB1.

# Chapter 7

# Extrinsic Evaluation

In Chapter 6, I presented an intrinsic evaluation of my approach to sentence simplification which relies on three main methods: the use of overlap metrics (e.g. Levenshtein distance (Levenshtein, 1966) and SARI (Xu *et al.*, 2016)) to compare system output with human simplified texts; automated assessments of the readability of system output; and surveys of human opinions about the grammaticality, comprehensibility, and meaning of system output. In previous work on text simplification, researchers have also used other human-centred evaluation methods, including behavioural studies of reading such as eye tracking (Klerke *et al.*, 2015; Timm, 2018) and reading comprehension testing (Orăsan *et al.*, 2018).

Assessing the performance of text simplification systems by comparing their output with human-produced gold standards, by using automatic methods to estimate the readability of their output, and by means of human-centred evaluation methods poses several difficulties.

Previous research in NLP evaluation has highlighted the difficulties of using overlap metrics to compare system output with human generated translation and text simplification (e.g. Rapp, 2009; Wieting *et al.*, 2019). The development of gold standard datasets for text simplification is problematic because they are

difficult to produce and numerous variant simplifications are acceptable. As a result, previous evaluations based on these gold standards may not accurately reflect the usefulness of the simplification system being evaluated. Even when human editors are provided with detailed guidelines for the simplification task, there is still likely to be a variety of means by which the editor might simplify a text to produce a reference simplification. Further, due to the difficulty of the human simplification task, it may be that evaluation measures such as BLEU and SARI are unable to exploit sufficiently large and representative sets of reference simplifications. In my thesis, overlap metrics were used to evaluate systems with respect to their ability to implement the transformation schemes presented in Chapter 5. However, this is distinct from their ability to simplify texts because, in some cases, the correct implementation of these schemes may lead to an unintended loss of information about temporal sequencing or other discourse relations in the system output. As a result, such output may become more difficult to understand or may express a different meaning from the original text.

The evaluation of text simplification systems using automatic readability metrics is problematic because the extent to which all but a handful of readability metrics correlates with human reading comprehension is uncertain.

Human-centred evaluation of text simplification methods is also difficult. Evaluation via opinion surveys of readers is problematic because participants may have varying expectations about the upper and lower limits of sentence complexity, making responses to Likert items unreliable. Participants also vary in terms of linguistic ability and personal background knowledge. These variables, which

affect reading behaviour and may influence responses to opinion surveys, are difficult to control. When using methods such as eye tracking, previous work has shown that differences in reading behaviour also depend on participants' reading goals (Yeari *et al.*, 2015). This variable can be controlled by requiring participants in reading studies to perform uniform tasks such as responding to text-related opinion surveys or multiple choice reading comprehension questions. One adverse effect of this is that these evaluations may be of limited validity when considering the usefulness of system output for other purposes. While we may learn from a study whether a sentence simplification method improves participants' performance in answering short reading comprehension questions, it is not clear whether similar benefits would be obtained in terms of readers' abilities to be entertained by the text or to understand it well enough to be able to summarise it for friends.

Given that text simplification is usually made for a particular purpose, the evaluation method used should offer insights into the suitability of the text simplification system for this purpose. Extrinsic evaluation offers the possibility of meeting this requirement. Previous work in NLP has claimed that text simplification can be used to improve automatic text processing (e.g. Vickrey and Koller, 2008; Evans, 2011; Niklaus *et al.*, 2016; Hasler *et al.*, 2017), though the evidence for this has been fairly limited. In this chapter, I explore whether different implementations of my approach to sentence simplification can facilitate three NLP tasks: multidocument summarisation (MDS), semantic role labelling (SRL), and information extraction (IE).

I integrated the sentence simplification method into the three NLP applica-

tions as a preprocessing step and evaluated its impact on each of these as a response to research question **RQ-5**, which is concerned with determining whether or not my approach to automatic sentence simplification can facilitate subsequent text processing tasks.

I chose to extrinsically evaluate the OB1 system which exploits handcrafted rule activation patterns rather than the STARS system which exploits machine-learned rule activation patterns. The reason for this selection is that one of the evaluations that I performed is based on an automatic semantic role labelling task. Unfortunately, existing annotated data are unsuitable for extrinsic evaluation via a semantic role labelling system (see Section 7.3.1 for more details). As a result, manual evaluation of the SRL system when processing automatically simplified text was required. This manual evaluation is a time consuming and labour-intensive task that requires access to output from the relevant sentence simplification method. Development of the STARS system began relatively late in my research and was ongoing at the point when manual evaluation of the semantic role labeller needed to begin. For this reason, despite the fact that the STARS system proved to be more accurate than OB1 when simplifying Type 1 sentences,[1] I evaluated the OB1 system extrinsically via the SRL task. For consistency, I evaluated OB1, rather than STARS, extrinsically in all of the experiments described in this chapter.

In Section 7.1, I present a brief overview of previous related work. In the remainder of the chapter, I use a consistent structure to present my extrinsic

[1]Type 1 sentences in texts of the registers of health and literature.

evaluation experiments. In each of Sections 7.2, 7.3, and 7.4, I begin with a description of the NLP task by which the evaluation is being made. This is followed by descriptions of the test data used and the NLP systems performing each of the tasks. After this, I provide motivation for integration of the sentence simplification method as a preprocessing step in the NLP task and describe the evaluation method to be used in each case. I present and discuss the results of each experiment.

## 7.1 Previous Related Work

In previous work, researchers have sought to determine whether or not a preprocessing step based on text simplification can facilitate subsequent natural language processing. In this thesis, one of my aims is to investigate the impact of a system simplifying sentences which contain compound clauses. Hogan (2007) and Collins (1999) observed that, for dependency parsers, dependencies involving coordination are identified with by far the worst accuracy of any dependency type ($F_1$-*score* $\approx 61\%$). This is one factor motivating my research in this direction.

Sentence simplification has been applied as a preprocessing step in neural machine translation and hierarchical machine translation (Hasler *et al.*, 2017). However, it should be noted that the type of simplification applied in their work was crowd-sourced sentence compression. One contribution of the experiments that I describe in the current chapter is an investigation of the use of a fully automatic information-preserving approach to sentence simplification as a preprocessing step in the NLP applications.

Vickrey and Koller (2008) applied their sentence simplification method to improve performance on the CoNLL-2005 shared task on semantic role labelling.[2] For sentence simplification, their method exploits full syntactic parsing with a set of 154 parse tree transformations and a machine learning component to determine which transformation operations to apply to an input sentence. They find that a semantic role labelling system based on a syntactic analysis of automatically simplified versions of input sentences outperforms a strong baseline. In their evaluation, Vickrey and Koller (2008) focus on the overall performance of their semantic role labelling system rather than on the particular contribution made by the sentence simplification method. In this thesis, I isolate sentence simplification as a preprocessing step and investigate its impact on the SRL task.

In previous work, sentence simplification methods have also been shown to improve the performance of a variety of automatic machine translation (Štajner and Popović, 2018) and information extraction systems (Evans, 2011; Niklaus *et al.*, 2016).

Štajner and Popović (2018) showed that high quality simplification of input sentences containing relative clauses, obtained from the RegenT simplifier (Siddharthan, 2011), can improve the quality of output of English-to-German and English-to-Serbian neural machine translation systems. This improvement was evidenced by a reduction in the amount of human post-editing needed to correct the output of the MT systems. The findings of their experiment were complex, revealing that the quality of the MT output also depends on the quality of the

[2] http://www.lsi.upc.edu/~srlconll/spec.html. Last accessed 8th January 2020.

simplification performed. Further, the simplification of more complex relative clauses (e.g. those in which the relative pronoun is distant from the modified head noun or in which there are multiple possible antecedents of the relative pronoun) led to improvements in the output of the MT systems. By contrast, the simplification of less complex phrases evoked no improvement in this output.

Niklaus *et al.* (2016) applied automatic sentence simplification to improve the NLP task of open relation extraction by making input texts more amenable to syntactic analysis using a dependency parser. The types of simplification that they implemented included the conversion of sentences containing clause compounds and various types of finite and non-finite relative clauses into collections of simpler sentences. They showed that this type of simplification enables state-of-the-art open relation extraction systems to obtain improved accuracy and reduced information loss when processing complex sentences.

In earlier work, Evans (2011) integrated a domain-tuned sentence simplification system into a clinical information extraction system. These tools were designed to perform the information extraction task described later in this chapter. Evans showed that integrating a sentence simplification step into his system improved its performance in the subsequent IE task. I provide more details about Evans's previous work in Section 7.4.

## 7.2 Multidocument Summarisation

Multidocument summarisation is the task of automatically generating summaries of clusters of documents. In task 2 of DUC-2004,[3] "short" summaries with a maximum length of 665 characters were to be generated for each of 50 document clusters, each containing ten topic-related documents.[4]

In my extrinsic evaluation experiment, I used the MEAD (Radev *et al.*, 2006) automatic multidocument summarisation tool to generate summaries from:

1. Clusters comprising the original news articles. In this context, I refer to the MDS system as MEAD.

2. Clusters comprising news articles that were first processed using the OB1 sentence simplification system, exploiting handcrafted rule activation patterns and simplifying both Type 1 and Type 2 sentences (Section 5.2). In this setting, documents in each cluster are expected to contain reduced numbers of compound clauses and complex$_{\overline{RF}}$ NPs. In this context, I refer to the MDS system as MEAD$_{OB1}$.

3. Clusters comprising news articles that were first processed using the OB1 sentence simplification system, exploiting handcrafted rule activation patterns and simplifying only Type 1 sentences. In this setting, utilising only

[3]Information about the DUC conferences is accessible from https://www-nlpir.nist.gov/projects/duc/index.html. Last accessed 8th January 2020. Guidelines about the tasks presented in DUC-2004 are available at https://www-nlpir.nist.gov/projects/duc/guidelines/2004.html. Last accessed 8th January 2020.

[4]This contrasts with "very short" summaries, which have a length of 75 characters, to be generated in task 1 of the DUC-2004 evaluation.

the more accurate of the two types of sentence transformation, documents in each cluster are expected to contain reduced numbers of compound clauses. In this context, I refer to the MDS system as $\text{MEAD}^{CEV}_{OB1}$.

MEAD was selected for this purpose because it is freely available, it works out of the box, and it has been used as a baseline system by many NLP researchers (e.g. Reeve *et al.*, 2007; Abu-Jbara and Radev, 2011; Gerani *et al.*, 2014). In this section, of the two experimental settings, I focus on the performance of $\text{MEAD}_{OB1}$ rather than $\text{MEAD}^{CEV}_{OB1}$. In this setting, the simplification method (OB1) is expected to have applied a larger number of transformation operations than $\text{OB1}_{CEV}$. As a result, OB1 has the potential to transform input sentences into larger numbers of shorter sentences which may be packed more effectively into a 665-character summary by $\text{MEAD}_{OB1}$. In the other two experiments, based on the SRL and IE tasks (Sections 7.3 and 7.4, respectively), I extrinsically evaluate the variant of OB1 which only simplifies Type 1 sentences.

### 7.2.1 Test Data (MDS)

Of the 50 topic-related document clusters available to participants in task 2 of DUC-2004, I had access to 38, which were used for testing in my experiments. These document clusters contain 248 630 words in total. Each cluster represents a news 'topic' and contains ten news articles, each with an average length of 654 words ($\sigma = 434.01$). The task guidelines specify that output summaries of the clusters should be "general" and not focused on any particular aspect of the news stories. Table 7.1 provides more detailed information on the document

clusters to be summarised. Columns *Docs* and *Sents* display the total number of documents and sentences contained in the 38 document collections. Column *CEV/SSEV Sents* lists the total number of sentences in the document clusters that contain one or more of either clause coordinators or left boundaries of subordinate clauses. Column *Tokens* displays the total number of words, punctuation marks, and signs of syntactic complexity in the document clusters. Columns *CEV* and *SSEV* respectively show the numbers of clause coordinators and left boundaries of subordinate clauses occurring in the dataset.

Table 7.1: *Characteristics of the test data used for extrinsic evaluation of the sentence simplification method with respect to the multidocument summarisation task*

| **Docs** | **Sents** | **CEV/SSEV Sents** | **Tokens**[5] | **CEV** | **SSEV** |
|---|---|---|---|---|---|
| 380 | 10 899 | 4 417 | 251 128 | 1 002 | 5 143 |

### 7.2.2 Multidocument Summarisation System

For extrinsic evaluation of my sentence simplification method, I used MEAD[6] (Radev *et al.*, 2006) to generate multidocument summaries of the document clusters appearing in the test data used for task 2 of DUC-2004. MEAD is a publicly available and customisable multidocument summarisation system based on sentence extraction. In their approach, sentences are scored according to a variety of linguistic features encoding information on the position and length of the sen-

[5]Here, *tokens* denotes words, punctuation marks, and signs.

[6]`http://www.summarization.com/mead/download/MEAD-3.12.tar.gz`, downloaded from `http://www.summarization.com/mead/`. Last accessed 8th January 2020.

tence, the similarity of the sentence to the first sentence in the document, and its similarity to a centroid which can be viewed as an averaged vector representation of every sentence in the document.

When running MEAD, I optimised the system in terms of the length of the summaries produced when summarising collections of news articles in their original forms. The length of summaries generated by MEAD is customisable, but is specified in terms of words rather than characters. MEAD performed best in the intended task of generating summaries with a length of 665 characters when summary length was set at 130 words.

### 7.2.3 Motivation (Sentence Simplification for MDS)

My intuition was that extractive summarisation methods, which automatically construct summaries by selecting a set of sentences from source documents and then sequencing them to form a summary of those documents, can benefit from the types of sentence simplification implemented in my approach (Section 5.2.1). In the context of MDS, in the original documents to be summarised, long sentences may contain multiple propositions of varying degrees of relevance. Sentence simplification can generate output texts in which long complex sentences are converted into sequences of short simple sentences. Given a maximum word limit, extractive summarisation systems would, theoretically, be able to select shorter sentences containing more relevant propositions and ignore sentences focused on less relevant propositions. In this way, the summaries generated from syntactically simplified texts may be more focused and relevant than those derived from

the original documents.

The average lengths of sentences in the original and automatically simplified versions of the document clusters are 41.38 and 39.81 words/tokens, respectively. This is not a statistically significant difference ($p = 0.08$). However, my use of readability metrics such as propositional "idea" density (Covington, 2012) to evaluate the original and simplified versions of the documents in the test data lends some marginal support to my intuition. The 38 original document clusters in my test data have a mean propositional density of 0.4706 ideas per word and 13.127 ideas per sentence. The simplified versions of these clusters produced by OB1 have a propositional density of 0.4179 ideas per word and 11.909 ideas per sentence. Paired sample two-tailed t-tests show that these differences are statistically significant ($p \ll 0.05$ in both cases). This implies that propositions are distributed more widely, potentially over different sentences, in the simplified versions of the document clusters than they are in the original versions.

### 7.2.4 Evaluation Method (MDS)

I evaluated performance of the MDS system by comparison of its output with human-produced summaries of the document clusters. For this purpose, I used ROUGE[7] evaluation metrics (Lin, 2004) to assess the similarities of summaries generated by MEAD in the three settings to summaries generated by humans from the original document clusters. In terms of statistically significant differences between the ROUGE scores of different versions of MEAD, I observed no differences

[7] I used the implementation available at https://github.com/kavgan/ROUGE-2.0. Last accessed 8th January 2020.

when ROUGE was based on unigrams (ROUGE-1), bigrams (ROUGE-2), or trigrams (ROUGE-3). I did not investigate the potential impact of other variants of the metric, such as those exploiting stop word lists or synonym detection.

### 7.2.5 Results (MDS)

There was no statistically significant difference between the accuracy of MEAD and MEAD$_{OB1}$ ($ROUGE1 = 0.3449$ and 0.3453, respectively, $p = 0.4665$). There was also no statistically significant difference between the accuracy of MEAD and MEAD$^{CEV}_{OB1}$. For MEAD$^{CEV}_{OB1}$, $ROUGE1 = 0.3439$ ($p = 0.6279$).

In terms of the accessibility of the output summaries, there is no statistically significant difference between the lengths, numbers of ideas expressed, or propositional densities of output summaries generated by MEAD and MEAD$_{OB1}$ or MEAD$^{CEV}_{OB1}$ ($p > 0.1$ and $p > 0.3$ in all cases, respectively). The fact that there were statistically significant differences between the per sentence propositional density scores of the input sentences in the original and simplified versions of the document clusters indicates that my intuition was incorrect (Section 7.2.3). Summarising document clusters which have lower levels of per sentence propositional density does not significantly affect the per sentence propositional density of the output summaries. If the simplified document clusters had contained sentences that were significantly shorter than those in the original clusters, a greater effect may have been observed. Unfortunately, as noted in Section 7.2.3, there was no significant difference between the lengths of sentences in the two versions in this experiment.

In the MEAD$_{OB1}$ setting, of the 4417 sentences in the test set containing compound clauses or nominally bound relative clauses, OB1 transformed 1327 (30.04%). It failed to simplify three sentences containing clause coordinators and 2450 sentences containing subordinate clauses (not necessarily nominally bound) as they did not match any of the handcrafted rule activation patterns. In many cases, the subordinate clauses in these sentences are the obligatory arguments of clause complement verbs occurring in constructions that resemble complex$_{\overline{RF}}$ NPs, such as *Martin told the court that he was in fear of his life.* The remaining 640 sentences that were not simplified contain other left boundaries of subordinate clauses that are not immediately preceded by nominal words. The handcrafted rule activation patterns used by OB1 when simplifying Type 2 sentences (Section 5.2.2) are based on identifying sequences of nominal words immediately followed by clause boundaries. Just 10 of the 139 sentences extracted for use in the topic summaries by MEAD$_{OB1}$ were derived from simplified sentences (7.19%). Of these, all but one was derived from the first sentence of the source document.

Tables 7.2 and 7.3 present examples of summaries derived from the two sets of document clusters. In these tables, the *Gold Standard* columns contain one of the four human-produced reference-summaries produced for each of the topic clusters. The methodology used to generate gold standard summaries was not purely extractive. For this reason, sentences in the gold standards are marked with bullet points rather than numbers indicating the location of each sentence. The *MEAD* columns contain summaries produced from the original document clusters while the *MEAD*$_{OB1}$ columns contain summaries produced from clusters

of automatically simplified documents. In the $MEAD_{OB1}$ columns, sentences generated by the sentence simplification method are *italicised*.

Inspection of Tables 7.2 and 7.3 reveals that the summaries of simplified document clusters do contain larger numbers of shorter sentences than those of the original clusters, in line with my intuition. However, this is not sufficient to greatly improve the quality of the generated summaries. Regardless of the extent to which my intuition may hold, clause compounding and modification of NPs by relative clauses can both serve as methods compressing information from multiple sentences into a single sentence, facilitating the summarisation task.

While use of $\text{MEAD}_{OB1}$ and $\text{MEAD}^{CEV}_{OB1}$ does not harm the performance of multidocument summarisation, this result is a negative response to research question **RQ-5**, with respect to this task. The automatic sentence simplification method does not facilitate subsequent text processing in this context. However, two points should be noted about this evaluation. First, MEAD was used as a black box. It is possible that a MDS method which integrates the simplification process would achieve better performance. Second, ROUGE was used as the evaluation metric. Given that the gold standard summaries used in task 2 of DUC-2004 were produced using non-extractive summarisation methods, the Pyramid evaluation method (Nenkova and Passonneau, 2004) may be more effective at highlighting differences in performance between the MEAD and $\text{MEAD}_{OB1}$ systems. This is because the Pyramid method focuses on the concepts mentioned in summaries rather than on a strict matching of words and phrases at the surface level. In future evaluations, it will be interesting to apply this method. Finally,

Table 7.2: *Summaries of original and simplified versions of document cluster d30040t generated using MEAD*

| **Gold Standard** | **MEAD** | **MEAD$_{OB1}$** |
|---|---|---|
| • Israeli security officials delayed the take-off of two planes from Gaza International Airport, further straining the Mid-East peace process.<br>• Considered a milestone toward Palestinian autonomy, thousands cheered Gaza's opening in November.<br>• Though some equipment was still not installed, Palestinian planes as well as planes from Egypt and other nations landed on opening day and were met by Arafat.<br>• The airport's opening and the stipulation that Israel would control the airspace and monitor passengers were included in the U.S. brokered Wye River accord.<br>• Since Wye, however, Israel and the Palestinians have accused each other of failing to honor its provisions. | **(1)** Israel has threatened to close down the Palestinian-run Gaza airport over a security violation, an Israeli official said Tuesday, a move that could further undermine the already fragile peace process.<br>**(2)** Palestinian airport workers refused to allow Israeli security officials to check the identity of passengers who arrived on an Egyptian plane Sunday afternoon, most of whom worked for the Palestinian Authority, according to Civilian Airport Authority Director Nir Yarkoni.<br>**(3)** Israeli security officials delayed two planes from taking off from the Palestinian airport on Wednesday, the latest tensions in a rare area of Israeli-Palestinian cooperation.<br>**(4)** A plane bound for Amman, Jordan was delayed for two hours over an Israeli demand that Palestinian officials inspect the luggage of a Palestinian passenger, Israel radio reported. | **(1)** Israel has threatened to close down the Palestinian-run Gaza airport over a security violation, an Israeli official said Tuesday, a move that could further undermine the already fragile peace process.<br>**(3)** Israeli security officials delayed two planes from taking off from the Palestinian airport on Wednesday, the latest tensions in a rare area of Israeli-Palestinian cooperation.<br>**(4)** A plane bound for Amman, Jordan was delayed for two hours over an Israeli demand that Palestinian officials inspect the luggage of a Palestinian passenger, Israel radio reported.<br>**(5)** Israel retains security control over flights arriving at Gaza International Airport, and Palestinian security officials consult with their Israeli counterparts.<br>**(6)** *Earlier this week, Israel threatened to close down the Gaza airport when Palestinian airport workers refused to allow Israeli security officials to check the identity of passengers.*[8] |

[8] The original version of this sentence contained a relative clause modifying *passengers*.

Table 7.3: *Summaries of original and simplified versions of document cluster d30008t generated using MEAD*

| **Gold Standard** | **MEAD** | **MEAD$_{OB1}$** |
|---|---|---|
| • Philippine and Indonesian presidents may not attend upcoming APEC summit in Malaysia due to Mahathir's arrest of Anwar.<br>• Malaysian leaders discuss replacement for Anwar.<br>• Philippine ambassador is asked to explain his president's support for Anwar.<br>• Issues at upcoming APEC summit will include the Asian economic crisis and IMF.<br>• Taiwan's president pressured by China to send representative.<br>• Mahathir's economic and political moves will be issues at the summit.<br>• World financial officials advise reform; topic likely to dominate talks.<br>• US-ASEAN delegation to attend; likes Thai economic efforts.<br>• APEC leaders to taste local Malaysian food at luncheon after summit. | **(1)** Indonesian President B.J. Habibie finds attending a summit of Asia-Pacific leaders "difficult" because of his concerns about the arrest of Malaysia's former deputy prime minister, a Thai newspaper reported Sunday.<br>**(2)** The leaders of Malaysia's ruling party met Tuesday to discuss a replacement for ousted deputy prime minister Anwar Ibrahim, who faces trial next month in a case that will test the country's legal system.<br>**(3)** After an unusual, one-on-one chat Tuesday night, the Philippine and Indonesian presidents were considering staying away from an Asia-Pacific summit in Malaysia to protest the treatment of their jailed friend Anwar Ibrahim.<br>**(4)** The Philippine ambassador to Malaysia said Friday he was summoned to the Malaysian Foreign Ministry to explain his president's statements in support of dissident Anwar Ibrahim. | **(1)** Indonesian President B.J. Habibie finds attending a summit of Asia-Pacific leaders "difficult" because of his concerns about the arrest of Malaysia's former deputy prime minister, a Thai newspaper reported Sunday.<br>**(2)** *The leaders of Malaysia's ruling party met Tuesday to discuss a replacement for ousted deputy prime minister Anwar Ibrahim.*<br>**(3)** After an unusual, one-on-one chat Tuesday night, the Philippine and Indonesian presidents were considering staying away from an Asia-Pacific summit in Malaysia to protest the treatment of their jailed friend Anwar Ibrahim.<br>**(4)** The Philippine ambassador to Malaysia said Friday he was summoned to the Malaysian Foreign Ministry to explain his president's statements in support of dissident Anwar Ibrahim.<br>**(5)** On Monday, during the trial of Malaysia's former deputy prime minister, Anwar Ibrahim, on charges of corruption and sex-related crimes, Anwar angrily denied Mahathir's claim that he had approved the bailout. |

in the absence of additional test data, the hypothesis that sentence simplification improves the summarisation of text collections containing larger numbers of compound clauses and complex$_{\overline{RF}}$ NPs must remain untested.

## 7.3 Semantic Role Labelling

Semantic role labelling (SRL) is the task of automatically detecting the different arguments of predicates expressed in input sentences. I evaluated a system performing SRL in accordance with the PropBank formalism (Palmer *et al.*, 2005). In this scheme, an "individual verb's semantic arguments are numbered, beginning with zero. For a particular verb, [A0] is generally the argument exhibiting features of a Prototypical Agent (Dowty, 1991), while [A1] is a Prototypical Patient or Theme. No consistent generalizations can be made across verbs for the higher-numbered arguments". The scheme includes semantic roles for "general, adjunct-like arguments" providing information on the verb's cause (AMCAU), direction (AMDIR), discourse relations (AMDIS), location (AMLOC), manner (AMMNR), modal function[9] (AMMOD), negation (AMNEG), purpose (AMPNC), and time (AMTMP), among others. For extrinsic evaluation of the sentence simplification method which exploits handcrafted rule activation patterns to simplify Type 1 sentences, I focused on verbal predicates[10] and the twelve listed semantic roles.

An example of SRL to analyse a sentence such as (47)

[9] In the case of verbs.

[10] As opposed to prepositional, adjectival, or other types of predicate.

Table 7.4: *Semantic role labelling of Sentence (47)*

| **A0** | **V** | **A1** | **A2** | **A3** | **AMDIS** | **AMNEG** | **AMTMP** |
|---|---|---|---|---|---|---|---|
| Disney | offered | to pay Mr. Steinberg a premium for his shares | | | | | |
| Disney | pay | his shares | Mr. Steinberg | a premium | | | |
| the New York investor | demand | the company also pay a premium to other shareholders | | | | n't | When Disney offered to pay Mr. Steinberg a premium for his shares |
| the company | pay | | other shareholders | a premium | also | | |

(47) *When Disney offered to pay Mr. Steinberg a premium for his shares, the New York investor didn't demand the company also pay a premium to other shareholders.*

is provided in Table 7.4. The table contains a row of information about the semantic roles associated with each of the four main verbs occurring in the sentence. For example, it encodes information about the agent (*the New York investor*), patient or theme (*the company also pay a premium to other shareholders*), time (*When Disney offered to pay Mr. Steinberg a premium for his shares*), and negation (*n't*) of the verb *demand*.

In the SRL task, I extrinsically evaluated the variant of OB1 simplifying only Type 1 sentences. I made this decision on the basis of two observations:

1. The variant of OB1 which simplifies Type 2 sentences is relatively inaccurate ($0.306 \leq F_1 \leq 0.577$). It is therefore unlikely to make a positive

contribution to tasks that rely on a detailed analysis of sentence structure.

2. The extrinsic evaluation of the SRL task is based on human evaluation. The focus on accuracy when simplifying only Type 1 sentences would significantly reduce the amount of manual annotation required, improving the feasibility of the task.

### 7.3.1 Test Data (SRL)

For the task of evaluating a semantic role labelling system for the purpose of extrinsically evaluating my approach to sentence simplification, no suitable test data exist. Although annotated data from the CoNLL-2005 shared task on SRL is available,[11] this test data is available only for the original versions of input sentences and not for the simplified versions which may be generated by a given sentence simplification system. Given that it is difficult to map verbs, their arguments, and the semantic labels of these arguments from sentences in their original form to groups of sentences in their automatically generated simplifications, I evaluated the output of the SRL system manually.[12] I applied the SRL system to the original and automatically simplified versions of texts of the news register (one of the datasets presented in Table 6.1 of Chapter 6).

[11] http://www.lsi.upc.edu/~srlconll/soft.html. Last accessed 8th January 2020.

[12] Initially, I attempted to develop an automatic method to map verbs and their arguments in sentences from the original test data to verbs and arguments in their simplified correlates, but this was error prone and it became apparent that expensive manual alignment would be needed. Execution of this task would have been beyond the resources available in the current project.

### 7.3.2 Semantic Role Labelling System

I made the extrinsic evaluation of the sentence simplification method using Senna (Collobert *et al.*, 2011), a SRL system which tags predicates and their arguments in accordance with the formalism used in *PropBank*. I used Senna in two settings to label the semantic roles of verb arguments in Type 1 sentences of the news register in the test data described in Section 7.3.1. In the first setting, Senna processed sentences in their original form (*Senna*) and in the second ($Senna_{OB1}^{CEV}$), in the simplified form produced by the sentence simplification method applying the transformation schemes presented in Section 5.2.1.1.

### 7.3.3 Motivation (Sentence Simplification for SRL)

In Section 6.2, I described the use of six metrics to assess the readability of the original and simplified versions of texts which include those used as test data for the SRL task. I found that the automatically simplified news texts have a lower propositional density (0.483 vs. 0.505) and reading grade level (5.4 vs. 10.3) and greater syntactic simplicity (89.07 vs. 46.81) and temporal consistency, assessed in terms of tense and aspect (30.15 vs. 27.76) than the original news texts. As a task dependent on accurate syntactic parsing (including partial parsing), my intuition is that automatic SRL will be more accurate when processing the simplified versions of the input texts.

### 7.3.4 Evaluation Method (SRL)

I applied Senna to the original and automatically simplified versions of the test data. For cases in which the SRL performed by Senna differed when processing the original and automatically simplified versions of input sentences, I manually inspected the two analyses, and recorded the number of cases for which SRL of the original sentence was superior to that of the simplified sentence, and vice versa. This assessment was made primarily by reference to the set of PropBank Frames available online[13] and the PropBank annotation guidelines (Bonial *et al.*, 2015).

This evaluation method has the disadvantage that it does not provide measures of the absolute accuracy of *Senna* and $Senna_{OB1}^{CEV}$. When the two systems agree on the arguments identified and the semantic role labels assigned, these cases are not manually inspected, so the accuracy with which the arguments are identified and the labels are assigned is not known. However, the main purpose of this evaluation is to investigate *differences* in the accuracy scores of the two systems. The evaluation method that I use has the advantage of identifying such differences and only requiring manual evaluation of cases where the arguments identified by the two systems differ. This is a relatively small proportion of the total number of arguments identified by the two systems.

[13] http://verbs.colorado.edu/propbank/framesets-english-aliases/. Last accessed 8th January 2020.

### 7.3.5 Results (SRL)

Tables 7.5–7.7 contain examples of the semantic roles labelled in three different sentences in the test data that I used. In these tables, arguments identified more accurately in simplified sentences are underlined.

My manual evaluation of output from Senna revealed that 86.39% (1707) of the arguments identified using Senna in the two versions of the texts were identical. Of the remaining arguments, 5.31% (105) of those correctly identified by Senna in the original versions of the texts were not identified in the simplified versions while 8.29% (164) of the arguments correctly identified by Senna in the simplified versions of the texts were not identified in the original versions. Of the 269 arguments identified in only one of the versions of the texts, 60.97% were arguments identified more accurately in the simplified version, while 39.03% were arguments identified more accurately in the original version of the text.

Table 7.8 shows the number of semantic roles labelled more accurately, by type, when Senna processes the original (*Orig*) and the automatically simplified (*Simp*) versions of news articles. To illustrate, when processing the original versions of the news texts, Senna correctly identifies the agents (arguments with semantic role label A0) of 14 verbs that it did not identify when processing the automatically simplified versions of those texts. Conversely, when processing the automatically simplified versions, Senna identified the agents of 23 verbs that it did not identify when processing the original versions.

These evaluations of *Senna* and $Senna_{OB1}^{CEV}$ show that the best accuracy is

Table 7.5: *Example 1 of more accurate semantic role labelling in automatically simplified text.*

| **Original Sentence** | | | | | |
|---|---|---|---|---|---|
| It's up to the prosecution to show the defendant was not acting in self defence and they have failed to do so. | | | | | |
| **A0** | **V** | **A1** | **A2** | **AMMNR** | **AMNEG** |
| | show | the defendant was not acting in self defence and they have failed to do so | | | |
| the defendant | acting | | | in self defence | not |
| | failed | they | to do so | | |
| they | do | so | | | |
| **Simplified Sentence** | | | | | |
| It's up to the prosecution to show the defendant was not acting in self defence. They have failed to do so. | | | | | |
| **A0** | **V** | **A1** | **A2** | **AMMNR** | **AMNEG** |
| | show | the defendant was not acting in self defence | | | |
| the defendant | acting | | | in self defence | not |
| | failed | They | to do so | | |
| They | do | so | | | |

Table 7.6: *Example 2 of more accurate semantic role labelling in automatically simplified text.*

**Original Sentence**

They were living apart but Conway had agreed to babysit for their children Daniel, 12, and Laura, 11.

| **A0** | **V** | **A1** | **AMMNR** | **AMTMP** |
|---|---|---|---|---|
| They | living | | | |
| Conway | agreed | to babysit for their children Daniel, 12, and Laura, 11 | | |
| Conway | babysit | | | for their children Daniel, 12, and Laura, 11 |

**Simplified Sentence**

They were living apart. Conway had agreed to babysit for their children Daniel, 12, and Laura, 11.

| **A0** | **V** | **A1** | **AMMNR** | **AMTMP** |
|---|---|---|---|---|
| They | living | | apart | |
| Conway | agreed | to babysit for their children Daniel, 12, and Laura, 11 | | |
| Conway | babysit | for their children Daniel, 12, and Laura, 11 | | |

Table 7.7: *Example 3 of more accurate semantic role labelling in automatically simplified text.*

| | | | | | | |
|---|---|---|---|---|---|---|
| **Original Sentence** | | | | | | |
| But Smith had already been arrested - her clothing had been found near his home and DNA tests linked him to it. | | | | | | |
| **A0** | **V** | **A1** | **A2** | **AMDIS** | **AMLOC** | **AMTMP** |
| | arrested | Smith | | But | | already |
| | found | her clothing | | | near his home and DNA tests linked him to it | |
| his home and DNA tests | linked | him | to it | | | |
| **Simplified Sentence** | | | | | | |
| But Smith has already been arrested - her clothing had been found near his home. DNA tests linked him to it. | | | | | | |
| **A0** | **V** | **A1** | **A2** | **AMDIS** | **AMLOC** | **AMTMP** |
| | arrested | Smith | | But | | already |
| | found | her clothing | | | near his home | |
| DNA tests | linked | him | to it | | | |

Table 7.8: *Positive differences in numbers of true positives obtained for semantic role labelling of original and simplified versions of input texts*

| **Role** | **Orig vs. Simp** | **Simp vs. Orig** |
|---|---|---|
| A0 (agent) | 14 | 23 |
| A1 (patient/theme) | 45 | 77 |
| A2 (argument less prominent than A1) | 14 | 13 |
| AMCAU (cause) | 0 | 1 |
| AMDIR (direction) | 4 | 0 |
| AMDIS (discourse relation) | 0 | 3 |
| AMLOC (location) | 3 | 13 |
| AMMNR (manner) | 4 | 6 |
| AMNEG (negation) | 0 | 1 |
| AMPNC (purpose) | 1 | 6 |
| AMTMP (time) | 12 | 27 |
| V (verb) | 2 | 3 |
| **Total** | **99** | **173** |

```
MEDLEY: MAC3332
A 10-day-old newborn is brought to the emergency department because of
  generalized tonic-clonic seizures. He has had increasing lethargy over
  the past 2 days. He has pallor, tachycardia, and tachypnea; pulses are
  weak.
Laboratory studies show:
  Hematocrit 35%
Serum
  Na+ 112 mEq/L
  K+ 6.5 mEq/L
Arterial blood gas analysis on room air shows:
  pH 7.23
  Pco2 25 mm Hg
  Po2 87 mm Hg
A disorder involving which of the following organs is most likely
 responsible for this patient's symptoms?
```

Figure 7.1: *A clinical vignette*

obtained by the latter, although there are a minority of cases where the arguments of verbs are labelled more accurately in the original input sentences than in the simplified ones. However, overall, it can be concluded that the automatic sentence simplification method does facilitate the text processing task of semantic role labelling. This is a positive response to research question **RQ-5**.

## 7.4 Information Extraction

Information extraction (IE) is the automatic identification of selected types of entities, relations, or events in free text (Grishman, 2005). This section of the chapter concerns IE from vignettes which provide brief clinical descriptions of hypothetical patients for the purpose of assessing educational attainment in medical licensure. Figure 7.1 is an example of a clinical vignette.

The discourse structure of these vignettes consists of six elements:

1. basic information - patient's gender, profession, ethnicity, and health status;

2. chief complaint - the main concern motivating the patient to seek therapeutic intervention;

3. history - a narrative description of the patient's social, family, and medical history;

4. vital signs - a description of the patient's pulse and respiration rates, blood pressure, and temperature;

5. physical examination - a narrative description of clinical findings observed in the patient;

6. diagnostic study and laboratory study - the results of several different types of clinical test carried out on the patient.

In the IE task, each element in the discourse structure is represented by a template encoding related information. For example, the template for physical examinations holds information on each clinical finding/symptom (FINDING) observed in the examination, information on the technique used to elicit that finding (TECHNIQUE), the bodily location to which the technique was applied (LOCATION), the body system that the finding pertains to (SYSTEM), and any qualifying information about the finding (QUALIFIER). In this chapter, I focus on automatic extraction of information pertaining to physical examinations. The

goal of the IE system is to identify the phrases used in the clinical vignette that denote findings and related concepts and add them to its database entry for the vignette.

As in the extrinsic evaluation via SRL (Section 7.3), in the IE task, I extrinsically evaluated the variant of OB1 simplifying only Type 1 sentences. There are two reasons for this:

1. As noted in Section 7.3, the variant of OB1 which simplifies Type 2 sentences is relatively inaccurate ($0.306 \leq F_1 \leq 0.577$). It is therefore unlikely to make a positive contribution to tasks that rely on a detailed analysis of sentence structure.

2. Type 2 sentences are infrequent in the types of text processed in the clinical information extraction task.

For extrinsic evaluation of the sentence simplification method based on handcrafted rule activation patterns and simplifying Type 1 sentences (Section 5.2.1.1), the sentences in the test data were simplified using that method. I then ran the IE system in two settings. In the first ($\text{IE}_{ORIG}$), it processed the original collection of vignettes. In the second ($\text{IE}_{OB1}^{CEV}$), it processed the automatically simplified vignettes which are expected to contain a reduced number of compound clauses.

In Section 7.4.5, I comment briefly on results obtained when using other versions of the sentence simplification method: one simplifying Type 2 sentences and one simplifying both Type 1 and Type 2 sentences.

### 7.4.1 Test Data (IE)

For the experiment described in this chapter, my test data comprises a set of 286 clinical vignettes paired with manually completed IE templates, encoding information about TECHNIQUEs, LOCATIONs, SYSTEMs, and QUALIFIERs, associated with the 719 FINDINGs that they contain. This test data was developed in the context of an earlier project (Evans, 2011) and is based on clinical vignettes owned by the National Board of Medical Examiners.[14] The test data contains 28 425 words and 3286 sentences.[15] The completed IE templates were developed by Dr. Le An Ha at the University of Wolverhampton in 2008.

### 7.4.2 Information Extraction System

In this extrinsic evaluation experiment, I used a reduced version of the IE system described by Evans (2011) to identify sets of facts in clinical *vignettes*. The main differences between Evans's (2011) system and the one used in the current extrinsic evaluation is that his system exploits a slightly larger set of more specific information extraction patterns. Like the one presented in this section, Evans's (2011) IE system was designed for use in combination with a sentence simplification tool. This earlier method for sentence simplification was based on the detection of potential coordinators, which comprise a subset of the signs of syntactic complexity presented in Chapter 2. Other points of distinction are that Evans's (2011) sentence simplification method implements transformation schemes to simplify sentences containing subclausal compounds and domain-specific compound phrases

[14] https://www.nbme.org/. Last accessed 8th January 2020.
[15] This sentence count includes individual rows of tables presenting laboratory studies.

(including compound structures referring to obstetric/gynecological findings and patient medical histories) and includes classes of coordinators which link domain specific compound phrases. The earlier method implements no transformation schemes to simplify sentences containing complex$_{\overline{RF}}$ NPs (Type 2 sentences).

For the experiments described in this chapter, I used a reduced version of Evans's (2011) IE method which processes vignettes using the same tokenisation system as the sentence simplification module described in Section 5.2 of this thesis to identify sentences. For the purpose of IE but not sentence simplification, references to medical concepts were tagged on the basis of gazetteers developed in my previous work (Table 7.9) and a simple set of finite state transducers to group adjacent references to concepts (Table 7.10). As a result, the IE system can be applied to other texts of the same domain. It was not purpose-built to evaluate or facilitate the sentence simplification method, making it similar to the tools used in the other extrinsic evaluations (Sections 7.2 and 7.3).

After tagging references to clinical concepts in the vignettes, IE is performed using a small number of simple rules. To summarise briefly, vignettes are processed by considering each sentence in turn. Every mention of a clinical FINDING or SYMPTOM is taken as the basis for a new IE template. The first tagged TECHNIQUE, SYSTEM, and LOCATION within the sentence containing the SYMPTOM or FINDING is considered to be related to it.[16] QUALIFIERS (e.g. *bilateral* or *peripheral*) are extracted in the same way, except in sentences containing the word

[16]Versions of the system in which the closest tagged concept was extracted in each case, rather than the first, were significantly less accurate in both cases (overall accuracy of 0.6542 for IE$_{ORIG}$ and 0.6567 for IE$^{CEV}_{OB1}$).

Table 7.9: *Selected samples of the gazetteers used for concept tagging*

| Findings | Symptoms | Techniques |
|---|---|---|
| abnormality | ascite | a monofilament |
| convex appearance | dehydration | consciousness |
| dehydrated | dyspnea | CT scan |
| febrile | kyphosis | esophagogastroscopy |
| lethargic | motor deficit | percussion |
| otherwise normal | nocturia | plasma renin activity |
| pharyngitis | obesity | sensation decreased to pinprick |
| rhonchi | phobic | skin turgor |
| Tanner stage 2 | rubs | term memory |
| within normal limits | wheeze | x-rays |
| **Systems** | **Locations** | **Qualifiers** |
| adrenal gland | ankle | arteriolar |
| aortic | conjunctiva | bulbous |
| breath | dorsum | diffuse |
| L4 | extremities | fine |
| metatarsal | face | intact |
| obstetrical | forearm | metastatic |
| optic fundi | index finger | patchy |
| papillary | intercostal | prominent |
| phalanges | perineal | routine |
| tissue | sclera | superior |

Table 7.10: *Finite state transduction patterns to group adjacent concept tags*

| **Tag Sequence** | **Tag** | **Example** |
|---|---|---|
| {SYMPTOM\|FINDING} in {SYMPTOM\|FINDING} | {SYMPTOM\|FINDING} | delay in emptying |
| {A\|An} TAG TAG TECHNIQUE | TECHNIQUE | a gastric emptying scan |
| QUALIFIER QUALIFIER {SYSTEM\|LOCATION} of the SYSTEM | SYSTEM | left upper lobe of the thyroid gland |
| SYSTEM TECHNIQUE | TECHNIQUE | funduscopic examination |
| QUALIFIER TECHNIQUE | TECHNIQUE | physical examination |
| FINDING SYMPTOM | FINDING | nontender enlargement |
| QUALIFIER LOCATION | LOCATION | left supraclavicular |

*no*. In these cases, the QUALIFIER related to the FINDING is identified as *none*. Due to their scarcity in the development corpus, this rule was not extended to additional negative markers such as *never* or *not*.

### 7.4.3 Motivation (Sentence Simplification for IE)

An analysis of the readability of the original and simplified versions of the clinical vignettes did not provide a strong indication that the automatic sentence simplification method would improve the accuracy of the IE system. The 286 original clinical vignettes in the test data have a mean propositional density of 0.4826 ideas per word and 5.499 ideas per sentence. The values of these metrics for the simplified versions of the vignettes are 0.4803 ideas per word and 5.269 ideas per

sentence, respectively. Although they are of the correct polarity, these differences are not statistically significant ($p = 0.5327$ and $p = 0.1407$, respectively). However, previous work in sentence simplification for IE (Evans, 2011; Niklaus *et al.*, 2016) has demonstrated that automatic sentence simplification can improve the accuracy of IE systems. This provided motivation to evaluate the impact of the automatic sentence simplification method in this task.

### 7.4.4 Evaluation Method (IE)

For the IE task, my evaluation metric is based on $F_1$-score averaged over all slots in the IE templates and all templates in the test data. Identification of true positives is based on exact matching of system-identified slot fillers with those in the manually completed IE templates in the test data.

Chinchor (1992) notes that assessment of the statistical significance of differences in accuracy between different IE systems is challenging. In my evaluation, I used a bootstrapping method to obtain a more informative picture of the difference in performance between different versions of the IE system. In related work focused on my evaluation experiment, dos Santos *et al.* (2018) framed the comparison between two IE systems using a binomial regression model. Given that such models apply only when the variables being considered are independent, dos Santos *et al.* (2018) included a latent variable in the analysis to represent the effect of the text on the performance of the two systems (the two evaluations are not independent because both systems process the same text). Odds ratio was then used to show the probability of agreement between each IE system and the

Table 7.11: *Accuracy of information extraction when applying the OB1 system as a preprocessing step*

| Template slot | $\mathbf{IE}_{ORIG}$ | | $\mathbf{IE}_{OB1}^{CEV}$ | | |
|---|---|---|---|---|---|
| | **Acc** | **95% CI** | **Acc** | **95% CI** | **Best Performer** |
| FINDING | 0.8819 | [0.847, 0.914] | 0.8861 | [0.853, 0.917] | 0.5486 |
| TECHNIQUE | 0.8514 | [0.814, 0.886] | 0.8903 | [0.858, 0.922] | 0.9344 |
| SYSTEM | 0.8097 | [0.769, 0.850] | 0.8431 | [0.806, 0.881] | 0.873 |
| QUALIFIER | 0.7431 | [0.697, 0.786] | 0.7708 | [0.728, 0.814] | 0.794 |
| LOCATION | 0.8431 | [0.806, 0.881] | 0.8611 | [0.825, 0.894] | 0.735 |
| All | 0.8258 | [0.808, 0.843] | 0.8503 | [0.834, 0.867] | 0.976 |

gold standard.

### 7.4.5 Results

The accuracy scores obtained by each variant of the IE system are presented in Table 7.11. Inspection of this table indicates that while there are only marginal improvements in the accuracy with which FINDINGs are identified in the simplified versions of the input texts, related concepts tend to be identified more accurately. This is especially true of QUALIFIERs and TECHNIQUEs.

An example of the difference in performance of $\text{IE}_{ORIG}$ and $\text{IE}_{OB1}^{CEV}$ is provided for sentence (48). In these examples, identified FINDINGs are italicised and associated concepts are underlined. Multiword terms appear in square brackets. Here, the FINDING *obesity* is not tagged correctly in simplified sentence (49-a) because the SYMPTOM *striae* is erroneously grouped with *obesity* to form a new FINDING, *obesity striae* which does not match the FINDING listed in the gold standard. In future work, errors of this type could be addressed by including detection of specialised terminology in the tokenisation process used by the sentence

simplification method.

(48) She has truncal$_{LOC}$ *obesity* and pigmented$_{QUAL}$ abdominal$_{LOC}$ *striae*.

(49) a. She has truncal$_{LOC}$ [*obesity striae*].

b. She has pigmented$_{QUAL}$ abdominal$_{LOC}$ *striae*.

By contrast, LOCATIONS in the automatically generated simplification (49) are identified with greater accuracy than those in (48) because IE$_{ORIG}$ erroneously extracts the same LOCATION (*truncal*) for both FINDINGs.

I applied a bootstrapping method to obtain confidence intervals for the accuracy of extraction of each of the IE template slots. For this purpose, 50% of the output of each system was randomly sampled in each of 100 000 evaluations. The confidence intervals are presented in the *95% CI* columns of Table 7.11. The figures in the *Best Performer* column of this table indicate the proportion of evaluations for which the $IE_{OB1}^{CEV}$ system was more accurate than the $IE_{ORIG}$ system. Differences in the accuracy of information extraction were found to be statistically significant in all cases, using McNemar's test ($p < 0.00078$), with the exception of differences when extracting FINDINGs ($p = 0.6766$).

With reference to my extrinsic evaluation data and using a binomial regression model, dos Santos *et al.* (2018) show that the odds ratio of agreement between IE$_{OB1}^{CEV}$ and the gold standard is 1.5 times greater than that between IE$_{ORIG}$ and the gold standard. For all slots in the information extraction template, the probability of agreement between IE$_{ORIG}$ and the gold standard is 0.937. The

probability of agreement between $\text{IE}_{OB1}^{CEV}$ and the gold standard is 0.957. From this, they conclude that $\text{IE}_{ORIG}$ and $\text{IE}_{OB1}^{CEV}$ differ in their performance on the information extraction task. The probability of agreement with the gold standard is greater for $\text{IE}_{OB1}^{CEV}$ than for $\text{IE}_{ORIG}$, although the probability of agreement is already large for $\text{IE}_{ORIG}$. This evaluation indicates that research question **RQ-5** can be answered positively.

I did additional experiments using different variants of the sentence simplification method in the preprocessing step. A version simplifying only Type 2 sentences (containing $\text{complex}_{\overline{RF}}$ NPs) was found to have the same performance as $\text{IE}_{ORIG}$. A preprocessing step in which both Type 1 and Type 2 sentences were simplified performed no better than $\text{IE}_{OB1}^{CEV}$. This is to be expected because the clinical vignettes contain very few $\text{complex}_{\overline{RF}}$ NPs.

## 7.5 Contribution to Research Question RQ-5

Evaluation of the three NLP applications in each of the experiments presented in this chapter addresses research question **RQ-5**:

> *Does the automatic sentence simplification method facilitate subsequent text processing?*

I evaluated **RQ-5** with respect to the simplification of Type 1 sentences applied in three text processing tasks: multidocument summarisation, semantic role labelling, and clinical information extraction. In two of these tasks, multidocument summarisation and clinical information extraction, I extrinsically evaluated

the simplification of Type 2 sentences and the simplification of both sentence types together.

In the multidocument summarisation task, regardless of the type of simplification performed, use of my automatic methods in a preprocessing step evoked no change in accuracy. While the methods did not adversely affect the performance of the multidocument summarisation system, these evaluations were negative responses to research question **RQ-5**.

In the SRL task, an implementation of my approach to automatically simplify Type 1 sentences exploiting handcrafted rule activation patterns made a positive contribution. When integrated as a preprocessing step, the sentence simplification tool enabled the Senna SRL system to correctly identify and label larger numbers of arguments of verbal predicates. This evaluation was a positive response to **RQ-5**.

In the clinical information extraction task, the implementation of my approach to automatically simplify Type 1 sentences exploiting handcrafted rule activation patterns made a positive contribution. When integrated as a preprocessing step, the sentence simplification tool enabled a simple IE system to correctly identify four out of five template slot fillers with greater accuracy than the same system processing the original unsimplified texts. This evaluation was a second positive response to **RQ-5**.

Two of the three extrinsic evaluations presented in this thesis showed that the sentence simplification methods facilitate subsequent text processing. The NLP applications used for this purpose were black boxes. Tools designed to

better exploit the specific transformations applied by the sentence simplification systems have the potential to obtain greater improvements in accuracy.

# Chapter 8

# Conclusions

This thesis has been concerned with the development of an NLP pipeline for automatic sentence simplification using shallow methods for syntactic analysis. The aim of the method is to process input text and generate output with reduced per sentence propositional density and syntactic complexity. In previous work, these factors have been noted to correlate with both text comprehensibility for human readers and the accuracy with which texts can be processed using NLP applications. In this chapter, I conclude the thesis by summarising findings from my responses to each of the five research questions addressed and describing potential impacts which may arise from these responses. I include directions for future research which may enhance subsequent responses to the five questions.

## 8.1 Research Question RQ-1

In my response to **RQ-1**, I developed a new annotated corpus encoding information about syntactic complexity (coordination and subordination) in English texts (Chapter 2). The annotation scheme used has two features that make it inexpensive to apply and thus enabled rapid development of the corpus. First, only a limited number of signs of syntactic complexity are considered markable, and

these can be automatically detected with great reliability by the annotation tool developed in this research. Annotators are not required to encode a complete syntactic analysis of the text. Instead, markables are tagged to indicate their syntactic functions as either coordinators or subordinate clause boundaries and to indicate the syntactic category and projection level of the constituents linked by the former or bounded by the latter. Second, the annotation is not dependent on other, potentially expensive annotation layers, as is the case for that presented by Maier *et al.* (2012). The scheme is thus portable and can be used to derive new language resources from unrestricted English text.

I showed that the resources produced using the annotation scheme are annotated with high levels of reliability and consistency (Chapter 2, Section 2.2.3). Assessments of inter-annotator consistency in three text registers imply that, with appropriate guidelines, it is possible to annotate texts from specialised domains with little degradation in reliability.

Evaluation of the NLP tools presented in Chapters 3–5 demonstrates that the development of a pipeline for automatic sentence simplification benefits from the availability of the annotated resources presented in this chapter. Insights gained from the empirical analysis of these resources can help subsequent researchers to prioritise the accurate processing of the most common types of syntactic complexity occurring in texts of different registers. The corpus analysis described in Chapter 2 provides information on the functions and frequency distributions of explicit textual signs indicating the occurrence of compound constituents and subordinate clauses in English sentences.

Analysis of the frequency distribution of class labels assigned by the human annotators reveals that in the majority of cases, the signs of syntactic complexity function as coordinators in compound constituents and as boundaries of subordinate clauses of different types. 74.11% of the signs are either coordinators in compound clauses or boundaries of subordinate clauses which can modify complex NPs. Analysis of syntactically annotated sentences in the Penn Treebank (Marcus *et al.*, 1993) shows that a large proportion (56.74%) of intrasentential clause conjoins and subordinate clauses are adjacent to signs of syntactic complexity specified in Chapter 2 of the thesis. These findings are of direct relevance to research question **RQ-1**.

Exploration of research question **RQ-1** brings several original contributions. The definition and annotation of a restricted set of signs of syntactic complexity with information about their syntactic linking and bounding functions led to the creation of a new resource that can be exploited in the subsequent development of syntactic analysis tools. The annotation encodes information that is absent from most existing syntactic treebanks and syntactically annotated resources. Although small in size, the set of signs is significantly larger than that annotated in previous related work. The corpus developed comprises texts of three registers (news, public healthcare information, and literature). The frequency distribution of signs and class labels provides insights into the nature of syntactic complexity in each one. As a result, the annotated corpus adds knowledge about the distribution of these textual markers of syntactic complexity and their functions. For example, this provides information enabling researchers to estimate the probability that a

given occurrence of the conjunction *and* coordinates prepositional phrases in texts of the news register. This enhances our knowledge of the English language.

With regard to the annotation of information about signs of syntactic complexity, there are two main directions for future work. The first concerns expansion of the annotation scheme. Currently, although the annotation scheme discriminates between signs bounding clauses in which all elements of the clause are extant (classes ES/SSEV) and those in which various elements have been elided (e.g. classes ES/SSMP, ES/SSMN, and ES/SSMA), class labels SSEV and ESEV still subsume several potential subclasses. This is due to the variety of types of clause that may be subordinated, which include independent clauses, nominal *that*-clauses, adverbial *when*-clauses, and some verbless clauses. Further, as noted in Section 4.2.3 (pages 86–88), subordinate clauses may have a range of functions in natural language. As a result, SSEV is by far the most frequent class label occurring in the annotated corpus. It is possible that the inclusion of such a wide-ranging class introduces unnecessary ambiguity in the automatic classification of signs of syntactic complexity. In future work, it will be interesting to investigate the effect of extending the set of class labels to include different tags for different types of finite clause.

While encoding a wide range of phenomena related to syntactic complexity, it may be argued that additional signs should be included in the annotation scheme used in this thesis. Though not reported here, the annotation of parentheses has been included in the most recent version of the scheme: they frequently bound subordinate constituents, especially in documents providing patient healthcare

information, where they bound subordinate noun phrases in the great majority of cases, as well as verb phrases and clauses. Previous work has described a wide range of functions of hyphens or dashes (Nunberg *et al.*, 2002), which is another candidate for inclusion in the set of signs of syntactic complexity to be annotated. The scheme may also be expanded to include a wider variety of conjunctions and other markers of syntactic complexity.

The second possible direction of future work involves extrinsic evaluation of the resources developed via exploitation by NLP applications. Following Maier *et al.* (2012), the automatic classification of signs of syntactic complexity (Chapter 3) would enable the addition of a second annotation layer to resources such as the Penn Treebank, which could then be exploited by supervised parsing methods. Chapter 7 of this thesis presents indirect evaluation of these resources via extrinsic evaluation of the sentence simplification method that exploits them.

## 8.2 Research Question RQ-2

As part of my response to research question **RQ-2**, Dornescu *et al.* (2013) developed an NLP tool to automatically tag signs of syntactic complexity in accordance with the annotation scheme presented in my response to **RQ-1**. This scheme specifies the syntactic linking and bounding functions of these signs. This sign tagger was presented in Chapter 3 of the thesis. Its parameters were set optimally in a performance-driven development process. Evaluation of the sign tagger indicates that it achieves acceptable levels of accuracy. It made a useful contribution to the NLP pipeline for sentence simplification and the component tools presented in

Chapters 4–5. I envisage that automatic sign tagging has the potential to make a contribution to other NLP tasks such as full syntactic parsing and alternate approaches to sentence simplification.

Development and evaluation of the automatic sign tagger is an original contribution. Although similar taggers have been developed in previous related work, they classify very limited sets of markers of syntactic complexity with respect to a less detailed taxonomy. Furthermore, those methods were not evaluated extrinsically with regard to both a sentence simplification system and via extrinsic evaluation of that sentence simplification system. Implementation of the sign tagger in joint work (Dornescu *et al.*, 2013) and the availability of an online demo of this tagger[1] are further contributions of my response to **RQ-2**.

The *raison d'etre* for the sign tagger is to implement a shallow method for sentence analysis which overcomes the practical difficulties in syntactically parsing long complex sentences. Where feasible, analyses provided by full parsing of input sentences will be preferable to shallow syntactic analysis for the task of sentence simplification. For this reason, in future work, it will be interesting to explore the development of a hybrid approach for sentence simplification in which the sign tagger is used as a last resort for the analysis of extremely long sentences and a syntactic parser is used for the analysis of shorter sentences whose parsing is computationally more feasible.

[1] http://rgcl.wlv.ac.uk/demos/SignTaggerWebDemo/. Last accessed 8th January 2020.

## 8.3 Research Question RQ-3

My response to **RQ-3** consisted of the development and evaluation of a new iterative approach to sentence simplification exploiting shallow syntactic analysis and a small set of sentence transformation schemes (Chapter 5). The syntactic analysis step was implemented using the automatic sign tagger presented in Chapter 3, while the sentence transformation schemes were implemented using handcrafted rule activation patterns (Section 5.2.2). Evaluation results indicate that the current version of the sentence simplification method is most effective when processing sentences containing compound clauses in texts of the registers of *news* and *health*. However, performance is relatively poor when processing sentences containing bound relative clauses and when performing any kind of sentence simplification in texts of the *literary* register.[2]

Exploration of research question **RQ-3** brought new findings and original contributions in four main areas. With regard to the overall approach, use of the sign tagger rather than a full syntactic parser for the purpose of sentence analysis improves the computational tractability of the method. The analysis phase has linear complexity, circumventing one of the major weaknesses of approaches based on syntactic parsing. Comparison of the performance of my system (OB1) with that of a method exploiting the Stanford parser (MUSST) demonstrates that OB1 compares favorably with that method. The iterative approach and the level of detail brought in the analysis step makes it possible to easily customise and extend

[2]In NLP, the processing of literary texts is always challenging, so this finding is not surprising.

the system to simplify a wider range of complex and compound constituents in future work. The transformation schemes used by the simplification algorithm formalise simplification operations which apply to multiple types of compound clauses and complex$_{\overline{RF}}$ NPs, including those with subject relativised clauses and object relativised clauses in which the NP is object of a preposition or verb. The implemented system performs a wider range of syntactic simplification operations and performs more detailed syntactic analysis than the system implemented in my previous work (Evans, 2011).

To address **RQ-3**, I evaluated four sentence simplification systems:[3]

**OB1** My new sentence simplification method based on shallow syntactic analysis using automatic sign tagging and handcrafted rule activation patterns,

**Bsln** A baseline system exploiting the same sentence transformation schemes and handcrafted rule activation patterns as OB1 but using simple heuristics in the syntactic analysis step,

**MUSST** A restricted version of the system developed by Scarton *et al.* (2017) which is based on full syntactic parsing,[4]

**Orcl** A baseline system exploiting the same sentence transformation schemes and handcrafted rule activation patterns as OB1 but using an oracle to obtain information about the syntactic linking and bounding functions of signs of syntactic complexity.

[3] I exclude the STARS system, evaluated in my response to **RQ-4** from the current discussion.

[4] The restrictions were introduced to enable fair evaluation of all systems in their simplification of Type 1 and Type 2 sentences, which are the focus of my thesis.

This evaluation showed that OB1 compares favourably with *Bsln* and *MUSST* with respect to the closeness of its output to human produced simplifications and the readability of its output.

Evaluation of OB1 led to several novel findings. Part of this evaluation involved the use of overlap metrics such as SARI and a new metric based on Levenshtein distance to assess the similarity between simplified sentences generated by the system and simplified sentences generated by human editors. This aspect of the evaluation showed that the method is successfully able to simplify Type 1 and Type 2 sentences to a moderate and limited extent, respectively. The approach was found to be effective when processing texts of several different registers and domains, though the simplification of literary texts and sentences containing $\text{complex}_{\overline{RF}}$ NPs was noted to be relatively unreliable. Evaluation of the sign tagger demonstrated its suitability for use in the sentence analysis step.

My use of the SARI evaluation metric indicated few statistically significant differences in the accuracy of the OB1 and *Bsln* systems when simplifying Type 2 sentences. A statistically significant difference in performance was only evident for sentences of the health register, where OB1 was superior ($p = 0.036$). By contrast, differences between the accuracy scores obtained by OB1 and *MUSST* are statistically significant, in favour of OB1, when simplifying Type 2 sentences in texts of all registers ($p \ll 0.01$).

Evaluation of the iterative rule-based approach to sentence transformation revealed that the rules used to simplify Type 1 sentences are considerably more accurate than those used to simplify Type 2 sentences. This may be due to the

fact that in the latter task it is necessary both to identify the spans of those clauses and to discriminate between free and bound relative clauses. There is considerable room for improvement of the sentence transformation process.

Finally, my response to **RQ-3** reveals that the performance of the sentence simplification method is acceptable. After simplifying Type 1 sentences, its output contains just over 55% of the number of compound clauses occurring in the input text. However, after simplifying Type 2 sentences, its output still contains more than 90% of the subordinate clauses present in the original text.[5] This latter finding is partially explained by the fact that the subordinate clauses in an input text may have a range of functions, with just over 25% modifying complex$_{\overline{RF}}$ NPs. As a result, they may not be of a type that can be simplified using the sentence transformation schemes proposed in this thesis without incurring large reductions in the comprehensibility of the text (See Section 4.2.4, Table 4.5).

Section 6.2 of the thesis presented a second type of evaluation involving the use of text readability measures to estimate the accessibility of sentences produced by the system. In line with expectation, texts produced by the OB1 system obtained larger values for the *syntactic simplicity* metric than the original texts did. However, values of the *referential cohesion* and *deep cohesion* metrics were smaller for texts produced by OB1 than they were for the original versions of the texts when simplifying Type 1 sentences. Interestingly, for texts of most registers and for simplification of both Type 1 and Type 2 sentences, the simplification method generates output that is more consistent in terms of verb tense than the

[5] 77% of the non-restrictive nominally bound finite subordinate clauses.

original text, though not in the collection of texts conveying public healthcare information.

Use of the *deep cohesion* metric to assess the readability of system output indicates that use of OB1 leads to some loss of discourse relations expressed in the original texts. When simplifying Type 1 sentences, one possible way to address this issue would be to post-edit simplified sentences derived from clauses following a conjunction such as *but* in the input. Insertion of a sentence-initial "canned" adverb (e.g. *however*) could be used to re-establish the discourse relation. I plan, in future work, to conduct a more detailed analysis of the impact of sentence simplification operations on the discourse structure of output texts.

In future work, more sophisticated NLP-based approaches to the automatic estimation of readability could also be applied, such as those developed by Si and Callan (2001) and Schwarm and Ostendorf (2005) which, respectively, exploit language models and use support vector machines exploiting linguistic features to assess the readability of input texts.

Section 6.3 of the thesis reported on a human-centred evaluation which revealed that participants found the output of the OB1 system to be acceptable. This was determined on the basis of participants' responses to opinion surveys focused on the grammaticality, accessibility, and meanings of sentences output by OB1. Subsequent human-centred evaluations would be improved through the recruitment of larger numbers of participants in the opinion surveys and the use of more objective psycholinguistic evaluation methods such as eye tracking, self-paced reading, rapid serial visual presentation tasks, or reading comprehension

testing.

## 8.4 Research Question RQ-4

In my response to **RQ-4**, I developed and evaluated a new approach for sentence simplification based on the iterative application of sentence transformation schemes implemented as rules with machine learned activation patterns (Sections 5.2.1.1, 5.2.1.2, and 5.2.3). Derivation of these patterns was achieved through the development of a shallow syntactic analysis tool which uses a machine learning (sequence tagging) approach to identify the spans of compound clauses and complex$_{\overline{RF}}$ NPs in input sentences (Chapter 4). The advantage of this approach is that sentence simplification tools can be developed without requiring the onerous handcrafting of activation patterns associated with sentence transformation rules. The transformation rules used in these tools will better exploit information observed in comparatively large sets of annotated data, which is less expensive to produce, making them more generalisable. The automatic derivation of rule activation patterns for use in sentence simplification is likely to make those systems easier to tune for texts of different registers and from different domains. I refer to my system exploiting machine-learned rule activation patterns as STARS.[6]

The implementation of STARS was based on the development and evaluation of automatic sentence analysis tools to identify the spans of compound clauses and several types of complex constituents in input sentences. Evaluation of these analysis components (Section 4.4) indicated that a method simplifying Type 1

[6]Originally from a Sequence Tagging Approach to Rewrite Sentences.

sentences in any text register can benefit from the tagger of compound clauses while simplification of Type 2 sentences in the register of news may benefit from the tagger of complex constituents.

In Chapter 5, I describe the development of a sentence simplification method which exploits the taggers of compound clauses and complex constituents presented in Chapter 4. Evaluation of the sentence simplification algorithms exploiting automatic tagging of compound clauses and complex constituents is presented in Section 6.1.

Development of the automatic taggers (Chapter 4) and the sentence simplification method which integrates them (Section 5.2) constitutes the first part of my response to **RQ-4**. Evaluation of the integrated sentence simplification method, STARS, completes this response (Chapter 6).

When considered over all text registers, the difference in $F_1$-scores obtained by the OB1 and STARS systems is statistically significant when simplifying Type 1 sentences and Type 2 sentences. In the former case, the STARS system is superior while in the latter, the OB1 system is superior.

My response to **RQ-4** makes several original contributions. The development of a new sequence tagging method to identify the spans of compound clauses and complex$_{\overline{RF}}$ NPs is novel. Integration of the compound clause and complex constituent taggers into my approach to sentence simplification implements a tool that is easier to adapt and update and is capable of exploiting new tagging models derived with less human effort from new datasets which may belong to different text registers or comprise much larger amounts of text. The development of train-

ing data annotated with information about the spans of compound clauses and complex syntactic constituents and the analysis of the frequency distribution of different types of complex constituents is an additional contribution to research. Evaluation of the sequence tagging approach provides insights into the linguistic features characterising different elements of sentences containing compound clauses and complex$_{\overline{RF}}$ NPs. Detailed comparative evaluation of the STARS system for sentence simplification with the OB1 system which exploits handcrafted patterns provides insights useful for the future development of a hybrid approach.

A comparison of the statistics in Tables 4.4 and 4.5 with those in Tables 4.16 and 4.18 in Chapter 4 suggests some correlation between the accuracy of the taggers and the proportion of register-specific sequences of each type present in the training data. As a result, it may be useful in future work to expand the training set so that the tagging model can exploit more information about compound clauses and different types of complex constituents as they manifest in texts of different registers. In the training data, there are relatively few sequences containing compound clauses in texts of the registers of literature and news and containing complex constituents in literary texts. In future work, it may also be useful to supplement the training set with additional sequences from these text registers.

In the final quarter of 2019, on the basis of code examples provided in Tobias Sterbak's tutorial on *Named Entity Recognition with Bert*,[7] I implemented new

[7] Available at https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/. Last accessed: 8th January 2020.

versions of the sequence taggers to identify the spans of compound clauses and complex constituents. These versions exploit the large cased pretrained model based on Bidirectional Encoder Representations from Transformers (BERT).[8] At the time of writing, approaches exploiting pretrained BERT models with application-specific finetuning steps have achieved significantly improved accuracy in a range of NLP tasks. For the purpose of finetuning, I used the training data presented in Section 4.2, converted to the BIO format, to train the BERT sequence tagging models.

During finetuning, one tenth of the training data was used to validate the BERT models. Once derived, they were tested using the validation datasets presented in Sections 4.4.1 and 4.4.2 of this thesis but encoded in the BIO format. Calculated over all three text registers, these models achieved micro-averaged $F_1 = 0.8231$ when tagging compound clauses and $F_1 = 0.7505$ when tagging complex constituents. This compares with $F_1 = 0.7281$ and $F_1 = 0.5492$, respectively, for the CRF tagging models presented in Section 4.3. These preliminary results imply that one promising direction of future research may involve integrating the pretrained BERT models rather than the CRF tagging models into the STARS method for sentence simplification.

STARS (Section 5.2) exploits shallow syntactic analysis steps implemented using data-driven approaches (tagging of signs of syntactic complexity, compound clauses, and complex$_{\overline{RF}}$ NPs) and a small set of simple transformation rules. For this reason, I expect that it will be portable for use with other languages. In

[8]For the finetuning process, the *BertForTokenClassification* model was used.

future work, it will be interesting to explore this type of adaptation.

## 8.5 Research Question RQ-5

My response to **RQ-5** was based on extrinsic evaluation of the OB1 sentence simplification system via three NLP applications: multidocument summarisation (MDS), semantic role labelling (SRL), and information extraction (IE). Evaluations of this type were presented in Chapter 7 of the thesis.

Exploration of this research question makes several original contributions. While automatic sentence simplification methods have been evaluated extrinsically in previous work, with respect to subsequent language processing tasks such as information extraction (Evans, 2011; Niklaus *et al.*, 2016), machine translation (Štajner and Popović, 2018), and semantic role labelling (Vickrey and Koller, 2008),[9] investigation of its effect on the performance of automatic multidocument text summarisation is a novel contribution. Unfortunately, extrinsic evaluation of the OB1 approach to sentence simplification via a black box MDS system indicates that my method evokes no change in the accuracy of that system. There is a possibility that improvements could be evoked in the dependent MDS system if such a system could access and exploit information about the specific types of sentence simplified and the particular sentence transformation schemes (and rule activation patterns) that were applied.

The extrinsic evaluations described in Chapter 7 showed that integration of

[9] In their study, as previously noted, the role of sentence simplification was intrinsic. I am not aware of other studies into the specific contribution brought by a text simplification module to SRL.

my approach to sentence simplification as a preprocessing step in SRL and IE systems evokes improved accuracy.

In addition to the MDS, SRL, and IE systems used to extrinsically evaluate OB1, future work could include evaluation via other tasks in NLP. In collaboration with Hanna Bechara at the University of Wolverhampton, I integrated OB1 as a preprocessing step in a slightly reduced version of Gupta *et al.*'s (2014) system to calculate text similarity between sentences. In this context, as in MDS, integration of the sentence simplification method did not lead to improvements in the accuracy of the dependent NLP task.

In future work, it would be interesting to extrinsically evaluate both the STARS system, which is able to more accurately transform Type 1 sentences, and updated versions of STARS integrating improved sequence tagging models.

Beyond NLP applications, as noted in chapters 6 and 7 of this thesis, it is difficult to assess the extent to which text simplification systems may help readers to better comprehend a text. It has been noted by Klare (1976) and others that reading comprehension and text readability depends on factors related to both the text and the reader (e.g. reading competence). Studies suggest that the effect of changing a text with respect to established readability metrics on human perceptions of readability depends to a great extent on the interest of the reader. Simplifications of a text are more useful when its subject matter is outside the reader's background knowledge. They are less useful when its subject matter is of interest to the reader.

The research described in this thesis was motivated by a project to develop

language technologies to improve text accessibility for autistic individuals (Orăsan *et al.*, 2018). Although there is a wide range of psycholinguistic research investigating the reading difficulties of this population, this research is, of necessity, based on artificial examples in which linguistic variables are controlled. They focus on features of reading difficulty. There has been less research conducted into the effect of text simplification operations on reading comprehension.

In future work, it will be interesting to apply methods from cognitive science to investigate the impact of text simplification operations proposed to mitigate features of text difficulty in real-world reading tasks. By combining information about the magnitude of changes in reading comprehension brought about through application of specific simplification operations with empirical information about the relative frequencies with which these features occur, it may be possible to prioritise the development of assistive language technologies addressing the most serious and most commonly occurring features of text difficulty. This process will be possible only through active collaboration between researchers in cognitive sciences (e.g. clinical linguistics and psycholinguistics), researchers in natural language processing, and end users of the assistive language technologies to be developed.

# Bibliography

Abu-Jbara, A. and Radev, D. (2011). Coherent citation-based summarization of scientific papers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 500–509, Stroudsburg, PA, USA. Association for Computational Linguistics.

Agarwal, R. and Boggess, L. (1992). A simple but useful approach to conjunct identification. In *Proceedings of the 30th Annual Meeting for Computational Linguistics*, pages 15–21, Newark, Delaware. Association for Computational Linguistics.

Alabbas, M., Khalaf, Z. A., and Khasan, K. M. (2014). BASRAH: an automatic system to identify the meter of Arabic poetry. *Natural Language Engineering*, **20**(1), 131–149.

Aluisio, S. M., Specia, L., Pardo, T. A. S., Maziero, E. G., and Fortes, R. P. (2008a). A corpus analysis of simple account texts and the proposal of simplification strategies: First steps towards text simplification systems. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication, (SIGDOC '08)*, pages 240–248, Lisbon, Portugal.

Aluisio, S. M., Specia, L., Pardo, T. A. S., Maziero, E. G., and Fortes, R. P.

(2008b). Towards Brazilian Portuguese automatic text simplification systems. In *Proceedings of the Eighth ACM Symposium on Document Engineering (DocEng '08)*, pages 240–248, New York, USA.

Amoia, M. and Romanelli, M. (2012). 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.

Angrosh, M. and Siddharthan, A. (2014). Text simplification using synchronous dependency grammars: Generalising automatically harvested rules. In *Proceedings of the 8th International Natural Language Generation Conference*, pages 16–25, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Angrosh, M., Nomoto, T., and Siddharthan, A. (2014). 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. Association for Computational Linguistics.

Barbu, E., Martín-Valdivia, M. T., Martínez-Cámara, E., and na López, L. A. U. (2015). Language technologies applied to document simplification for helping

autistic people. *Expert Systems with Applications: An International Journal*, **42**(12), 5076–5086.

Bautista, S. and Saggion, H. (2014). Can numerical expressions be simpler? Implementation and demonstration of a numerical simplification system for Spanish. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M. A., and Schasberger, B. (1995). Bracketing Guidelines for Treebank II Style Penn Treebank Project. Technical report, University of Pennsylvania.

Biran, O., Brody, S., and Elhadad, N. (2011). Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: shortpapers (ACL-2011)*, pages 496–501, Portland, Oregon.

Bonial, C., Bonn, J., Conger, K., Hwang, J., Palmer, M., and Reese, N. (2015). English PropBank Annotation Guidelines. Technical report, Center for Computational Language and Education Research, Institute of Cognitive Science, University of Colorado at Boulder.

Bos, J. (2008). Wide-coverage semantic analysis with boxer. In *Proceedings of the 2008 Conference in Semantics in Text Processing*, pages 277–286, Venice, Italy.

Bosma, W. (2005). Image retrieval supports multimedia authoring. In E. Zudilova-Seinstra and T. Adriaansen, editors, *Linguistic Engineering meets Cognitive Engineering in Multimodal Systems*, pages 89–94. ITC-irst, Trento, Italy.

Bott, S., Saggion, H., and Figueroa, D. (2012a). A hybrid system for Spanish text simplification. In *Proceedings of the NAACL-HLT 2012 Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, pages 75–84, Montreal, Canada. Association of Computational Linguistics.

Bott, S., Rello, L., Drndarevic, B., and Saggion, H. (2012b). Can Spanish be simpler? LexSiS: Lexical Simplification for Spanish. In *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics*, Lecture Notes in Computer Science, pages 8–15, Samos, Greece. Springer.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.

Brill, E. (1994). Some advances in transformation-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, Washington.

Brouwers, L., Bernhard, D., Ligozat, A.-L., and Francois, T. (2014). Syntactic sentence simplification for french. In *Proceedings of the 3rd Workshop on Pre-*

*dicting and Improving Text Readability for Target Reader Populations (PITR) at EACL 2014*, pages 47–56, Gothenburg, Sweden. Association for Computational Linguistics.

Brown, C., Snodgrass, T., Kemper, S. J., Herman, R., and Covington, M. A. (2008). Automatic measurement of propositional idea density from part-of-speech tagging. *Behavior Research Methods*, **40 (2)**, 540–545.

Candido, A., Maziero, E., Specia, L., Gasperin, C., Pardo, T., and Aluisio, S. (2009). Supporting the adaptation of texts for poor literacy readers: a text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 34–42.

Canning, Y. (2002). *Syntactic Simplification of Text*. Ph.d. thesis, University of Sunderland.

Caplan, D. and Waters, G. S. (1999). Verbal working memory and sentence comprehension. *Behavioural and Brain Sciences*, **22**, 77–126.

Caramazza, A. and Zurif, E. B. (1976). Dissociation of algorithmic and heuristic processes in language comprehension: Evidence from aphasia. *Brain and Language*, **3**, 572–582.

Cederberg, S. and Widdows, D. (2003). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction.

In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, Edmonton, Canada.

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

Chandrasekar, R., Doran, C., and Srinivas, B. (1996). Motivations and methods for text simplification. In *Proceedings of the 16th International Conference on Computational Lingusitics (COLING '96)*, pages 1041–1044, Copenhagen, Denmark.

Chantree, F., Kilgarriff, A., Roeck, A. D., and Willis, A. (2005). Using a Distributional Thesaurus to Resolve Coordination Ambiguities. Technical Report 2005/02, The Open University, Milton Keynes, England.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine $n$-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 173–180, Ann Arbor.

Cherry, L. L. and Vesterman, W. (1981). Writing Tools The STYLE and DICTION programs. Computer Science Technical Report 91, Bell Laboratories, Murray Hill, New Jersey.

Chinchor, N. (1992). The statistical significance of the MUC-4 results. In *Proceedings of the Fourth Message Understanding Conference*, pages 30–50, McLean, Virginia.

Chomsky, N. (1977). *On Wh-Movement*, pages 71–132. Academic Press, New York.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin, and Use*. Greenwood Publishing Group, Santa Barbara, California.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**(1), 37–46.

Cohn, T. and Lapata, M. (2009). Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, **20**(34), 637–74.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. Ph.d thesis, University of Pennsylvania.

Collins, M. and Koo, T. (2005). Discriminative reranking for natural language parsing. *Computational Linguistics*, **31**(1), 25–69.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, **12**, 2493–2537.

Copestake, A. and Flickinger, D. (2000). An open source grammar development environment and broad-coverage english grammar using hpsg. In *Proceedings of LREC 2000*, pages 591–600.

Coster, W. and Kauchak, D. (2011). Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association*

*for Computational Linguistics (ACL-2011)*, pages 665–669, Portland, Oregon. Association of Computational Linguistics.

Covington, M. A. (2012). CPIDR® 5.1 User Manual. Technical report, Institute for Artificial Intelligence, University of Georgia, Athens, Georgia, U.S.A.

Daelemans, W., Höthker, A., and Tjong Kim Sang, E. (2004). Automatic sentence simplification for subtitling in dutch and english. In M. Lino, M. Xavier, F. Ferreira, R. Costa, and R. Silva, editors, *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1045–1048, Lisbon.

Daelemans, W., Zavrel, J., van der Sloot, K., and van den Bosch, A. (2010). TiMBL: Tilburg Memory-Based Learner Version 6.3 Reference Guide. Technical Report 10-01, ILK, Tilburg, The Netherlands.

De Belder, J. and Moens, M.-F. (2010). Text simplification for children. In *Proceedings of the SIGIR Workshop on Accessible Search Systems*, pages 19–26, Geneva, Switzerland.

De Belder, J., Deschacht, K., and Moens, M.-F. (2010). Lexical simplfication. In *Proceedings of the 1st International Conference on Interdisciplinary Research on Technology, Education, and Communication (Itrec-2010)*, Kortrijk, Belgium.

de Marneffe, M.-C., MacCartney, W., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *In Proceedings of*

*the International Conference on Language Resources and Evaluation (LREC 2004*, pages 449–54, Genoa, Italy.

DeFrancesco, C. and Perkins, K. (2012). An analysis of the proposition density, sentence and clause types, and nonfinite verbal usage in two college textbooks. In M. S. Plakhotnik, S. M. Nielsen, and D. M. Pane, editors, *Proceedings of the 11th Annual College of Education & GSN Research Conference*, pages 20–25, Miami: Florida International University.

Delisle, S. and Szpakowicz, S. (1995). Realistic parsing: Practical solutions of difficult problems. In *Proceedings of the 2nd Conference of the Pacific Association for Computational Linguistics*, pages 260–265, Queensland, Australia.

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

Dornescu, I., Evans, R., and Orasan, C. (2013). A tagging approach to identify complex constituents for text simplification. In *Proceedings of Recent Advances in Natural Language Processing*, pages 221 – 229, Hissar, Bulgaria.

dos Santos, L. S. F. C., Prates, M. O., de Oliveira Maia, G., Almeida, G. L. M. D., maria Cotta, D., Pedroso, R. C., and de Aquino araújo, A. (2018). Assessing if an automated method for identifying features in texts is better than another: Discussions and results. Technical report.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, **67**, 547–619.

Elhadad, N. (2006). Comprehending technical texts: predicting and defining unfamiliar terms. In *AMIA Annual Symposium Proceedings*, pages 239–243.

Evans, R. (2011). Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and Linguistic Computing*, **26 (4)**, 371–388.

Evans, R. and Orasan, C. (2013). Annotating signs of syntactic complexity to support sentence simplification. In I. Habernal and V. Matousek, editors, *Text, Speech and Dialogue. Proceedings of the 16th International Conference TSD 2013*, pages 92–104. Springer, Plzen, Czech Republic.

Evans, R. and Orasan, C. (2019a). Identifying signs of syntactic complexity for rule-based sentence simplification. *Natural Language Engineering*, **25 (1)**, 69–119.

Evans, R. and Orasan, C. (2019b). Sentence simplification for semantic role labelling and information extraction. In *Proceedings of the International Conference "Recent Advances in Natural Language Processing '2019" (RANLP-2019)*, pages 285–294, Varna, Bulgaria.

Evans, R., Orasan, C., and Dornescu, I. (2014). An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 131–140, Gothenburg, Sweden. Association for Computational Linguistics.

Feblowitz, D. and Kauchak, D. (2013). Sentence simplification as tree transduction. In *Proceedings of the 2nd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 1–10, Sofia, Bulgaria. Association for Computational Linguistics.

Ferreira, F., Anes, M. D., and Horine, M. D. (1996). Exploring the use of prosody during language comprehension using the auditory moving window technique. *Journal of Psycholinguistic Research*, **25 (2)**, 273–290.

Ferrés, D., Marimon, M., and Saggion, H. (2015). A web-based text simplification system for english. *Procesamiento del Languaje Natural*, **55**, 191–194.

Frazier, L. and Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, **14**, 178–210.

Gaizauskas, R., Foster, J., Wilks, Y., Arundel, J., Clough, P., and Piao, S. (2001). The Meter corpus: A corpus for analysing journalistic text reuse. In *Proceedings of Corpus Linguistics 2001 Conference*, pages 214–223. Lancaster University Centre for Computer Corpus Research on Language.

Gerani, S., Mehdad, Y., Carenini, G., Ng, R. T., and Nejat, B. (2014). Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613, Doha, Qatar. Association for Computational Linguistics.

Gerber, L. and Hovy, E. H. (1998). Improving translation quality by manipulating sentence length. In D. Farwell, L. Gerber, and E. H. Hovy, editors, *AMTA*, volume 1529 of *Lecture Notes in Computer Science*, pages 448–460. Springer.

Gibson, E. (1998). Limguistic complexity: locality of syntactic dependencies. *Cognition*, **68**(1), 1–76.

Gibson, E. and Thomas, J. (1996). The processing complexity of english center-embedded and self-embedded structures. In C. Schutze, editor, *Proceedings of the NELS 26 Workshop on Language Processing: MIT Working Papers in Linguistics*. MIT Press.

Glavas, G. and Stajner, S. (2013). Event-centered simplication of news stories. In *Proceedings of the Student Workshop Held in Conjunction with RANLP-2013*, pages 71–78, Hissar, Bulgaria. RANLP.

Goldberg, M. (1999). An unsupervised model for statistically determining coordinate phrase attachment. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 610–614, College Park, Maryland.

Gómez-Rodríguez, C. and Vilares, D. (2018). Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1314–1324. Association for Computational Linguistics.

Gonzalez-Dios, I., Aranzabe, M. J., and de Ilarraza, A. D. (2018). The corpus

of Basque simplified texts (CBST). *Language Resources and Evaluation*, **52**, 217–247.

Gordon, P. C., Hendrick, R., and Johnson, M. (2001). Memory interference during language processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **27**(6), 1411–1423.

Grishman, R. (2005). Information extraction. In R. Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, pages 545–559. Oxford University Press.

Grover, C., Matheson, C., Mikheev, A., and Moens, M. (2000). LT TTT - a flexible tokenisation tool. In *In Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 1147–1154.

Gupta, R., Béchara, H., El Maarouf, I., and Orasan, C. (2014). UoW: NLP techniques developed at the University of Wolverhampton for semantic similarity and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 785–789.

Hajič, J. and Zemánek, P. (2004). Prague Arabic dependency treebank: Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110–117.

Hasler, E., de Gispert, A., Stahlberg, F., and Waite, A. (2017). Source sentence simplification for statistical machine translation. *Computer Speech & language*, **45**, 221–235.

Hepple, M. (2000). Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics*, pages 278–285, Hong Kong, China. Association for Computational Linguistics.

Hogan, D. (2007). Coordinate noun phrase disambiguation in a generative parsing model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 680–687, Prague, Czech Republic. Association for Computational Linguistics.

Inui, K., Fujita, A., Takahashi, T., Iida, R., and Iwakura, T. (2003). Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing - Volume 16*, PARAPHRASE '03, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jay, T. B. (2003). *The Psychology of Language*. Pearson, Upper Saddle Rive, NJ.

Jonnalagadda, S., Tari, L., Hakenberg, J., Baral, C., and Gonzalez, G. (2009). Towards effective sentence simplification for automatic processing of biomedical text. In *Proceedings of NAACL HLT 2009: Short Papers*, pages 177–180, Boulder, Colorado. Association for Computational Linguistics.

Just, M. A., Carpenter, P. A., and Thulborn, K. R. (1996). Brain activation modulated by sentence comprehension. *Science*, **274**, 114–116.

Kajiwara, T. and Yamamoto, K. (2015). Evaluation dataset and system for

japanese lexical simplification. In *Proceedings of the ACL-IJCNLP 2015 Student Research Workshop*, pages 35–40.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht.

Kandula, S., Curtis, D., and Zeng-Treitler, Q. (2010). A semantic and syntactic text simplification tool for health content. In *AMIA Annual Symposium Proceedings*, pages 366–370.

Kawahara, D. and Kurohashi, S. (2008). Coordination disambiguation without any similarities. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 425–432, Manchester, England.

Kim, M.-Y. and Lee, J.-H. (2003). S-clause segmentation for efficient syntactic analysis using decision trees. In *Proceedings of the Australasian Language Technology Workshop*, Melbourne, Australia.

Kincaid, J. P., Fishburne, R. P., Rogers, R. L., and Chissom, B. S. (1986). *Derivation of New Readability Formulas (Automatic Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*. CNTECHTRA, 8-75 edition.

King, J. and Just, M. A. (1991). Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language*, **30**(5), 580–602.

Kintsch, W. and van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, **85 (5)**, 363–394.

Kintsch, W. and Welsch, D. M. (1991). The construction–integration model: A framework for studying memory for text. In W. E. Hockley and S. Lewandowsky, editors, *Relating Theory and Data: Essays on Human Memory*, pages 367–385. Hillsdale, NJ: Erlbaum.

Klare, G. R. (1976). A second look at the validity of readability formulas. *Journal of Reading Behavior*, **VIII**(2), 129–152.

Klerke, S., Alonso, H. M., and Søgaard, A. (2015). Looking hard: Eye tracking for detecting grammaticality of automatically compressed sentences. In *NODALIDA*, pages 97–105. Linköping University Electronic Press / ACL.

Kover, S. T., Haebig, E., Oakes, A., McDuffie, A., Hagerman, R. J., and Abbeduto, L. (2012). Syntactic comprehension in boys with autism spectrum disorders: Evidence from specific constructions. In *Proceedings of the 2012 International Meeting for Autism Research*, Athens, Greece. International Society for Autism Research.

Kübler, S., Hinrichs, E., Maier, W., and Klett, E. (2009). Parsing coordinations. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 406–414, Athens, Greece. Association for Computational Linguistics.

Kudo, T. (2005). CRF++: Yet another CRF toolkit. *Software available at http://crfpp. sourceforge. net*.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann.

Lakshmi, S., Ram, R. V. S., and Sobha, L. D. (2012). Clause boundary identification for malayalam using crf. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012)*, pages 83–92, Mumbai, India. Association for Computational Linguistics.

Lei, C.-U., Man, K. L., and Ting, T. O. (2014). Using Coh-Metrix to analyse writing skills of students: A case study in a technological common core curriculum course. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol II, IMECS 2014*, pages 3–6. IMECS.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions and insertions and reversals. *Soviet Physics Doklady*, **10 (8)**, 707–710.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

MacDonald, M. C., Just, M. A., and Carpenter, P. A. (1992). Working memory constraints on the processing of syntactic ambiguity. *Cognitive Psychology*, **24**, 56–98.

Maier, W., Kübler, S., Hinrichs, E., and Kriwanek, J. (2012). Annotating coordination in the penn treebank. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 166–174, Jeju, Republic of Korea. Association for Computational Linguistics.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, **19**(2), 313–330.

Martos, J., Freire, S., González, A., Gil, D., Evans, R., Jordanova, V., Cerga, A., Shishkova, A., and Orasan, C. (2013). User Preferences: Updated. Technical Report D2.2, Deletrea, Madrid, Spain.

Max, A. (2000). *Syntactic Simplification - An Application to Text for Aphasic Readers*. MPhil in computer speech and language processing, University of Cambridge, Wolfson College.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

McDonald, R. T. and Nivre, J. (2011). Analyzing and integrating dependency parsers. *Computational Linguistics*, **37**(1), 197–230.

McNamara, D. S., Graesser, A. C., McCarthy, P. M., and Cai, Z. (2014). *Automated Evaluation of Text and Discourse with Coh-Metrix*. Cambridge University Press.

Meltzer, J. A., McCardle, J. J., Schafer, R. J., and Braun, A. R. (2009). Neural aspects of sentence comprehension: Syntactic complexity, reversibility, and reanalysis. *Cereb Cortex*, **20 (8)**, 1853–1864.

Mishra, K., Soni, A., Sharma, R., and Sharma, D. (2014). Exploring the effects of sentence simplification on hindi to english machine translation system. In *Proceedings of the Workshop on Automatic Text Simplification: Methods and Applications in the Multilingual Society*, pages 21–19, Dublin, Ireland. Association for Computational Linguistics.

Miwa, M., Stre, R., Miyao, Y., and Tsujii, J. (2010). Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 788–796, Beijing, China. Association for Computational Linguistics.

Muszyńska, E. (2016). Graph- and surface-level sentence chunking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics – Student Research Workshop*, pages 93–99, Berlin, Germany. Association for Computational Linguistics.

Nakov, P. and Hearst, M. (2005). Using the web as an implicit training set: Application to structural ambiguity resolution. In *Proceedings of Human Lan-*

*guage Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 835–842, Vancouver. Association for Computational Linguistics.

Narayan, S. and Gardent, C. (2014). Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 435–445, Baltimore, Maryland. Association for Computational Linguistics.

Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Niklaus, C., Bermeitinger, B., Handschuh, S., and Freitas, A. (2016). A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 170–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Norman, S., Kemper, S., Kynette, D., Cheung, H., and Anagnopoulos, C. (1991). Syntactic complexity and adults' running memory span. *Journal of Gerontology*, **46**(6), 346–351.

Nunberg, G., Briscoe, T., and Huddleston, R. (2002). Punctuation. In R. Hud-

dleston and G. K. Pullum, editors, *The Cambridge Grammar of the English Language*, pages 1724–1764. Cambridge University Press.

Ogden, C. K. (1932). *Basic English: A General Introduction with Rules and Grammar*. K. Paul, Trench, Trubner & Co., Ltd., London.

Omer, A. and Oakes, M. (2017). Arud, the metrical system of Arabic poetry, as a feature set for authorship attribution. In *Proceedings of the 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 431–436.

Orăsan, C., Evans, R., and Dornescu, I. (2013). Text simplification for people with autistic spectrum disorders. In D. Tufis, V. Rus, and C. Forascu, editors, *Towards Multilingual Europe 2020: A Romanian Perspective*, pages 287–312. Romanian Academy Publishing House.

Orăsan, C., Evans, R., and Mitkov, R. (2018). Intelligent text processing to help readers with autism. In K. Shaalan, A. E. Hassanien, and M. F. Tolba, editors, *Intelligent Natural Language Processing: Trends and Applications*, pages 713–740. Springer.

Paetzold, G. (2015). 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.

Paetzold, G. H. and Specia, L. (2013). Text simplification as tree transduction. In

*Proceedings of the 9th Brazillian Symposium in Information and Human Language Technology*, Fortaleza, CE, Brazil. Sociedade Brasileira de Computação.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, **31**(1), 71–106.

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*. Longman.

Radev, D., Blitzer, J., Winkel, A., Allison, T., and Topper, M. (2006). MEAD Documentation v3.10. Technical report, University of Michigan.

Rapp, R. (2009). The Back-translation Score: Automatic MT Evaluation at the Sentence Level without Reference Translations. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 133–136, Portland, Oregon. Association for Computational Linguistics.

Ratnaparkhi, A., Roukos, S., and Ward, R. T. (1994). A maximum entropy model for parsing. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 803–806, Yokohama, Japan.

Rayner, K., Chace, K. H., Slattery, T. J., and Ashby, J. (2006). Eye movements as reflections of comprehension processes in reading. *Scientific Studies of Reading*, **10**(3), 241–255.

Reeve, L. H., Han, H., and Brooks, A. D. (2007). The use of domain-specific concepts in biomedical text summarization. *Information Processing & Management*, **43**(6), 1765–1776. Text Summarization.

Rello, L., Baeza-Yates, R., Bott, S., and Saggion, H. (2013). Simplify or help?: Text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, pages 15:1–15:10, New York, NY, USA.

Rennes, E. and Jönsson, A. (2015). A tool for automatic simplification of swedish texts. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 317–320, Vilnius, Lithuania. LiU Electronic Press.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *The Journal of Artificial Intelligence Research*, (11), 95–130.

Rus, V., Moldovan, D., and Bolohan, O. (2002). Bracketing compound nouns for logic form derivation. In S. M. Haller and G. Simmons, editors, *Proceedings of the FLAIRS Conference*. AAAI Press.

Saggion, H. (2017). *Automatic Text Simplification*. Morgan & Claypool Publishers, San Rafael, California.

Saggion, H. (2018). Text Simplification. In R. Mitkov, editor, *The Oxford Handbook of Computational Linguistics 2nd Edition*. Oxford University Press.

Saggion, H., Štajner, S., Bott, S., Mille, S., Rello, L., and Drndarevic, B. (2015). Making it Simplext: Implementation and evaluation of a text simplification

system for Spanish. *ACM Transactions on Accessible Computing (TACCESS) – Special Issue on Speech and Language Processing for AT (Part 2)*, **6**(4).

Scarton, C., Aprosio, A. P., Tonelli, S., Martin-Wanton, T., and Specia, L. (2017). MUSST: A multilingual syntactic simplification tool. In *The Companion Volume of the IJCNLP 2017 Proceedings: System Demonstrations*, pages 25–28, Taipei, Taiwan.

Schwarm, S. and Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 523–530, Ann Arbor, Michigan. Association for Computational Linguistics.

Seretan, V. (2012). Acquisition of syntactic simplification rules for french. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 4019–4026. European Language Resources Association (ELRA).

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics.

Shardlow, M. and Nawaz, R. (2019). Neural text simplification of clinical letters with a domain specific phrase table. In *Proceedings of the 57th Annual Meeting*

*of the Association for Computational Linguistics*, pages 380–389, Florence, Italy. Association for Computational Linguistics.

Sheremetyeva, S. (2014). Automatic text simplification for handling intellectual property (the case of multiple patent claims). In *Proceedings of the Workshop on Automatic Text Simplification: Methods and Applications in the Multilingual Society*, pages 41–52, Dublin, Ireland. Association for Computational Linguistics.

Shieber, S. M. and Schabes, Y. (1990). Synchronous tree-adjoining grammars. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*, COLING '90, pages 253–258, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shimbo, M. and Hara, K. (2007). A discriminative learning model for coordinate conjunctions. In *Proceedings of the 2007 Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 610–619, Prague.

Si, L. and Callan, J. (2001). A statistical model for scientific readability. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 574–576, New York, USA. Association for Computing Machinery.

Siddharthan, A. (2004). *Syntactic Simplification and Text Cohesion*. Ph.d. thesis, University of Cambridge.

Siddharthan, A. (2006). Syntactic simplification and text cohesion. *Research on Language and Computation*, **4:1**, 77–109.

Siddharthan, A. (2011). Text simplification using typed dependencies: a comparison of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG '11)*, pages 2–11, Nancy, France. Association for Computational Linguistics.

Siddharthan, A. (2014). A survey of research on text simplification. *ITL-International Journal of Applied Linguistics: Special Issue on Recent Advances in Automatic Readability Assessment and Text Simplification*, **165**(2), 259–298.

Siddharthan, A. and Angrosh, M. A. (2011). 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, Gothenburg, Sweden. Association for Computational Linguistics.

Simov, K., Popova, G., and Osenova, P. (2002). HPSG-based syntactic treebank of Bulgarian (BulTreeBank). In A. Wilson, P. Rayson, and T. McEnery, editors, *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*, pages 135–142. Lincom-Europa, Munich.

Specia, L. (2010). Translating from complex to simplified sentences. In *Proceedings of the Conference on Computational Processing of the Portuguese Language*, pages 30–39, Porto Alegre, RS, Brazil. Springer.

Specia, L., Jauhar, S. K., and Mihalcea, R. (2012). SemEval-2012 Task 1: English 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 347–355, Montréal, Canada. Association for Computational Linguistics.

Srikumar, V., Reichart, R., Sammons, M., Rappoport, A., and Roth, D. (2008). Extraction of entailed semantic relations through syntax-based comma resolution. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 1030–1038, Columbus, OH, USA. Association for Computational Linguistics.

Štajner, S. and Popović, M. (2018). Improving machine translation of English relative clauses with automatic text simplification. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 39–48, Tilburg, the Netherlands. Association for Computational Linguistics.

Steedman, M. (1987). Combinatory grammars and parasitic gaps. *Natural Language & and Linguistic Theory*, **5**(3), 403–439.

Suter, J., Ebling, S., and Volk, M. (2016). Rule-based automatic text simplification for German. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 279–287, Bochum, Germany.

Sutton, C. and McCallum, A. (2011). An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, **4:4**, 268–373.

Tager-Flusberg, H. (1981). Sentence comprehension in autistic children. *Applied Psycholinguistics*, **2:1**, 5–24.

Timm, A. (2018). *Looking at Text Simplification - Using Eye Tracking to Evaluate the Readability of Automatically Simplified Sentences*. Bachelor thesis, Institutionen för datavetenskap, Linköpings universitet.

Tomita, M. (1985). *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems*. Kluwer Academic Publishers, Norwell, MA, USA.

Štajner, S., Calixto, I., and Saggion, H. (2015). Automatic text simplification for Spanish: Comparative evaluation of various simplification strategies. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2015)*, pages 618–626, Hissar, Bulgaria.

van Delden, S. and Gomez, F. (2002). Combining finite state automata and a greedy learning algorithm to determine the syntactic roles of commas. In *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '02, pages 293–301, Washington, DC, USA. IEEE Computer Society.

Vickrey, D. and Koller, D. (2008). Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio.

Viera, A. J. and Garrett, J. M. (2005). Understanding interobserver agreement: The kappa statistic. *Family Medicine*, **37**(5), 360–363.

Štajner, S. (2015). *New Data-Driven Approaches to Text Simplification*. Ph.d. thesis, University of Wolverhampton.

Vu, T., Tran, G. B., and Pham, S. B. (2014). *Learning to Simplify Children Stories with Limited Data*, pages 31–41. Springer, Bangkook, Thailand.

Vu, T., Hu, B., Munkhdalai, T., and Yu, H. (2018). Sentence simplification with memory-augmented neural networks. In *Processings of NAACL-HLT 2018*, pages 79–85, New Orleans, Louisiana. Association for Computational Linguistics.

Walker, A., Siddharthan, A., and Starkey, A. (2011). Investigation into human preference between common and unambiguous lexical substitutions. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*, pages 176–180, Nancy, France.

Waters, G. and Caplan, D. (1996). Processing resource capacity and the comprehension of garden path sentences. *Memory and Cognition*, **24**, 342–355.

Wendt, D., Brand, T., and Kollmeier, B. (2014). An eye-tracking paradigm for analyzing the processing time of sentences with different linguistic complexities. *PLoS ONE*, **9**(6).

Wieting, J., Berg-Kirkpatrick, T., Gimpel, K., and Neubig, G. (2019). Beyond BLEU: Training Neural Machine Translation with Semantic Similarity. In

*Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Woodsend, K. and Lapata, M. (2011). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 409–20, Edinburgh, Scotland. Association for Computational Linguistics.

Wubben, S., van den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL-12)*, pages 1015–1024, Jeju, Republic of South Korea. Association for Computational Linguistics.

Xu, W., Callison-Burch, C., and Napoles, C. (2015). Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, **3**, 283–297.

Xu, W., Napoles, C., Pavlick, E., Chen, Q., and Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. *TACL*, **4**, 401–415.

Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., and Lee, L. (2010). For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. In *Proceedings of Human Language Technologies: The 2010 Annual*

*Conference of the North American Chapter of the ACL*, pages 365–368, Los Angeles, California. Association of Computational Linguistics.

Yeari, M., van den Broek, P., and Oudega, M. (2015). Processing and memory of central versus peripheral information as a function of reading goals: evidence from eye-movements. *Reading and Writing*, **28 (8)**, 1071–1097.

Zeng-Treitler, Q., Goryachev, S., Kim, H., Keselman, A., and Rosendale, D. (2007). Making texts in electronic health records comprehensible to consumers: A prototype translator. In *AMIA Annual Symposium Proceedings*, pages 846–850.

Zhang, X. and Lapata, M. (2017). Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark.

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

# Appendix A

## List of Papers

Parts of this thesis appeared in the following peer-reviewed papers. For many of the topics discussed in both, the thesis provides more extensive coverage. It also covers complementary topics that were not included in the peer-reviewed papers. In this appendix, I provide a brief description of my contribution to each of these papers:

- Evans, R. and Orasan, C. (2019b). Sentence simplification for semantic role labelling and information extraction. In *Proceedings of the International Conference "Recent Advances in Natural Language Processing '2019" (RANLP-2019)*, pages 285–294, Varna, Bulgaria

  - I developed the sentence simplification method, ran the experiments, and contributed to analysis of the results.

- Evans, R. and Orasan, C. (2019a). Identifying signs of syntactic complexity for rule-based sentence simplification. *Natural Language Engineering*, **25 (1)**, 69–119

  - I developed the approach and contributed to its evaluation (development of evaluation scripts and data, readability assessments, surveys of

the opinions of human readers, and analysis of the evaluation results.

- Orăsan, C., Evans, R., and Mitkov, R. (2018). Intelligent text processing to help readers with autism. In K. Shaalan, A. E. Hassanien, and M. F. Tolba, editors, *Intelligent Natural Language Processing: Trends and Applications*, pages 713–740. Springer

  – I developed some of the text simplification components described in this paper and contributed to analysis and interpretation of the evaluation results.

- Evans, R., Orasan, C., and Dornescu, I. (2014). An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 131–140, Gothenburg, Sweden. Association for Computational Linguistics

  – I developed the sentence simplification method presented in this paper.

- Evans, R. and Orasan, C. (2013). Annotating signs of syntactic complexity to support sentence simplification. In I. Habernal and V. Matousek, editors, *Text, Speech and Dialogue. Proceedings of the 16th International Conference TSD 2013*, pages 92–104. Springer, Plzen, Czech Republic

  – I developed the annotation scheme and contributed to annotation of the corpus. I contributed to the analysis of the reliability of the annotation.

- Dornescu, I., Evans, R., and Orasan, C. (2013). A tagging approach to identify complex constituents for text simplification. In *Proceedings of Recent Advances in Natural Language Processing*, pages 221 – 229, Hissar, Bulgaria

  – I conceptualised the sign tagger and contributed to feature engineering and to development of the annotated evaluation data.

- Orăsan, C., Evans, R., and Dornescu, I. (2013). Text simplification for people with autistic spectrum disorders. In D. Tufis, V. Rus, and C. Forascu, editors, *Towards Multilingual Europe 2020: A Romanian Perspective*, pages 287–312. Romanian Academy Publishing House

  – I developed the sentence simplification method presented in this paper.

# Appendix B

# Example Sign Usage by Tag

## B.1 Coordinators

### B.1.1 Nominal Conjoins

This group of classes is used to denote signs of syntactic complexity that coordinate nominal conjoins. The conjoins are considered to be of matching levels of syntactic projection, including lexical (50), intermediate (51), and phrasal (52) levels.

(50) It is vital that truth [$_{CLN}$ and] justice are seen to be done.

(51) My property, house, vehicle, savings ... as well as my private company [$_{CIN}$ and] clients.

(52) Mr Justice Forbes told the pharmacists that both Mr Young [$_{CMN1}$ and] his girlfriend, Collette Jackson, 24, of Runcorn, Cheshire, had been devastated by the premature loss of their son.

(53) 'This case is not about whether GM crops are a good $\phi_i$ [$_{CMN4}$ or] a bad thing$_i$,' he said.

The coordination of conjoins at the intermediate projection level can be recog-

nised on the basis that they are usually multi-word phrases that share a specifier. This type of coordination can be distinguished from coordination of lexical nominal conjoins because an adjective in one conjoin should not modify the head noun of the other conjoin. It can be distinguished from coordination of phrasal nominal conjoins because the second conjoin cannot have an initial specifier (e.g. determiner), instead the two conjoins share the specifier of the first. Sentence (53) includes an instance of nominal coordination in which the conjoins are maximal projections and the head of the first has been elided.

### B.1.2 Verbal Conjoins

This group of classes is used to denote signs of syntactic complexity that coordinate verbal conjoins. The conjoins are considered to be of matching levels of syntactic projection, including lexical (54), maximal (55), and clausal (57) levels. In this thesis, I consider clauses to be extended projections of verbs.

(54) It states that the NHS does not have sole responsibility for providing [$_{CLV}$ and] funding long-term nursing care.

(55) "I will be going to see it in a couple of days [$_{CMV1}$ and] am really looking forward to celebrating with them then."

(56) They sent$_i$ one surveillance team to follow the London suppliers as they drove up the motorway[$_{CMV2}$ , and] $\phi_i$ another team to Francis's flat in the city centre.

(57) Brian and I wanted to live here[$_{CEV}$ ,] we wanted to emigrate, and if I could, I would.

Sentence (56) demonstrates coordination of verbal conjoins in which the head of the second is elided and the elliptical element is coindexed with the head of the first conjoin. Readers infer that the argument, *another team*, was also *sent*.

### B.1.3 Prepositional Conjoins

This group of classes is used to denote signs of syntactic complexity that coordinate prepositional conjoins. The conjoins are considered to be of matching levels of syntactic projection, including lexical (58) and phrasal (59) levels.

(58) Banfield carried out a sustained and systematic year-long campaign of abuse, mostly while on duty in uniform in [$_{CLP}$ or] near Parkside police station, Cambridge, where he served as a custody sergeant.

(59) Denning was charged in Prague with having sexually assaulted an array of boys, some as young as 12[$_{CMP}$ , and] of being the head of a paedophile ring which included two Frenchmen and an American.

The annotated corpus presented in Section 2.2.3 contains sentences such as (60) which appear to motivate the addition of a class label to denote coordinated prepositional phrases in which the head of the right conjoin has been elided.

(60) Asked why, he was heard to say: 'That's for me to know [and] you to

find out.'

However, in this sentence, the word *for* functions as a conjunction with a clause complement rather than as a preposition. The coordinator has the class label CEV.

### B.1.4 Descriptive Conjoins (Adjectival and Adverbial)

This group of classes includes those denoting signs of syntactic complexity that coordinate adjectival and adverbial conjoins. The conjoins share the same syntactic category and are of matching levels of syntactic projection, including morphemic (61), lexical ((62), (64)) and phrasal ((63), (65)) levels.

(61) Ultrasonography of the abdomen shows cholelithiasis with intra- [$_{CPA}$ and] extra-hepatic biliary dilation.

(62) "He had a stable [$_{CLA}$ and] loving family."

(63) "Yolanda was painfully shy [$_{CMA1}$ and] quite unable to join in the friendly banter in the classroom," she said.

(64) 'I have struggled for nearly three years to reach this point and it has taken every thing inside me mentally, physically[$_{CLAdv}$ ,] emotionally to get here.

(65) Francis organised the supply: sometimes through a Midlands gypsy who has become a millionaire from armed robbery and drug deals[$_{CMAdv}$ ,] sometimes through delivery "mules" whom he escorted to Jamaica.

In the annotation undertaken so far, coordination at the morphemic level has only been observed for adjectival conjoins.

### B.1.5 Combinatory and Quantificational

This pair of class labels comprises those denoting signs of syntactic complexity that coordinate their conjoins into atomic phrases with non-compositional meaning (66). This property means that syntactic simplification of sentences containing those phrases is not amenable to standard approaches. Simplified sentences derived from them will have meanings inconsistent with those of the original.

Coordination linking quantificational information (67) also creates phrases whose meaning can be regarded as non-compositional.

(66) The 62-year-old ex-SAS man, who was awarded the Military Medal by the Queen, was left high [$_{COMBINATORY}$ and] dry in the witness box after the judge, Mr Justice Morland, walked out, fed up with his unsolicited outbursts.

(67) And after the jury had been out for six [$_{CLQ}$ and] a half days, he was convicted only of grievous bodily harm.

One way to reduce syntactic complexity in a text is to rewrite sentences containing coordination as sets of simpler sentences derived by replacing compound constituents with each of their conjoins. Sentence (68) is derived in this way from (66). It will be noted that this transformation fails to preserve the original

intended meaning, that the 62-year-old SAS man was left without resources or help in the witness box.

(68) * The 62-year-old ex-SAS man, who was awarded the Military Medal by the Queen, was left high in the witness box... The 62-year-old ex-SAS man, who was awarded the Military Medal by the Queen, was left dry in the witness box...

The class COMBINATORY is also used to denote coordinators used in aphoristic sentences, which are often proverbs or fixed expressions (e.g. *first come, first served*; *the more, the merrier*) and in named entities. In some cases, combinatory coordination is signalled by the occurrence of adverbials such as *between*, *both*, *either*, or *together* in the sentence.

## B.2 Boundaries of Subordinate Clauses

### B.2.1 Nominal Subordinate Clauses

This pair of classes denotes signs of syntactic complexity bounding non-finite subordinate nominal clauses whose extant elements are maximal projections of nouns. They denote the left (69) and right (70) boundaries of these clauses.

(69) Actor Alec Baldwin will play Mr Conductor[$_{SSMN}$ ,] a new character specially created for Thomas And The Magic Railroad.

(70) His friends, Mark Picard, and Earl Petrie, both 24 and both of Kingston[$_{ESMN}$ ,]

were jailed for three months and 12 months respectively for their parts in the attack on Mr Lee and a man trying to help him.

### B.2.2 Finite and Non-Finite Verbal Subordinate Clauses

This group of classes denotes signs of syntactic complexity bounding finite and non-finite subordinate clauses whose extant elements are verbal. These elements may include maximal and extended projections of verbs. They denote the left (71) and right (72) boundaries of these clauses ((73); (74)).

(71) And he claimed he was engaged in dirty tricks$[_{SSMV}$ ,] keeping a book of people who used to collect envelopes of cash.

(72) "Being put into a psychiatric ward with people with long-term mental illnesses who are shaking with the drugs they are taking$[_{ESMV}$ ,] there's no way you can feel normal and be OK with yourself," she told BBC TV's That's Esther programme with Esther Rantzen.

(73) "That's simply not true," said Grobbelaar$[_{SSEV}$ , who] is suing The Sun for libel over allegations of match-fixing.

(74) Addaction, which still receives annually £3.5m of public and charitable funds$[_{ESEV}$ ,] continues to claim that there is no evidence that he committed any offence while he worked for them.

### B.2.3 Subordinate Prepositional Clauses

This pair of classes denotes signs of syntactic complexity bounding non-finite prepositional clauses whose extant elements are maximal projections of prepositions. They denote the left (75) and right (76) boundaries of these constituents.

(75) A month later[$_{SSMP}$ ,] on January 3 1983, Barwell was back on their patch.

(76) After more than a week's deliberation[$_{ESMP}$ ,] an Australian jury did not even find the drunken thug guilty of manslaughter.

### B.2.4 Subordinate Adjectival and Adverbial Clauses

This group of classes denotes signs of syntactic complexity bounding non-finite adjectival and adverbial clauses. They include left ((77); (79)) and right ((78); (80)) boundaries.

(77) A jury at Bristol Crown Court heard that Mr Guscott[$_{SSMA}$ ,] furious at having to brake, decided to 'teach Mr Jones a lesson'.

(78) Andrew Hawkins, 42, of Ham Farm Lane, Bristol[$_{ESMA}$ ,] admitted 14 specimen charges under trading standards laws, but Exeter Crown court was told that he had altered the odometers on hundreds of cars in Britain's worst car clocking case.

(79) He goes around putting two fingers up to everyone else[$_{SSMAdv}$ ,] usually

quite literally.

(80) Earlier[$_{ESMAdv}$, ] Judge Caroline Simpson had expressed her personal sympathy for Mr Hagland's friends and relatives.

### B.2.5 Speech-Related Subordinate Clauses

This group of classes denotes signs of syntactic complexity bounding a range of speech-related subordinate clauses, including interjections ((81); (82)), tag questions (83), and reported speech ((84); (85)).

(81) 'No[$_{SSMI}$,] my lord,' conceded Collins's barrister Robert Howe.

(82) 'I have told you before I am a father who has lost his son and I have the right to do anything to find out how I lost my son and, please[$_{ESMI}$ ,] I have asked you several times not to capitalise on my grief.'

(83) 'They've done a pretty good job[$_{STQ}$ ,] haven't they?' he'd told me earlier, looking around.

(84) "Always remember," he said[$_{SSCM}$ ,] "that The Beatles were a rock'n'roll band and that's why we were so good for so long, if that's not too immodest."

(85) "It was real shock for all of us[$_{ESCM}$ ,]" one said yesterday.

## B.3 Special Uses

The annotation scheme includes the class label SPECIAL to denote signs of potential syntactic complexity that have particular types of coordinating, bounding, specifying, or referential functions. It is also used to label cases of coordination involving conjoins that annotators are unable to categorise confidently.

Many possible signs of syntactic complexity can be classified as having SPECIAL uses, including instances of six of the seven distinct major types presented in Chapter 2 (Sections 2.2.1.1 and 2.2.1.2). The only potential signs that have not, so far, evidenced special uses, in the resources described in Section 2.2.3, are those consisting of a punctuation mark followed by a wh-word.

A significant proportion of potential signs of syntactic complexity are used with a range of functions that differentiates them from those presented in Section 2.2.1 of this thesis (*Annotation Scheme*). These include:

- signs with a specifying function.
- signs with an anaphoric function.
- signs linking conjoins that have additional patterns of elision.
- signs linking conjoins in ill-assorted coordination (Quirk *et al.*, 1985).
- signs linking or bounding conjoins that annotators are unable to categorise confidently.

These functions are elaborated in Sections B.3.1–B.3.5 of this appendix. As

with the examples cited in Section B.1.5, application of the automatic simplification method presented in Chapter 5 to sentences with these characteristics leads to output whose meaning is not consistent with that of the originals. For this reason, such signs of syntactic complexity are classified as SPECIAL to enable them to be identified and processed using more suitable methods.

### B.3.1 NP Specifier

Signs such as [*that*] and [*which*], which frequently occur as the left boundaries of subordinate clauses, can also function as the specifiers of noun phrases ((86); (87)).

(86) "We will have to see [$_{SPECIAL}$ which] ones need to be pursued," said Mr Van Miert, adding that as the airlines had not operated the schemes for as long as BA, fines were likely to be lower.

(87) "I'm quite happy to abandon [$_{SPECIAL}$ that] specific point" he said.

### B.3.2 Anaphoric

The sign [*that*], which frequently occurs as a complementiser (the left boundary of a subordinate clause), can also function as an anaphor (88).

(88) 'Because of your involvement in the past with trying to stop all [$_{SPECIAL}$ that] in your work, you more than anybody else should have known the misery of people who had become addicted.'

### B.3.3 Coordination Involving Additional Patterns of Elision

Examples of coordinators linking nominal (Section B.1.1), verbal (Section B.1.2), and prepositional (Section B.1.3) constituents, in which the head of one conjoin is elided, have already been presented. However, there are instances of coordination involving additional types of conjoin and patterns of ellipsis. To illustrate, Sentence (89) contains a coordinator linking two clausal conjoins in which the head verb has been elided from the second. In Sentence (90), two clauses are coordinated but the second conjoin has been entirely elided save for the negative modifier.

(89) The 38-page judgment stated that Mrs Coughlan, a tetraplegic, was entitled to free nursing care because her primary need for accommodation was$_i$ a health need [$_{SPECIAL}$ and] her nursing needs $\phi_i$ not 'incidental'.

(90) I intended to say: 'You went through a red light', only I don't remember whether I said it [$_{SPECIAL}$ or] not as the next thing I knew I was being grabbed by the defendant.

### B.3.4 Ill-Assorted Coordination

In general, it is assumed that coordination links conjoins that match in terms of form, function, and meaning. This implies that they are usually of matching syntactic categories. However, there are a proportion of instances that do not follow this pattern. Sentence (91) demonstrates ill-assorted coordination of a

verb and an adjective.[1]

(91) "Name something that is currently on BBC1 that gets people excited [$_{SPECIAL}$ and] talking about it.

Sentence (92) demonstrates ill-assorted coordination of an adjectival phrase and a prepositional phrase.

(92) Whether it was unlawful depended on whether the nursing services were "merely incidental or ancillary to the provision of accommodation which a local authority is under a duty to provide" [$_{SPECIAL}$ and] "of a nature" which a local authority providing social services could be expected to provide.

### B.3.5 Cases of Uncertainty

In some cases, the syntactic category of subordinate constituents is unclear. For example, in Sentence (93), the subordinate constituent functions as an adverbial, but is headed by a verb. In Sentence (94), it is difficult to derive the relation of the subordinate constituent to the main clause because the context in which this sentence occurs is absent. This demonstrates one of the limitations of the implemented annotation method which displays only the sentence containing the sign being annotated.

[1]The categorisation of *talking* as a verb in this case, rather than an adjective, follows from the observation that it cannot be modified by an adjective modifier, such as the adverb *very*.

(93) "It's a massive thing for anyone to face[$_{SPECIAL}$ ,] let alone a teenager - at that age it's very difficult."

(94) Not immediately[$_{SPECIAL}$ , but] it has opened the debate and put pressure on ministers to make clear who can expect to receive free long-term care.

# Appendix C

## Token Features: Tagging Compound Clauses and Complex Constituents

1. The token
2. The part of speech of the token or, for signs of syntactic complexity, the syntactic function
3. The token number in the document
4. Sentence length
5. The number of words between the token and the next:
   (a) word with a particular part of speech tag:
      i. IN (preposition)
      ii. VBD (past tense verb)
      iii. DT (determiner)
   (b) sign of syntactic complexity
6. The part of speech of the first word in the sequence
7. The position of the token in the sentence: the `first third`, `second third`, or `third third` of the sentence.

8. The number of verbs in the sentence that precede the token

9. The number of verbs in the sentence that follow the token

10. The number of words that follow the last sign of syntactic complexity in the same sentence as the token

11. The relationship of the token to the word *because*:

    (a) Independent of this word

    (b) Occurs prior to this word in the sentence

    (c) Occurs subsequent to this word in the sentence

    (d) Occurs both prior to and subsequent to this word in the sentence

    (e) Is the word *because*

12. Boolean features:

    (a) The token is a relative pronoun (wh-word or *that*).

    (b) The token matches the first lexical word in the sequence.

    (c) The part of speech of the token matches that of the first token in the sentence

    (d) The token has a part of speech tag or a sign tag that matches the tag of the subsequent sign of syntactic complexity in the sentence.

    (e) The token has a part of speech that matches the part of speech of the first token following the next sign of syntactic complexity.

(f) The token is verbal (has part of speech VB, VBG, VBD, VBN, or RB).

(g) The token is a clause complement word (See Table C.1 for an indicative list of such words).

(h) The sentence in which the token appears also contains a clause complement verb.

(i) The token is the word *when*.

(j) The sentence in which the token occurs contains the word *said*.

(k) The token is a colon.

(l) The token is the word *who* and the subsequent tokens include a comma immediately followed by a past tense verb (tagged VBD).

(m) The token is either of the words *that* or *which* and the rest of the sentence contains a comma immediately followed by a determiner (tagged DT).

(n) The token is a comparative conjunction (see Table C.2 for an indicative list of such conjunctions.

(o) The token is an adversative conjunction (see Table C.3 for an indicative list of such conjunctions.

(p) The token is a final/illative conjunction (see Table C.4 for an indicative list of such conjunctions.

(q) The token is the word *some*.

(r) The token is a possessive (tagged with any PoS tag that ends with the character *$*.)

13. The number of commas that occur in the same sentence as the token.

14. The number of determiners in the same sentence as the token.

15. The number of signs of syntactic complexity in the same sentence as the token.

16. The token is nominal (part of speech is CD, JJ, JJR, JJS, NN, NNP, NNPS, NNS, POS, or WDT, and possessive variants) verbal (part of speech is MD, RB, RBR, RBS, RP, VB, VBD, VBG, VBN, VBP, VBZ, WP, or WRB), or functional (any other part of speech).

17. The token is a coordinator: Yes (*and*, *but*, or *or*), Maybe (consisting of a punctuation mark followed by *and*, *but*, or *or*), or No (any other token).

18. Acoustic form of the token. Here, the token is transcribed to a simplified form such that consonant clusters are rendered as 'C', single consonants as 'c', vowel sequences as 'V', and single vowels as 'v'. A word such as *consonant* is thus transcribed as 'cvCvcvC'.[1]

19. The length of the token in characters.

[1]Development of this feature was inspired by Omer and Oakes's (2017) use of Alabbas et al.'s (2014) BASRAH system as a feature for authorship attribution of Arabic poetry.

Table C.1: *Clause complement words.*

| | | | | |
|---|---|---|---|---|
| | | **Verbs** | | |
| accept | acknowledge | add | admit | agree |
| allege | announce | answer | appreciate | argue |
| ask | aware | believe | certain | claim |
| clear | complain | concern | conclude | confirm |
| convince | decide | demonstrate | deny | disappoint |
| disclose | discover | doubt | dread | emerge |
| emphasise | ensure | establish | expect | explain |
| fear | feel | find | given | guess |
| hear | hold | hope | illustrate | indicate |
| infer | insist | intimate | imply | know |
| learn | maintain | mean | note | order |
| plain | possible | promise | protest | prove |
| provide | record | realise | recognise | recommend |
| read | realise | record | relate | remain |
| report | retort | reveal | rule | satisfy |
| saw | say | see | show | state |
| suggest | suspect | tell | terrified | testify |
| think | warn | | | |
| | | **Nouns** | | |
| allegation | admission | belief | manner | scale |
| view | way | | | |
| | | **Adjectives** | | |
| disappointed | obvious | | | |

Table C.2: *Comparative conjunctions*

| | | | |
|---|---|---|---|
| both (.*) and | by the same token | correspondingly | equally |
| in the same way | just as | likewise | similarly |

Table C.3: *Adversative conjunctions*

| | | | | | |
|---|---|---|---|---|---|
| although | contrariwise | conversely | despite | however | instead |
| nevertheless | nonetheless | though | whereas | while | yet |

Table C.4: *Final/illative conjunctions*

| | |
|---|---|
| hence | in consequence |
| of course | so that |
| so then | therefore |
| thus | |