Technical Report

UCAM-CL-TR-597
ISSN 1476-2986

Number 597

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Syntactic simplification and text cohesion

Advaith Siddharthan

August 2004

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted November 2003 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Gonville and Caius College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Abstract

Syntactic simplification is the process of reducing the grammatical complexity of a text, while retaining its information content and meaning. The aim of syntactic simplification is to make text easier to comprehend for human readers, or process by programs. In this thesis, I describe how syntactic simplification can be achieved using shallow robust analysis, a small set of hand-crafted simplification rules and a detailed analysis of the discourse-level aspects of syntactically rewriting text. I offer a treatment of relative clauses, apposition, coordination and subordination.

I present novel techniques for relative clause and appositive attachment. I argue that these attachment decisions are not purely syntactic. My approaches rely on a shallow discourse model and on animacy information obtained from a lexical knowledge base. I also show how clause and appositive boundaries can be determined reliably using a decision procedure based on local context, represented by part-of-speech tags and noun chunks.

I then formalise the interactions that take place between syntax and discourse during the simplification process. This is important because the usefulness of syntactic simplification in making a text accessible to a wider audience can be undermined if the rewritten text lacks cohesion. I describe how various generation issues like sentence ordering, cue-word selection, referring-expression generation, determiner choice and pronominal use can be resolved so as to preserve conjunctive and anaphoric cohesive-relations during syntactic simplification.

In order to perform syntactic simplification, I have had to address various natural language processing problems, including clause and appositive identification and attachment, pronoun resolution and referring-expression generation. I evaluate my approaches to solving each problem individually, and also present a holistic evaluation of my syntactic simplification system.

Acknowledgments

Ann Copestake, my supervisor, for being constantly accessible and supportive, for pointing me in all the right directions, and for reading draft after draft of this thesis.

Ted Briscoe, for invaluable comments on drafts of conference papers, Karen Sparck Jones and Simone Teufel for many fruitful discussions and numerous pointers to research papers. John Carroll, for introducing me to the field of text simplification. Numerous anonymous referees for feedback on drafts of conference papers.

Paula, Naila, Fabre, Eric, Ben, Aline, Judita, Anna, Donnla, Martin and Jana, from the NLP group and ten different countries, for so many interesting conversations. In particular, Paula and Naila (who bravely, if not uncomplainingly, put up with me at GS24) and Fabre, for their company during coffee breaks, punting trips, beer festivals, formal halls and more.

Also at the Computer Laboratory, Margaret Levitt and Lise Gough, for making administration look easy.

Ben, Prerna, Becks, Poshak and many others in Cambridge, Sunalini and Tara in Delhi and Mumbai, for keeping me wined and dined and for being friends.

My parents and brother, who in numerous ways have helped me get this far.

Clubs that have offered me respite from Cambridge. The CU Hillwalking Club, where I got used to walking with my head in the clouds. The CU Underwater Explorations Group, where I learnt to dive, and come back up to earth. The Little Shelford Cricket Club, where I released energy by swatting cricket balls.

My sources of funding. The Cambridge Commonwealth Trust, the Overseas Research Studentship Awards, Gonville & Caius College and all those gullible undergraduate supervisees.

Everyone else who has contributed to making these three years productive and enjoyable. The list is long.

Contents

List of Tables

List of Algorithms

List of Figures

1 *Introduction*

Text simplification can be defined as any process that reduces the syntactic or lexical complexity of a text while attempting to preserve its meaning and information content. The aim of text simplification is to make text easier to comprehend for a human user, or process by a program. An example of a piece of newspaper text simplified by hand for people with aphasia follows.

> **Original (From the Sunderland Echo)**
>
> City Clamping Services (CCS) were slammed by 41-year-old civil engineer Matthew Agar when they clamped his VW Polo in a private car park in Nile Street last month, and extracted the £75 on-the-spot cash fine which has outraged him and other clamped motorists.
>
> While the law generally supports clampers operating on private land, Mr Agar claims CCS's sign was not prominent enough to be a proper warning since it was too small and far away from where he parked to be legible.
>
> **Simplified by hand for aphasics**
>
> 41-year-old civil engineer Matthew Agar slammed City Clamping Services (CCS). He slammed them when they clamped his VW Polo in a private car park in Nile Street last month. They extracted the £75 on-the-spot cash fine. It has shocked him and other clamped drivers.
>
> The law generally backs clampers working on private land. But Mr Agar claims CCS's sign was not prominent enough to be a proper warning. The sign was not prominent since it was too little and far away from where he parked to be readable.

This example[1] was produced for the PSET (Practical Simplification of English Text) project (Devlin and Tait, 1998; Carroll et al., 1998) and illustrates many kinds of text simplification, including the dis-embedding of relative clauses, the separation of subordinate clauses and coordinated verb phrases, the conversion from passive to active voice and the replacement of difficult words with easier synonyms. I list a few examples below.

[1] This example is taken from the PSET project featured at the London Science Museum exhibition *'The Human Factor': Designing Products, Places and Jobs for People* (1999).

Dis-embedding relative clauses:

> ...extracted the £75 on-the-spot cash fine which has outraged him and other clamped motorists...
>
> ↓
>
> ... extracted the £75 on-the-spot cash fine. It has shocked him and other clamped drivers...

Conversion from passive to active voice:

> City Clamping Services (CCS) were slammed by 41-year-old civil engineer Matthew Agar...
>
> ↓
>
> 41-year-old civil engineer Matthew Agar slammed City Clamping Services (CCS)...

Separation of subordinated clauses:

> While the law generally supports clampers operating on private land, Mr Agar claims CCS's sign was not prominent enough to be a proper warning...
>
> ↓
>
> The law generally supports clampers operating on private land. But Mr Agar claims CCS's sign was not prominent enough to be a proper warning...

Lexical simplification:

supports → backs	motorists → drivers	outraged → shocked
legible → readable	operating → working	

This example raises the issue of specifying the criteria for judging one text to be simpler than another. A common method for assessing whether a text is suitable for a particular reading age is by means of using a *readability metric*, such as the Flesch readability score, proposed in 1943 and more recently popularised by Microsoft Word. These metrics are based solely on surface attributes of a text, such as average sentence and word lengths. The term *readability* is therefore a misnomer; these metrics do not attempt to judge how readable, well written or cohesive a text is, or even whether it is grammatical. Rather, they suggest what reading age a text (that is assumed to be well written, cohesive and relevant in content) is suitable for, by means of a calibration with school reading grades. I discuss these metrics (and how they should and should not be used) in detail in section 6.2, along with other methods of measuring readability.

Returning to the example above, the Flesch readability score for the original text is 40.3 (judged to be suitable for 12th grade and above), while the corresponding score for the simplified-by-hand text is 69.4 (judged to be suitable for 6th grade and above). This suggests that simplification can be expected to make the original news report accessible to a much wider audience, as long as the simplification process leaves the text well written and cohesive.

1.1 The Objectives of this Thesis

Syntactic and lexical simplification are different natural language processing tasks, requiring different resources, tools and techniques to perform and evaluate. This thesis restricts itself to simplifying difficult syntactic constructs and does not offer a treatment of lexical simplification. I now outline the objectives of this thesis, before discussing the uses of syntactic simplification in section 1.2.

My primary objective in this thesis is to provide a theory of syntactic simplification that formalises the interactions that take place between syntax and discourse during the simplification process. This is required in order to ensure that the simplified text remains cohesive, an essential requirement for it to be useful. I provide an overview of my theory of syntactic simplification in section 1.5 and present the details in chapters 3 – 5.

My second objective is to design a modular architecture for syntactic simplification that is firmly founded in theory and to present and evaluate robust shallow methods for implementing each module, providing an account of relative clauses, appositive phrases and conjoined clauses (coordinating, subordinating and correlative). The aim is to produce a working system that is fast enough that it can be used interactively at runtime. A major objective is to conduct a comprehensive evaluation of each component in my architecture individually and to also conduct a holistic evaluation of the complete syntactic simplification system.

My final goal is to relate the relatively nascent field of text-simplification to other more established areas in natural language processing.

I focus on the genre of newspaper reports. This genre is interesting for many reasons. As the example in the introduction suggests, newspaper reports have ample scope for simplification. Further, news reports often have complicated sentences right at the beginning, which serve as a summary of the report. This can make them inaccessible to many groups of people. In fact, the British Aphasia Society has specifically identified reading newspapers as a literary task that would help aphasics keep in touch with the world (Parr, 1993). As reports are aimed at presenting information, often in a narrative style, I can hope to avoid many troublesome issues that might arise in more literary genre, for example, preserving sarcasm and other higher order intentions of the writer. And finally, newspaper text is readily available in large quantities in electronic form for evaluation purposes.

1.2 What use is Syntactic Simplification?

In order to motivate this thesis, I discuss various human readers who might benefit from syntactic simplification in sections 1.2.1 – 1.2.3[2] and then discuss its uses in other computer applications in section 1.2.4.

1.2.1 *Syntax and Deafness*

Reading comprehension requires more than just knowledge of words and grammar. The reader also needs a cognitive base for language (which develops from learning to ma-

[2] Most of the experiments in this section are described in Quigley and Paul (1984) and Caplan (1992).

Syntactic Construct /	**Deaf Students**		**Hearing Students**
Evaluation Group Profile	**10 years**	**18 years**	**Avg. Across Ages 10-18**
Coordination	56%	86%	92%
Pronominalisation	39%	78%	90%
Passive Voice	54%	72%	78%
Relative Clause (RC)	51%	59%	84%
RC attachment	27%	56%	82%
Subordination	22%	59%	84%

Table 1.1. Summary of comprehension tests on deaf students (Quigley et al., 1977)

nipulate and expand a variety of linguistic experiences) in order to construct a plausible meaning for a sentence. Deaf children face many reading difficulties due to experiental and linguistic deficits incurred in their early childhood (Quigley and Paul, 1984) and usually learn to read with inadequately developed cognitive and linguistic skills. As both syntactic analysis and semantic interpretation are constrained by the same working memory (Carpenter et al., 1994), the more the working memory that is required for storing information during a parse, the less is the working memory available for "processing" meaning. As a result, the deaf have trouble comprehending syntactically complex sentences. I now summarise two studies that suggest that an automated syntactic simplification would indeed benefit the deaf.

The first study (Quigley et al., 1977) involved comprehension tests on a random sample of deaf students aged 10-19. I summarise the results in table 1.1, which shows the average percentage of examples for which the deaf students successfully answered a comprehension test. This experiment shows that 10-year-old deaf children have difficulty with all complex constructs. By the time they are 18, they are better able to comprehend coordination, pronominalisation and passive voice, but still have significant difficulty with relative clauses and subordinate clauses. Table 1.1 makes a distinction between comprehending relative clause constructs and correctly attaching them. To clarify this distinction, consider the following sentence:

The boy who hit the girl ran home.

Clause attachment is tested by asking the question *Who hit the girl?*. Clause comprehension is tested by asking the question *Who ran home?*. A tendency to interpret the above sentence as *the girl ran home* has been observed not only in the deaf, but also during first language acquisition in hearing children.

The second study (Robbins and Hatcher, 1981), involving comprehension tests on deaf children aged 9-12 years, found that passive voice, relative clauses, conjunctions and pronouns affected comprehension the most. Interestingly, Robbins and Hatcher (1981) also found that controlling for word recognition did not improve comprehension on sentences containing these constructs. This is a strong indication that syntactic simplification is indeed worth pursuing independent of lexical simplification.

Syntactic Construct / Experiment	Exp. 1	Exp. 2	Exp. 3
Active Voice	3.9	4.2	4.0
Passive Voice	2.8	3.2	3.2
Relative Clause (object position)	1.9	2.6	2.7
Coordination	1.5	2.0	1.9
Relative Clause (subject position)	1.2	1.3	1.4

Table 1.2. Summary of comprehension tests on aphasics (Caplan, 1992)

1.2.2 *Syntax and Aphasia*

Aphasia is a language disorder resulting from physical brain damage, usually due to a stroke or accident. While there are a variety of language problems associated with aphasia, depending on the extent and location of brain damage and the level of pre-aphasia literacy among other things, aphasics in general have trouble with long sentences, infrequent words and complicated grammatical constructs. Their language problems cause them to feel alienated from the rest of the world and they have themselves identified reading newspapers as a literary task that would help them keep in touch (Parr, 1993).

Shewan and Canter (1971) investigated the relative effects of syntactic complexity, vocabulary and sentence length on auditory comprehension in aphasics. Length was increased by adding prepositional phrases and adjectives, lexical difficulty was measured by frequency of use in normal language and syntactic complexity was increased using passivisation and negations. They concluded that syntactic complexity provided the most difficulty for aphasics. The significance of their findings for us is unclear, however, as their tests were on auditory rather than reading comprehension.

An experiment that is more informative about the relative difficulty of syntactic constructs for aphasics when reading is described in Caplan (1992). The author reports three experiments, involving 56, 37 and 49 aphasic patients, that test comprehension on sentences containing different syntactic constructs. I summarise these results in table 1.2. The subjects were presented with 5 examples of each sentence type. Table 1.2 shows the mean correct scores of the aphasic subjects on object manipulation tasks based on each example sentence. The maximum possible score is 5. The results in table 1.2 indicate a significant decrease in comprehension when sentences contain coordinated or embedded clauses or passive voice and indicate that syntactic simplification would indeed be useful for aphasics.

1.2.3 *Working Memory and Reading Levels*

As discussed above, the deaf require more higher order processing than the hearing, making comprehension difficult if the initial syntactic processing overloads their working memory. Aphasics tend to have reduced working memory due to physical brain damage. The extent to which this is a source of their comprehension problems is still a matter of debate, but it is widely accepted that it contributes to it.

Working memory can limit reading comprehension even for people without disabilities. There is a large body of research that suggests that there are differences in the way highly skilled and poor readers read. The most striking difference is at the word level. Vocabulary plays a primary role in reading and people for whom mapping words to meanings requires effort tend to be bad readers (Anderson and Freebody, 1981). Poor and beginning readers tend to have poor word processing skills; they rely overly on context and higher order mental processes and lack the efficient decoding skills of skilled readers. In fact they have to devote so much working memory to basic word processing that higher level processing suffers (Anderson, 1981; Quigley and Paul, 1984). This might also hold for people reading a language they are not confident in; for example, non native-English speakers across the world surfing a predominantly English internet.

There are also differences in the way information is chunked by poor and skilled readers. Skilled readers have a better ability to recode concepts and relations into single chunks, thus freeing up working memory for higher level processing (Daneman and Carpenter, 1980). In fact, Mason and Kendall (1979) showed that splitting complex sentences into several shorter ones resulted in better comprehension for less skilled readers. They attributed these results to the reduction in the amount of information stored in working memory during syntactic processing, arguing that this freed up working memory for higher level semantic processing at sentence boundaries. These studies suggest that syntactic simplification can aid comprehension by leaving more working memory available for higher order processing, not just for aphasics and the deaf, but also for a much wider target group including second language learners, non native-speakers, adult literacy students and people with low reading ages.

1.2.4 Assisting other NLP Applications

The previous section discussed how human readers might benefit from text simplification. Syntactic simplification might also be of use to other applications, as described below.

Syntactic simplification results in shorter sentences. It could therefore be used to preprocess texts before feeding them to a full-blown parser. This was the motivation for some early work (Chandrasekar et al., 1996; Chandrasekar and Srinivas, 1997) on simplification. Long sentences are problematic for parsers due to their high levels of ambiguity. Shortening sentences prior to parsing would increase parser throughput and reduce parser timeouts. It has been suggested (Chandrasekar and Srinivas, 1997) that the parses of simplified sentences can be combined to give the parse for the original sentence.

It is well documented that the performance of machine translation systems decreases with increased sentence length (Gerber and Hovy, 1998). It is therefore plausible that simplified sentences will also be easier to translate correctly. Text simplification could also improve the performance of summarisation systems based on sentence extraction as it results in less information per sentence and hence smaller units of information are extracted.

An increasing number of people are connecting to the internet using hand held devices and mobile phones. These devices have small screens with limited space to display text. Software that displays text in short sentences that fit on the screen might improve the practicality of these devices.

1.3 Some Related Fields

I now describe some areas in natural language processing that relate to the idea of simplifying text. Then, in section 1.4, I summarise the literature in the specific area of automatic text simplification.

1.3.1 Controlled Generation

While text simplification is a relatively unresearched field in natural language processing, there has been considerable interest in controlled generation, largely due to interest from industries in creating better (less ambiguous and easier to translate) user manuals (Wojcik et al., 1990; Wojcik and Hoard, 1996). EasyEnglish, part of IBM's internal editing environment that is used as a preprocessing step for machine-translating IBM manuals (Bernth, 1998), aims to help authors remove ambiguity prior to translation. For example, given the sentence:

> A message is sent to the operator requesting the correct tape volume.

EasyEnglish suggests a choice of the following unambiguous alternatives to the author:

> A message that requests the correct tape volume is sent to the operator
>
> OR
>
> A message is sent to the operator that requests the correct tape volume

Systems like EasyEnglish are essentially authoring tools that detect ambiguity, ungrammaticality and complicated constructs and help an author revise a document. They do not revise or generate documents automatically and are controlled-generation *aids* rather than natural language generation systems.

Natural language generation systems often adopt some form of user model to tailor text according to the end users' domain knowledge, usually providing the user a limited set of options like *expert*, *intermediate* or *beginner*. These options have traditionally been used to determine the level of technical detail in a generated text. More recently, there has been an acknowledgement that tailoring computer-generated text to the reading skills of particular user groups can be as important as tailoring the text to their domain knowledge. As an example, the STOP project (Reiter et al., 1999) was aimed at generating smoking-cessation letters based on questionnaires that the smokers filled online. The user questionnaire only affected content selection in STOP, but when analysing its performance, Reiter et al. (2003) commented that it would have been desirable to use the questionnaire for taking decisions in the microplanning and realisation stages as well, because the smokers had a wide range of reading abilities and did not always comprehend the generated text. In related work, Williams et al. (2003) examined the impact of discourse level choices on readability in the domain of reporting the results of literacy assessment tests. Williams et al. (2003) used the results of the test to control both the content and the realisation of the generated report.

1.3.2 Text Summarisation

While the simplification task suggests that the generated text retain all the information contained in the input text, removing less central information might aid comprehension among people who have poor reading ability. Aphasics are known to have difficulty with sentences containing multiple modifiers for nouns and verbs. A method of filtering out the less informative modifiers (adjectives, adverbs and prepositional phrases) might therefore benefit them. Filtering out the less informative portions of a text is a task that is central to text summarisation and automatic abstract generation.

In this respect, it is interesting to survey one particular subfield of text summarisation—*sentence shortening*. Grefenstette (1998) proposed the use of sentence shortening to generate telegraphic texts that would help a blind reader (with a text-to-speech software) skim a page in a manner similar to sighted readers. He provided eight levels of telegraphic reduction. The first (the most drastic) generated a stream of all the proper nouns in the text. The second generated all nouns in subject or object position. The third, in addition, included the head verbs. The least drastic reduction generated all subjects, head verbs, objects, subclauses and prepositions and dependent noun heads. Reproducing from an example in his paper, the sentence:

> Former Democratic National Committee finance director Richard Sullivan faced more pointed questioning from Republicans during his second day on the witness stand in the Senate's fund-raising investigation.

got shortened (with different levels of reduction) to:

- Richard Sullivan Republicans Senate.
- Richard Sullivan pointed questioning.
- Richard Sullivan faced pointed questioning.
- Richard Sullivan faced pointed questioning from Republicans during day on stand in Senate fund-raising investigation.

Grefenstette (1998) provided a simple rule-based approach to telegraphic reduction of the kind illustrated above. Since then, Knight and Marcu (2000) and Riezler et al. (2003) have explored statistical models for sentence shortening. Knight and Marcu (2000) used a noisy channel model that assumed that a shortened sentence (the source) gets expanded into the long sentence by a noisy channel. The problem of sentence shortening was then to, given the long sentence, decide what the most plausible short sentence was; in other words, to maximise the probability that the long sentence (that they were trying to shorten) had a generated short sentence as its source. Knight and Marcu (2000) used a supervised approach to learn the properties of the noisy channel (for example, that it frequently introduces determiners, adjectives, prepositional phrases etc). They used as a training corpus, a set of sentences that had been shortened by humans. Sentences and their shortened-by-hand versions were parsed and the trees compared. Then if, for example, the structure *(S (NP ...) (VP ...) (PP ...))* appeared in the shortened version as *(S (NP ...) (VP ...))*, it could be learnt that the channel introduced noise of the form $S \rightarrow NP\ VP \longrightarrow S \rightarrow NP\ VP\ PP$. Knight and Marcu (2000) used the training phase to learn noise introduction rules (also called transfer rules) like the one above and also derive the probabilities associated with their application by the channel. This provided the model of the noisy channel, which was then used to determine the most probable shortened sentence, given a long sentence and the desired level of reduction. For the evaluation, four human judges were presented with the original sentence and shortened versions produced using the noisy channel model and by humans (without being told which is which). They were asked to judge grammaticality and the importance of the selected words on a scale of $1 - 5$. They report that the average judge scores for the noisy channel approach were *grammaticality=4.34, importance=3.38* while the corresponding scores for human-generated sentences were *grammaticality=4.92, importance=4.24*.

Riezler et al. (2003) also tried to learn transfer rules automatically from a corpus. However, they used a more linguistically rich feature-structure grammar that produced fine grained dependency structures. This allowed them to handle structural modifications like nominalisation in addition to the deletion operations handled by Grefenstette (1998) and Knight and Marcu (2000). In the Riezler et al. (2003) approach, transfer rules for generating the shortened sentences were learnt using a maximum entropy model and then filtered using a constraint based generator, which guaranteed optimal grammaticality of the output. They reported similar results to Knight and Marcu (2000), with two human judges averaging 3.5 out of 5 when judging the importance of the words in the shortened text.

Sentence shortening is related to syntactic simplification in the sense that, like syntactic simplification, it results in shorter sentences. However, unlike syntactic simplification, sentence shortening does not necessarily preserve either information content or grammaticality (only Riezler et al. (2003) attempt to ensure grammaticality). This is because sentence shortening aims to help readers improve reading time by filtering out the less informative portions of a text. This is a different objective to that of syntactic simplification, which aims to help people with lower reading ages achieve better comprehension on the text.

Besides sentence shortening, there are other aspects of summarisation that relate to simplification. There has been research on how to pack the maximum information into a summary sentence (McKeown et al., 1995). The intuitions that govern how sentences should be combined could also be used to split them. One interesting issue is *content conflation* (Elhadad and Robin, 1992) where sentences like:

> Portland defeated Utah 101–97. It was a tight game where the lead kept changing.

can be conflated to generate:

> Portland outlasted Utah 101–97.

This is an example of paraphrasing, which can be a form of simplification. When applied forwards, text conflation generates shorter sentences due to the semantic richness of the verb. When applied backwards, difficult verbs like *outlast* can be replaced by paraphrases like *narrowly defeat* or *beat in a close game*. We can then lexically simplify:

> Portland outlasted Utah 101–97.

to:

> Portland beat Utah 101–97 in a close game.

1.4 Previous attempts at Text Simplification

Compared to controlled generation and text summarisation, there has been significantly less work done on the automatic simplification of existing text. Interestingly, the two main groups involved with text simplification have had very different motivations. The group at UPenn (Chandrasekar et al., 1996; Chandrasekar and Srinivas, 1997) viewed text simplification as a preprocessing tool to improve the performance of their parser. The PSET project on the other hand focused its research on simplifying newspaper text for aphasics (Carroll et al., 1998; Carroll et al., 1999b).

1.4.1 *Summary of Chandrasekar et al.'s Work*

Chandrasekar et al.'s motivation for text simplification was largely to reduce sentence length as a preprocessing step for a parser. They treated text simplification as a two-stage process— *analysis* followed by *transformation*. Their research focused on dis-embedding relative clauses and appositives and separating out coordinated clauses.

Their first approach (Chandrasekar et al., 1996) was to hand-craft simplification rules, the example from their paper being:

```
V W:NP, X:REL_PRON Y, Z. ⟶ V W Z. W Y.
```

which can be read as "if a sentence consists of any text *V* followed by a noun phrase *W*, a relative pronoun *X* and a sequence of words *Y* enclosed in commas and a sequence of words *Z*, then the embedded clause can be made into a new sentence with *W* as the subject noun phrase". This rule can, for example, be used to perform the following simplification:

> John, who was the CEO of a company, played golf.
>
> ↓
>
> John played golf. John was the CEO of a company.

In practice, linear pattern-matching rules like the hand-crafted one above do not work very well. For example, to simplify:

> A friend from London, who was the CEO of a company, played golf, usually on Sundays.

it is necessary to decide whether the relative clause attaches to *friend* or *London* and whether the clause ends at *company* or *golf*. And if a parser is used to resolve these ambiguities (as in their second approach summarised below), the intended use of text simplification as a preprocessor to a parser is harder to justify.

Their second approach (Chandrasekar and Srinivas, 1997) was to have the program learn simplification rules from an aligned corpus of sentences and their hand-simplified forms. The original and simplified sentences were parsed using a Lightweight Dependency Analyser (LDA) (Srinivas, 1997) that acted on the output of a supertagger (Joshi and Srinivas, 1994). These parses were chunked into phrases. Simplification rules were induced from a comparison of the structures of the chunked parses of the original and hand-simplified text. The learning algorithm worked by flattening subtrees that were the same on both sides of the rule, replacing identical strings of words with variables and then computing tree→trees transformations to obtain rules in terms of these variables.

This approach involved the manual simplification of a reasonable quantity of text. The authors justified this approach on the basis that hand-crafting rules is time consuming. However, it is likely that the intuitions used to manually simplify sentences can be encodable in rules without too much time overhead. And while this approach is interesting from the machine learning point of view, it seems unlikely that a system that learns from a corpus that has been simplified by hand will outperform a system in which the rules themselves have been hand-crafted.

Text simplification can increase the throughput of a parser only if it reduces the syntactic ambiguity in the text. Hence, a text simplification system has to be able to make disambiguation decisions without a parser in order to be of use to parsing. This early work on syntactic simplification therefore raised more issues than it addressed. And since the authors did not provide any evaluations, it is difficult to assess how well their approaches to text simplification worked.

1.4.2 Summary of the PSET project

The PSET project (Devlin and Tait, 1998; Carroll et al., 1998), in contrast, was aimed at people with aphasia rather than at parsers and was more justified in making use of a parser for the analysis stage. For syntactic simplification, the PSET project roughly followed the approach of Chandrasekar et al. PSET used a probabilistic LR parser (Briscoe and Carroll, 1995) for the analysis stage and unification-based pattern matching of hand-crafted rules over phrase-marker trees for the transformation stage. The project reports that on 100 news articles, the parser returned 81% full parses, 15% parse fragments and 4% parse failures.

An example of the kind of simplification rule used in the syntactic-simplification component of the PSET project is:

(S (?a) (S (?b) (S (?c)))) ⟶ (?a) (?c)

The left hand side of this rule unifies with structures of the form shown in figure 1.1 and the rule simply discards the conjunction *(?b)* and makes new sentences out of *(?a)* and *(?c)*. This rule can be used, for example, to perform the following simplification:

> The proceedings are unfair and any punishment from the guild would be unjustified.
>
> ↓
>
> The proceedings are unfair. Any punishment from the guild would be unjustified.

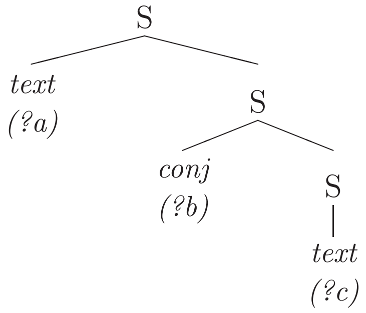


Figure 1.1. The structure matched by the pattern *(S (?a) (S (?b) (S (?c))))*

The PSET project explored a wide range of simplification options, including lexical simplification, conversion of passives to actives and resolving pronouns. Lexical simplification involves replacing difficult words with simpler synonyms. The PSET project used *WordNet* (Miller et al., 1993) to identify synonyms and obtained word frequency statistics from the Oxford Psycholinguistic Database (Quinlan, 1992) to determine the relative difficulty of words (Devlin and Tait, 1998).

The syntactic component of PSET comprised three components— anaphora resolution, syntactic simplification and anaphora replacement. The anaphora resolution algorithm was based on CogNIAC (Baldwin, 1997) and Canning et al. (2000b) report a recall of 60% with precision of 84% on newspaper text.

The syntactic constructs that the PSET project simplified were coordinated clauses and passive voice. Canning (2002) reports that there were only 75 instances of coordination in her corpus of 100 news reports from the Sunderland Echo. This meant that the level of simplification achieved was unlikely to be useful. As I describe in this thesis, a treatment of relative clauses, subordination and apposition can result in a higher level of simplification.

The attempt at converting passive voice to active had mixed success. Canning (2002) reports that only one out five passive constructs had an expressed surface agent. The rest were agentless; for example, in *She was taken to Sunderland Royal Hospital.* Further, passive constructs were often deeply embedded within a sentence, making the agent difficult to recover.

Canning (2002) reports that in her 100 news report corpus, there were only 33 agentive passive constructs. Out of these, her program converted only 55% correctly to active voice. Even the correctly converted sentences sometimes seemed odd; for example:

He was struck down by the brain disease last October.
↓
The brain disease last October struck him down.

The main contribution of the syntactic component of PSET was the application of a pronoun resolution algorithm to text simplification (Canning, 2002). The aim was to replace pronouns with their antecedent noun phrases, to help aphasics who might otherwise have difficulty in resolving them. Intra-sentential anaphora were not replaced, to avoid producing sentences like *Mr Smith said Mr Smith was unhappy.*

Canning (2002) conducted an evaluation of the effect of pronoun replacement on comprehension on 16 aphasic subjects and reported 20% faster reading times and 7% better scores on question answering tests when pronouns were replaced. User trials were only carried out for pronoun-replacement, however, and the passive-voice activisation and conjunction-separation modules were only evaluated on the basis of the grammaticality and meaning-preservation. Canning (2002) reports an accuracy of 75% for simplifying subordination and an accuracy of 55% for simplifying passive voice.

The fact that on average there was only one construct simplified per news report meant that the PSET project did not need to analyse the effects on syntactic simplification on text cohesion. As I describe in the next section, the issue of cohesion becomes important when simplifying relative clauses, apposition and subordination.

1.5 My Approach to Text Simplification

I begin by refining the definition of *text simplification* that I provided on page 1 to:

> **Text Simplification**: Any process that involves syntactic or lexical simplification of a text and results in a cohesive text.

where lexical and syntactic simplification are defined as:

> **Syntactic Simplification**: Any process that reduces the syntactic complexity of a text while preserving its meaning and information content.
>
> **Lexical Simplification**: Any process that reduces the lexical complexity of a text while preserving its meaning and information content.

Under these definitions, the approaches used by Chandrasekar et al. and the PSET project (described in section 1.4) qualify only as syntactic and, in the case of PSET, lexical simplification. The two-stage theory (*analysis* and *transformation*) of syntactic simplification used by them does not address the issue of text cohesion and cannot guarantee that the resulting text is *simpler*, only that it has simpler syntax. While these approaches result in text that is judged more readable (suitable for a lower reading age) by readability metrics such as the Flesch readability score (introduced on page 2 and described in detail in section 6.2), they cannot guarantee a prerequisite for using such metrics to make that judgement— that the simplified text is well written and cohesive.

My theory of text simplification therefore decomposes the task into three stages— *analysis*, *transformation* and *regeneration*. The first two stages correspond to those in the two-stage theory proposed by Chandrasekar et al. The text needs to be analysed in order to mark-up syntactic constructs that can be simplified. The analysed text can then be transformed using a set of hand crafted rules, similar to those described in sections 1.4.1 and 1.4.2.

My *regeneration* stage addresses the issue of preserving text cohesion. *Cohesion* is defined by Halliday and Hasan (1976) as the phenomenon where the interpretation of some element of discourse depends on the interpretation of another element and the presupposing element cannot be effectively decoded without recourse to the presupposed element. These cohesive relations between elements can be conjunctive or anaphoric, and sentence-level syntactic transforms have the potential to disrupt both. My approach tries to ensure that the simplification process does not change the presupposed element or make it inaccessible to the reader at the time of interpreting the presupposing element in the simplified sentences. For example[3], consider:

> Mr. Anthony, who runs an employment agency, decries program trading, but he isn't sure it should be strictly regulated.

The subordinate clause, *but he isn't sure it should be strictly regulated* presupposes the clause *Mr. Anthony decries program trading.* If the sentence is naively simplified to:

> Mr. Anthony decries program trading. Mr. Anthony runs an employment agency. But he isn't sure it should be strictly regulated.

the presupposed element is erroneously changed to *Mr. Anthony runs an employment agency.* Even worse, anaphoric cohesion is also adversely affected, as the pronoun *it* now appears to refer to *an employment agency* rather than to *program trading.*

In chapter 5 on *regeneration*, I describe how various generation issues like sentence ordering, cue-word selection, referring-expression generation and determiner choice can be resolved so as to preserve conjunctive cohesive-relations during syntactic simplification. This can still result in breaking anaphoric cohesive-relations. For example, if the first sentence in the text:

> Dr. Knudson found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent, who had necessarily had the disease. Under a microscope he could actually see that a bit of chromosome 13 was missing.

is simplified as in:

> Dr. Knudson found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent. This parent had necessarily had the disease. Under a microscope **he** could actually see that a bit of chromosome 13 was missing.

then the pronoun *he* in the final sentence is difficult to resolve correctly. My theory of how to detect and correct these breaks in anaphoric cohesion is also detailed in chapter 5. In brief, my approach relies on maintaining a model of discourse (discussed in section 1.6) to detect when anaphoric cohesive-relations are broken. Broken links are fixed by using a pronoun resolution algorithm (section 3.1) to find the correct antecedent and replacing the pronoun with a referring expression (section 5.3).

1.6 Theories of Discourse

As discussed in section 1.5 above, I need to address the issue of preserving conjunctive and anaphoric cohesion when simplifying text. In order to do that, I need to model the discourse structure of the text. I now introduce three theories of discourse that I make extensive use of in this thesis.

[3] Most of the examples in this thesis are taken from the Guardian and the Wall Street Journal.

Grosz and Sidner (1986) distinguishes between three aspects of discourse structure—*linguistic structure*, *intentional structure* and *attentional state*. The linguistic structure of a text comprises its division into units of discourse. The intentional structure comprises the intentions that are the communicative basis of the discourse and the relations between discourse units that help to realise these intentions. The attentional state is a model of the focus of attention during a discourse. I describe two models of attentional state (centering and salience) in sections 1.6.1–1.6.2 and a model of linguistic and intentional structure (Rhetorical Structure Theory) in section 1.6.3.

1.6.1 Centering

The development of centering (Grosz and Sidner, 1986; Grosz et al., 1995) as a model of attentional state has been largely motivated by two factors. The first is the need to formalise a notion of connectedness in text in order to explain why, for example, the discourse in example 1.1 appears intuitively to be more connected and coherent than the discourse in example 1.2[4], despite both discourses containing identical propositional content:

(1.1) a. John went to his favourite music store to buy a piano.
b. He had frequented the store for many years.
c. He was excited that he could finally buy a piano.
d. He arrived just as the store was closing for the day.

(1.2) a. John went to his favourite music store to buy a piano.
b. It was a store John had frequented for many years.
c. He was excited that he could finally buy a piano.
d. It was closing just as John arrived.

The second is to model anaphoric cohesion in text by relating the attentional state to the use of anaphoric expressions to explain why, for example, the use of the pronoun *he* is inappropriate for referring to *Terry* in 1.3(c)[5]:

(1.3) a. Tony was furious at being woken up so early.
b. He told Terry to get lost and hung up.
c. Of course, he hadn't intended to upset Tony.

Centering is a model of the local aspects of attentional state. It does not provide an account of entities that are globally relevant throughout the discourse. In centering theory, the term *center* is used for an entity that links an utterance to other utterances in the same discourse segment. The term *utterance* is used for a sentence in context.

[4] The examples 1.1–1.2 are taken from Grosz et al. (1995).

[5] The example 1.3 is an abbreviated version of an example in Grosz et al. (1995).

Hence, the centers introduced by a sentence are determined not just by that sentence but also by the surrounding context. Every utterance U in a discourse introduces a set of forward-looking centers $C_f(U)$ (that contains all the discourse entities evoked by the utterance U) and exactly one backward-looking center $C_b(U)$.

The set of forward-looking centers $C_f(U)$ is ordered according to the prominence of its member entities in the utterance U. This prominence is generally accepted to be determined by grammatical function, with subjects being ranked higher than objects, which are in turn ranked higher than everything else.

The backward-looking center $C_b(U_n)$ of an utterance U_n is defined as the entity with the highest rank in $C_f(U_{n-1})$ that is evoked in the utterance U_n. The backward-looking center $C_b(U_n)$ thus serves as a link with the preceding utterance U_{n-1}. The ordered set of forward-looking centers $C_f(U_{n-1})$ models the (local aspects of) attentional state after utterance U_{n-1} and contains ranked predictions about what the backward-looking center of the utterance U_n will be. Abrupt changes in the focus of the discourse are reflected in changes in the backward-looking center. A discourse is then modelled by the transitions in the backward-looking centers from sentence to sentence. There are three types of transitions:

1. **Center Continuation**: $C_b(U_n) = C_b(U_{n-1})$ and this entity is the most highly ranked element in $C_f(U_n)$. In this case, this entity is the most likely candidate to be $C_b(U_{n+1})$ as well. This represents the nice simple case when the discourse stays focused on the same entity.

2. **Center Retaining**: $C_b(U_n) = C_b(U_{n-1})$ but this entity is not the most highly ranked element in $C_f(U_n)$. In this case, though C_b is retained from sentence U_{n-1} to U_n, it is likely to change in U_{n+1}. U_n is then likely to be a connecting sentence that evokes the next focus of the discourse.

3. **Center Shift**: $C_b(U_n) \neq C_b(U_{n-1})$. The focus of the discourse has shifted.

Thus, a center-retaining transition followed by a center shift results in a gradual change of focus through a connecting sentence. On the other hand, sequences of center-shifts are likely to make a text disconnected and incoherent. Centering theory postulates two rules that constrain center-realisation:

1. **Rule 1**: If any element in $C_f(U_n)$ is realised by a pronoun in U_{n+1}, then the center $C_b(U_{n+1})$ must also be realised by a pronoun

2. **Rule 2**: Sequences of center continuation are considered less disruptive than sequences of retaining, which are in turn less disruptive than sequences of shifts.

Centering theory then predicts, using rule 2, that example 1.1 is preferable to example 1.2 on the basis that it consist of three continuations, as against three shifts. Using a pronoun for *Terry* in 1.3(c) is unacceptable because it violates rule 1. As *Terry* is a member of $C_f(U_b)$ that is realised as a pronoun in U_c, *Tony*, being $C_b(U_c)$, must also be realised as a pronoun in U_c.

I now turn my attention to the use of centering theory for pronoun resolution. While centering is a useful theory for modelling attentional state and the local aspects of text cohesion, it is not by itself a recipe for pronoun-resolution methods. The first problem with using centering as a theory for pronoun resolution is its non-incrementality. The lists C_f are constructed only at sentence boundaries. This means that intra-sentential pronouns cannot be resolved till the end of the sentence when that list is made available. To overcome this problem, the centering-based pronoun-resolution procedure by Tetreault (1999) maintains an incremental $C_{f-partial}$ list for the sentence under consideration. When a pronoun needs to be resolved, this list is searched before the C_f lists of previous sentences. The Tetreault (1999) algorithm, though conforming to centering theory, does not make use of the distinctive features of the theory. It does not use the backward-looking center, or the three transitions that form the basis of the theory, or even rule 2. It merely searches the ordered C_f lists of previous utterances (starting with the most immediate) for an entity that conforms with any syntax (binding) or agreement constraints on the use of the pronoun. It therefore looks very similar to pronoun-resolution methods based on other models of attentional state like salience (introduced in the next section).

1.6.2 Salience

The salience-based model (Strube, 1998) is more specifically directed towards pronoun-resolution than centering theory is. It does not use the notion of a backward-looking center; rather, it models attentional state by means of a salience list S. This is similar to to the forward-looking centers C_f in centering theory, but differs in important ways. The list S is maintained incrementally, not constructed at sentence boundaries. Further, at any point in the discourse, S can contain all the entities introduced in discourse till that point (though it usually pruned for efficiency reasons). This is a significant difference from the notion of a forward-looking center, where $C_f(U_n)$ only contains entities evoked within U_n, and means that S is capable of modelling a global attentional state as well as a local one. The salience list S is ordered according to the position of an entity in the discourse. Within a sentence, entities are ordered from left to right. Entities in more recent sentences are ranked higher than entities in less recent sentences. Salience-based approaches can therefore be implemented with a fairly shallow level of syntactic processing. The salience model of discourse, however, only addresses the issue of attentional state and does not provide the insights into local coherence that are provided by the centering model.

Interestingly, pronoun-resolution algorithms based on salience were in use well before Strube (1998) proposed salience as an alternative discourse model to centering. In the framework of Lappin and Leass (1994), the discourse model consisted of a set of *co-reference classes*. Each co-reference class corresponded to one entity that had been evoked in the discourse and came with an associated salience value (computed using a salience function, described below). Every entity encountered in the discourse was either added to an existing co-reference class (whose salience value was suitably modified) or used to form a new co-reference class. Pronoun resolution then reduced to selecting the co-reference class with the highest salience value that satisfied all the syntax and agreement constraints.

There are only two differences between the approaches of Lappin and Leass (1994) and Strube (1998). The first is that the set of co-reference classes and associated salience values used by Lappin and Leass (1994) is replaced by an ordered list S in Strube (1998). The second difference is in computing salience values (in Lappin and Leass (1994)) and ordering S (in Strube (1998)). Lappin and Leass (1994) use a salience function that sums the weights associated with any of the features:

Salience Factor	**Weight**
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect Object / Oblique emphasis	40
Head Noun emphasis	80

that are active for a salience class to calculate its salience value, while Strube (1998) calculates order only based on position. The Lappin and Leass (1994) model has the advantage of flexibility as it allows for experimentation in the choice of salience features and weights. I discuss this further in section 3.1.

1.6.3 Rhetorical Structure Theory

Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) is a discourse theory that attempts to model the linguistic and intentional structure of a text. Its starting point is that a coherent[6] text should not have gaps in it. Hence, every text span[7] has a purpose and is related to the rest of the text my means of some relation. In RST, these relations are called *rhetorical relations*. Mann and Thompson (1988) list 23 rhetorical relations that can link text spans. I reproduce an abbreviated list below:

> **Rhetorical Relations**: motivation, antithesis, background, evidence, concession, condition, elaboration, circumstance, restatement, sequence, contrast.

Unlike centering theory, RST does not consider referential relations; rather, it uses rhetorical relations to capture the writer's intentions for using a particular text span.

An important concept in RST is that of nuclearity. For most of the relations, one of the involved text spans (the nucleus) is more important than the other (the satellite). Mann and Thompson (1988) claim that the majority of text spans in naturally occurring text are related to each other by nucleus-satellite relations. Exceptions are called multinuclear relations, for example, *sequence* and *contrast*.

The authors define each relation in terms of constraints on the nucleus and satellite, and the intentions of the writer. For example, the *concession* relation is defined as:

[6] I use the functional definition in which coherence is the phenomenon at the level of interpretation that is analogous to what cohesion is at the level of linguistic structure (Halliday and Hasan, 1976). So it is possible for a text to be coherent even when not cohesive, provided that world knowledge can rule out spurious interpretations. On the other hand, a cohesive text is unlikely to be incoherent, unless it is nonsensical or schizophrenic.

[7] A *text span* is similar to a clause. The notion of a text span has not been formally defined in a manner that is universally accepted, but it is generally agreed that text spans have independent functional integrity. Hence, they are either clauses, or larger units comprising clauses.

The Concession Relation

1. **Constraint on nucleus**: The writer has positive regard for the nucleus.
2. **Constraint on satellite**: The writer is not claiming that the satellite does not hold.
3. **Constraint on both**: The writer acknowledges a potential or apparent incompatibility between the nucleus and satellite. Recognising the compatibility between the two increases the reader's positive regard for the nucleus.
4. **Writer's intentions**: To increase the reader's positive regard for the nucleus.

The definitions of rhetorical relations are therefore purely functional. In particular, the definitions do not address the issue of how these relations are actually signalled in text. In fact, Mann and Thompson (1988) make the claim that rhetorical relations need not be signalled linguistically at all, and that less than half the rhetorical relations in naturally occurring text are actually signalled. When rhetorical relations are signalled linguistically, it is usually by means of cue-words or cue-phrases. For example, the *concession* relation defined above can be signalled by cue-words like *but*, *though*, *however...*

Another major claim of RST is that any discourse can be represented as a rhetorical-structure tree with a unique root that spans the entire text and all the subtrees linked by rhetorical relations or *schemas*. Schemas are like multi-nuclear relations, but each component has a distinct functional label; for example, an *Article* schema may have the components— *Title*, *Author*, *Abstract*, *Section* and *References*.

Figure 1.2[8] shows the rhetorical structure tree for a text containing two sentences:

> The people waiting in line carried a message, a refutation, of claims that the jobless could be employed if only they showed enough moxie. Every rule has exceptions, but the tragic and too-common tableaux of hundreds or even thousands of people snake-lining up for any task with a paycheck illustrates a lack of jobs, not laziness.

This example illustrates some of the arbitrariness of the theory, in that *if only they showed enough moxie* is not considered a text span while *not laziness* is. It also illustrates why rhetorical relations can be difficult to identify computationally. While the *concession* and *antithesis* relations are signalled by the cue-words *but* and *not*, the *evidence* relation is significantly harder to identify (Marcu, 1997; Marcu, 2000). This means that in practice, RST is less useful than centering theory for judging local coherence in text. However, RST remains a useful (and widely used) framework for structuring discourse in natural language generation.

To summarise, centering provides a useful model of local cohesion in text, but is awkward to apply to pronoun-resolution. Salience is specifically directed towards pronoun-resolution and is amenable to shallow implementations, but does not address issues of cohesion. RST does not model issues of reference, but addresses coherence by stipulating that adjacent text spans are connected by rhetorical relations. The functional definition

[8] This example is taken from the Rhetorical Structure Theory web-page maintained by William Mann (http://www.sil.org/∼mannb/rst/).

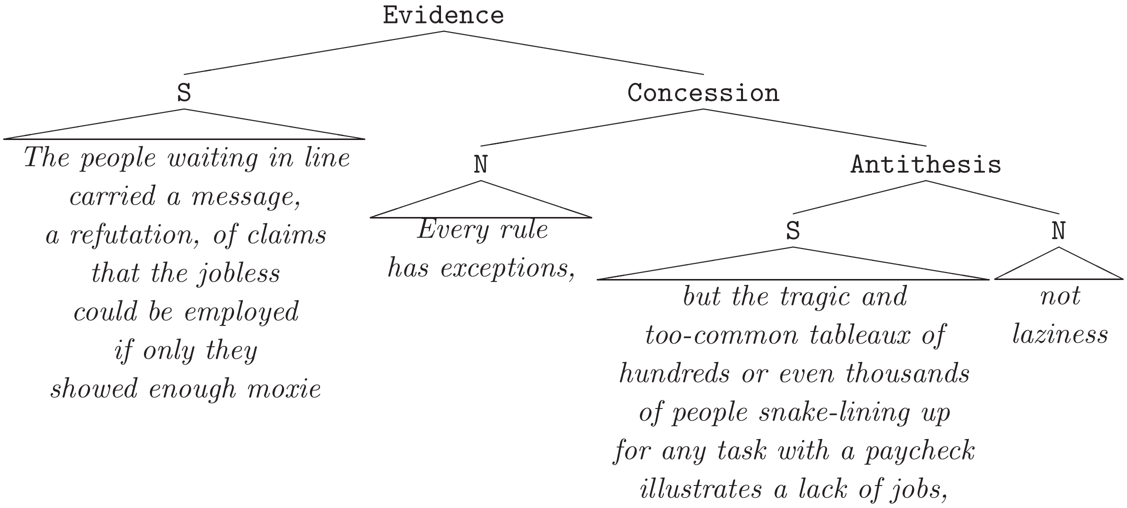

Evidence
S
Concession
The people waiting in line carried a message, a refutation, of claims that the jobless could be employed if only they showed enough moxie
N
Antithesis
Every rule has exceptions,
S
N
but the tragic and too-common tableaux of hundreds or even thousands of people snake-lining up for any task with a paycheck illustrates a lack of jobs,
not laziness

Figure 1.2. A rhetorical structure tree (*N* and *S* refer to the nucleus and satellite of the relation)

of these relations makes them popular in structuring discourse during generation, but at the same time makes RST a difficult theory to use in analysis.

1.7 Some Useful Tools and Resources

I now describe two resources that I use extensively throughout this thesis. The first (WordNet) is a lexical knowledge base and the second (LT TTT) is a tool for segmenting, part-of-speech tagging and chunking text.

1.7.1 WordNet

WordNet (Miller et al., 1993) is an electronic lexical knowledge-base that is inspired by current psycholinguistic theories of human lexical memory. It organises English nouns, verbs, adjectives and adverbs into synonym sets, each representing one underlying lexical concept. These synonym sets are linked by various relations. In this thesis, I only make use of the WordNet classifications for nouns and adjectives.

I use WordNet to obtain animacy information for nouns. I use this information for pronoun-resolution (section 3.1) and relative clause and appositive attachment (sections 3.2 and 3.4). I also use WordNet to find synonyms and antonyms for adjectives. I use this information for generating referring expressions (section 5.3). In this thesis, I use WordNet version 1.7[9].

WordNet organises noun synonym sets hierarchically using the *hyponymy* (X is an instance of Y) relation. Figure 1.3 shows an abbreviated path from a root node to two different senses of the word *dog* in the WordNet hierarchy. Each node is actually a synonym set, but for reasons of clarity, I just show a representative element of the synset; for example, *organism* in the figure represents the synonym set {*organism, being, living thing*}. WordNet relates synonym sets in many ways. Some useful relations between synonym sets that are directly encoded in WordNet are *hypernymy* (X a kind of Y),

[9] Available for download at http://www.cogsci.princeton.edu/∼wn.

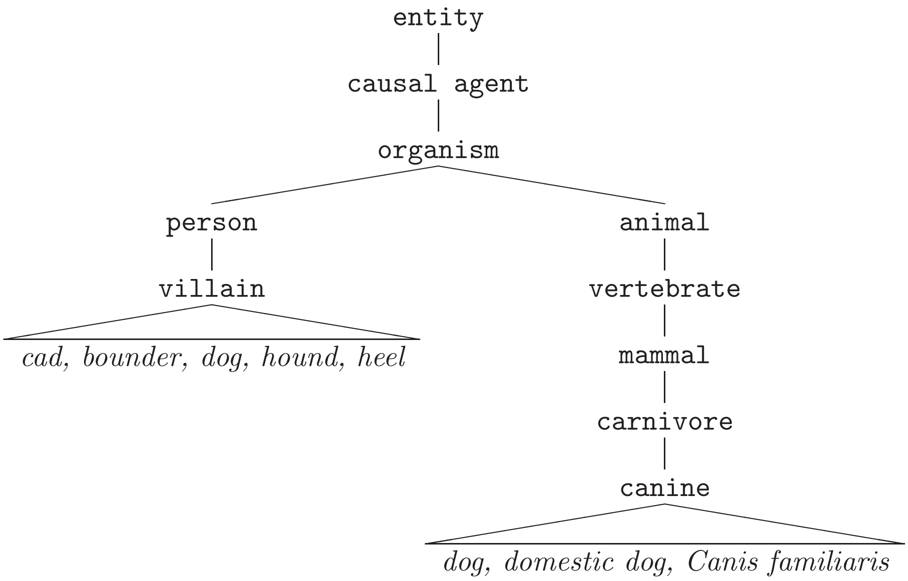


Figure 1.3. Two senses of *dog* in the WordNet hierarchy

hyponymy (X is an instance of Y), *meronymy* (X is a part of Y), *holonymy* (X contains Y) and *coordinateness* (X and Y have the same parent in the hierarchy).

WordNet also organises adjectives as synonym sets. However, with a few exceptions (for example, adjectives representing size, colour or temperature), these synonym sets are not organised hierarchically. Indeed, the most important relation that relates adjective synonym sets is *antonymy*, rather than *hyponymy*. WordNet provides relations for both strict antonymy (for example, *small* vs. *large*) and indirect antonymy (for example, *small* vs. *superior* via *inferior*).

1.7.2 LT TTT

For reasons discussed in chapter 3, I implement my syntactic simplification system without recourse to a parser, using the LT Text Tokenization Toolkit (LT TTT) (Grover et al., 2000; Mikheev et al., 1999) for text segmentation, part-of-speech tagging and noun chunking. I use LT TTT version 2.0[10] in this thesis.

The LT TTT provides a set of tools that can be used to tokenize text by introducing XML mark-up. Text can be processed at either the character level or at the level of XML elements. The toolkit comes with built-in rule sets to mark-up words, sentences and paragraphs as well as to perform basic chunking into noun phrases and verb groups.

LT TTT segments text into sentences using a sentence boundary disambiguator (Mikheev, 1998) that was trained using maximum entropy modelling techniques. The author reports an error rate of 0.8% on the Penn Wall Street Journal Corpus. It then uses the LT POS program (Mikheev, 1997) to assign part-of-speech labels to words in a text. LT POS is a probabilistic part-of-speech tagger based on Hidden Markov Models using Maximum Entropy probability estimators. The tagger has been trained on the Brown Corpus and achieves 96% to 98% accuracy when all the words in the text are found in the lexicon; on unknown words it achieves 88-92% accuracy.

[10] Available for download at http://www.ltg.ed.ac.uk/software/ttt.

The noun chunking is performed using a finite state transducer compiled from a hand-written grammar consisting of around 50 regular-expression rules. This grammar is also written in XML; for example, the top level rule for noun-chunking is:

```
<RULE ‘‘name=AllNounGroup’’ ‘‘type=DISJF’’ >
   <REL ‘‘type=REF’’ ‘‘match=QuantifiedNG’’ ></REL>
   <REL ‘‘type=REF’’ ‘‘match=PossOrBasicNG’’ ></REL>
   <REL ‘‘type=REF’’ ‘‘match=PronounNG’’ ></REL>
</RULE>
```

which states that *AllNounGroup* is the disjunction of *QuantifiedNG* (quantified noun group), *PossOrBasicNG* (possessive or basic noun group) and *PronounNG* (pronoun). These categories are similarly defined as either disjunctions or sequences of other categories. At the bottom level, rules are written in terms of the part-of-speech tags; for example, the definition of *PronounNG* is:

```
<RULE ‘‘name=PronounNG’’>
   <REL ‘‘match=W[C=‘‘((PRP)|(W?DT))$’’] ’’ ></REL>
</RULE>
```

which states that *PronounNG* matches an XML word-element *W* which has part-of-speech attribute *C* that matches the regular expression *((PRP)|(W?DT))$*.

The noun-chunking component of the LT TTT has an accuracy of 89% (on crossing brackets) when evaluated on the Brown Corpus (Grover, personal communication). The noun chunker only identifies elementary noun chunks, not noun phrases. For example, the noun chunks as identified by the LT TTT are enclosed in square brackets in the sentence below:

> [The percentage] of [lung cancer deaths] among [the workers] at [the West Groton], [Mass.], [paper factory] appears to be the highest for [any asbestos workers] studied in [Western] industrialized [countries], [he] said.

1.8 An Outline of this Thesis

I now present an outline of the rest of this thesis. I describe my architecture for text simplification in chapter 2. My three-stage theory of text simplification lends itself easily to a modular architecture with separate modules for *analysis*, *transformation* and *regeneration*. Chapter 2 specifies the natural language processing tasks that need to be performed in each module.

I describe my theories and techniques for performing these tasks in chapters 3 – 5. These chapters describe my implementations of the analysis, transformation and regeneration modules in my architecture and elaborate on my theory of text simplification where required. My focus is on shallow and robust techniques that can be used on chunked text (as described in section 1.7.2) and do not require full parsing.

Chapter 3 on *analysis* describes how I decide clause boundaries and attachment and resolve pronouns reliably without using a parser. I explore a range of techniques, both symbolic and statistical, for tackling these issues.

Chapter 4 on *transformation* describes the transformation rules I use for simplifying text and formalises the order in which these rules should be applied.

Chapter 5 on *regeneration* contains a detailed analysis of the discourse-level issues that arise from sentence-level syntactic transformations. It provides a theory of how to resolve various generation issues like sentence ordering, cue-word selection, referring-expression generation, determiner choice and pronominal use so as to preserve conjunctive and anaphoric cohesive-relations during syntactic simplification. It also describes my algorithms for resolving each of these issues.

My text simplification system addresses a range of NLP problems, in each of the three stages— *analysis*, *transformation* and *regeneration*. I present techniques for deciding clause boundaries and attachment, resolving anaphora, generating referring expressions and preserving discourse structure. I evaluate each technique individually as and when I present it. As an objective evaluation of each technique requires both a suitably marked-up corpus and suitable benchmarks to evaluate against, I have had to use different corpora to evaluate different techniques. I describe the corpora and benchmarks used alongside each evaluation. I also present an evaluation of the composite system on a corpus of newspaper articles in chapter 6.

Chapter 7 concludes with a summary of the main contributions of this thesis and suggestions of avenues for future work.

2 *Architecture*

As described in section 1.5, my theory of text simplification divides the task into three stages— *analysis*, *transformation* and *regeneration*[11]. My architecture uses one module for each of these stages, as shown in the block diagram in figure 2.1. The text is analysed in the *analysis* module and then passed on to the *transformation* module. The transformation module applies rules for syntactic simplification and calls the *regeneration* module to address issues of text cohesion. When no further simplification is possible, the transformation stage outputs the simplified text.

2.1 The Functions of the Three Modules

I now summarise the functions of each of the three modules in my architecture. Then, in section 2.2, I describe the internal representations used by these modules.

2.1.1 Analysis

The analysis module performs various functions. It segments text into sentences. This segmentation is important because my syntactic-simplification rules work at the level of the sentence. It then marks-up syntactic structures that can be simplified in each sentence. This mark-up has two components— clause/appositive identification and clause/appositive attachment. For example, simplifying 2.1(a) to 2.1(b) requires knowledge that the relative clause attaches to *Cathy Tinsall* rather than *South London* and that the relative clause does not end at the first comma, but extends to the end of the sentence.

[11] Parts of this chapter have been published previously in Siddharthan (2002a).

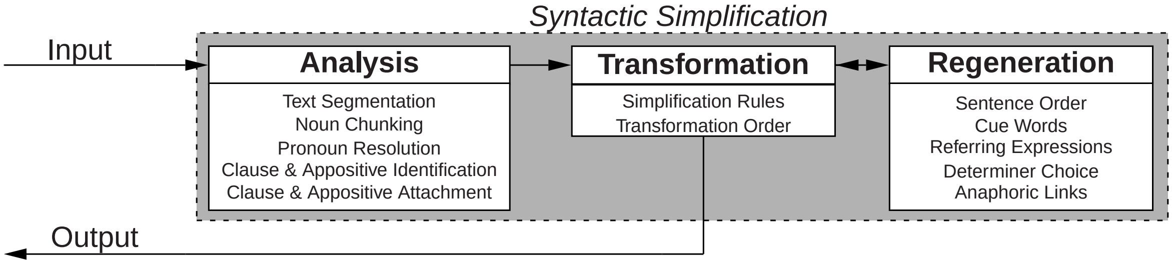


Figure 2.1. An architecture for a text simplification system

(2.1) a. 'The pace of life was slower in those days,' says **51-year-old Cathy Tinsall** from South London, *who had five children, three of them boys.*

b. 'The pace of life was slower in those days,' says 51-year-old Cathy Tinsall from South London. Cathy Tinsall had five children, three of them boys.

The analysis module also includes a pronoun-resolution component that co-refers third-person pronouns with their antecedents. This is for use by the regeneration module rather than the transformation module. If the regeneration module needs to replace a pronoun with a referring expression in order to preserve anaphoric cohesion (as mentioned in section 1.5 and expanded on in chapter 5), it needs to know what the correct antecedent is.

It is worth elaborating on why I include pronoun-resolution in the analysis module rather than the regeneration module in my architecture. As discussed in section 1.5, my regeneration stage maintains a model of discourse in order to detect adverse changes in text cohesion. It might then appear logical to use that discourse model for pronoun-resolution, which would mean resolving pronouns in the regeneration module. The decision to include pronoun-resolution in the analysis stage is a pragmatic one. The rationale is that it is desirable to use the best available pronoun-resolution algorithm to find the correct antecedent for a pronoun. For that reason, it is unnecessarily restrictive to constrain my pronoun-resolution algorithm to use the same discourse model that I use for analysing text cohesion. Also, from an architectural viewpoint, it is important to maintain modularity. My decision allows me the freedom to change to a better pronoun-resolution algorithm without having to reorient my theory (laid out in chapter 5) towards a different discourse model used by that algorithm.

I now provide a specification of the representation that the analysis module needs to output. This is based on the requirements of the transformation and regeneration modules and will be elaborated on in the subsequent sections on those modules. The analysis module can be developed and modified independently of the rest of the system as long as it meets this output specification.

Output Specification for Analysis Stage:

1. The text should be segmented into sentences.
2. Words should be part-of-speech tagged.
3. Elementary noun phrases should be marked-up and annotated with grammatical function information.
4. Boundaries and attachment should be marked-up for the clauses and phrases to be simplified.
5. Pronouns should be co-refered to their antecedents.

2.1.2 Transformation

The transformation stage takes as input a representation that marks the boundaries of the construct to be extracted as well as the noun phrase that the construct attaches to (item 4 in the output specification for the analysis stage). The transformation stage consists of straightforward hand-crafted rules like the following:

$$\texttt{V W}^{n}_{NP}\ \texttt{X [}_{RC}\texttt{RELPR}^{\#n}\ \texttt{Y] Z.}\longrightarrow \begin{array}{ll} (i) & \texttt{V W X Z.}\\ (ii) & \texttt{W Y.}\end{array}$$

This rule states that if, in my analysed text, a relative clause $RELPR\ Y$ attaches to a noun phrase W, then I can extract $W\ Y$ into a new sentence. I use superscript $\#n$ to indicate attachment to the noun phrase with superscript n.

Transformation rules are applied recursively on a sentence until no further simplification is possible. Individual transformation rules introduce constraints on potential sentence orderings. These constraints are resolved in the regeneration stage.

2.1.3 Regeneration

There are many standard generation issues that also crop up when regenerating transformed text. As described in section 1.5, addressing these issues is crucial for preserving the cohesion and meaning of the original text. The regeneration module contains components to perform each of the following tasks:

1. *Introducing Cue Words*
   In order to preserve the rhetorical relations (described in section 1.6.3) that existed between clauses in the original text, it might be necessary to introduce suitable cue words in the simplified sentences.

2. *Deciding Sentence Order*
   When the simplification rule splits a sentence into two, a decision needs to be made on the order in which to output the simplified sentences.

3. *Generating Referring Expressions*
   When simplification rules duplicate noun phrases, a referring expression needs to be used the second time as reproducing the whole noun phrase can make the text stilted.

4. *Selecting Determiners*
   When simplification rules duplicate noun phrases, a decision must be made on what determiners to use.

5. *Preserving Anaphoric Links*
   Splitting sentences or changing their voice can change the grammatical function of noun phrases and alter the order in which they are introduced into the discourse. This can affect the reader's ability to correctly resolve pronouns further on in the text. The regeneration module requires a component that detects and fixes broken anaphoric links.

2.2 Internal Representations

My system uses *XML* (eXtensible Mark-up Language) for its internal representations. I now use an example to show the internal representations at each stage of processing. Consider the following plain text sentence that is input to my analysis stage:

> The Soviets, who normally have few clients other than the state, will get "exposure to a market system," he says.

The output of my analysis stage is:

```
<s1> <np> <dt> The </dt> <nnps> Soviets </nnps> <index> 24 </index>
<grs/> </np> <,> , </,> <simp_nonrest-cl6> <relpr> who </relpr> <index>
25 </index> <coref> 24 </coref> <rb> normally </rb> <vbp> have </vbp>
<np> <jj> few </jj> <nns> clients </nns> <index> 26 </index> <grd/>
</np> <jj> other </jj> <in> than </in> <np> <dt> the </dt> <nn>
state </nn> <index> 27 </index> <gro/> </np> </simp_nonrest-cl6>
<,> , </,> <md> will </md> <vb> get </vb> <“> “ </“> <np> <nn>
exposure </nn> <index> 28 </index> <grd/> </np> <to> to </to>
<np> <dt> a </dt> <nn> market </nn> <nn> system </nn> <index>
29 </index> <gri/> </np> <,> , </,> <sym> ” </sym> <np> <prp> he
</prp> <index> 30 </index> <grs/> <coref> 8 </coref> </np> <vbz>
says </vbz> <.> . </.> </s1>
```

Words are enclosed in POS tags; for example *<nnps> Soviets </nnps>* is a plural proper noun. Noun chunks are enclosed in *<np>...</np>* tags. Sentences are enclosed in *<s1>...</s1>* tags. All these tags are introduced by the LT TTT (described in section 1.7.2). My analysis module introduces further mark-up. Noun chunks are numbered (there is an *<index> int </index>* within each *<np>...</np>* construct) and pronouns are co-referenced using an additional *<coref> int </coref>*. The tags *<grs/>*, *<grd/>*, *<gri/>* and *<gro/>* mark noun chunks with their grammatical relations (subject, direct object, indirect object and oblique). All markup tags for clauses to be simplified start with *simp_*, followed by an identifier for the construct and a unique integer identifier (so that the correct end tag can be found). In the example above, the nonrestrictive relative clause is enclosed in *<simp_nonrest-cl6>...</simp_nonrest-cl6>*. This is the input representation for the transformation stage.

The transformation stage then splits the sentence into two, stripping out the clause marker tags, introducing sentence marker tags and if necessary changing grammatical relation tags to give:

```
<s1> <np> <dt> The </dt> <nnps> Soviets </nnps> <index> 24 </index>
<grs/> </np> <md> will </md> <vb> get </vb> <“> “ </“> <np>
<nn> exposure </nn> <index> 28 </index> <grd/> </np> <to> to
</to> <np> <dt> a </dt> <nn> market </nn> <nn> system </nn>
<index> 29 </index> <gri/> </np> <,> , </,> <sym> ” </sym> <np>
<prp> he </prp> <index> 30 </index> <grs/> <coref> 8 </coref> </np>
<vbz> says </vbz> <.> . </.> </s1>
```

and:

```
<s1> <np> <dt> The </dt> <nnps> Soviets </nnps> <index> 24 </index>
<grs/> </np> <rb> normally </rb> <vbp> have </vbp> <np> <jj> few
</jj> <nns> clients </nns> <index> 26 </index> <grd/> </np> <jj>
other </jj> <in> than </in> <np> <dt> the </dt> <nn> state </nn>
<index> 27 </index> <gro/> </np> <.> . </.> </s1>
```

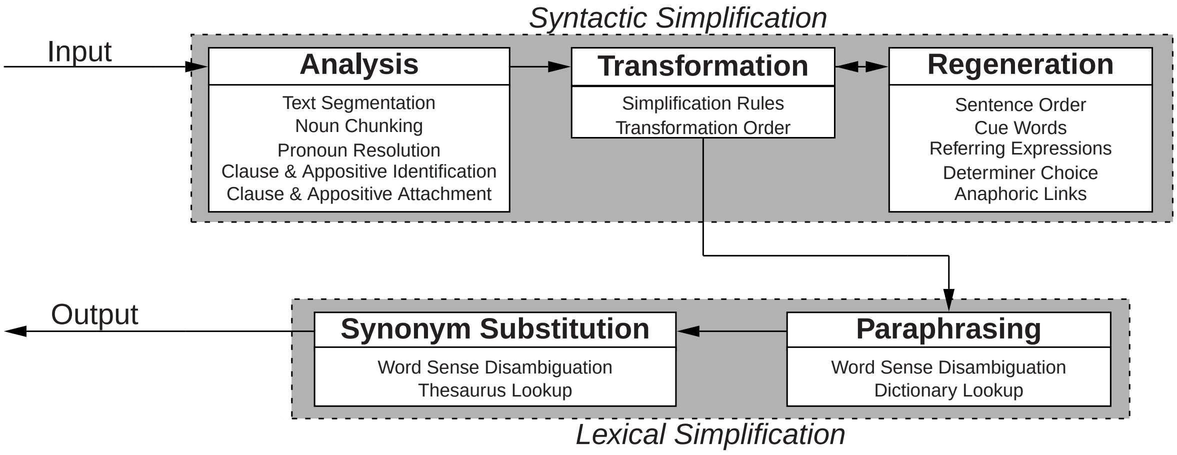


Figure 2.2. Incorporating lexical simplification into my architecture

The regeneration stage then performs all its tasks (in this case it only needs to perform sentence ordering) and strips all the tags introduced by the analysis stage to give:

> The Soviets will get “exposure to a market system,” he says. The Soviets normally have few clients other than the state.

XML, though useful as a representation language, is hard to read when printed. for the rest of this thesis, my examples will uses a simplified notation that is easy to read and only presents markup that is relevant to the example. For example, the output of the analysis stage above will be presented as:

> [The Soviets]^{1}, [$_{RC}$ [who]$^{2\#1}$ normally have few clients other than the state], will get “exposure to a market system,” he says.

where #1 represents a co-reference with the noun chunk *1* (*The Soviets*).

2.3 Extending my Architecture

The architecture for text simplification (figure 2.1) that I have presented in this chapter only deals with syntactic text-simplification. It is however easily extensible to include lexical simplification. Lexical simplification can involve paraphrasing words (especially verbs) with their dictionary definitions (Kaji et al., 2002) or replacing words with simpler synonyms (Devlin, 1999). This process can be carried out after syntactic simplification, as shown in figure 2.2. I do not, however, offer a treatment of lexical simplification in this thesis.

2.4 Comparing NLP Architectures

2.4.1 *Text Summarisation*

It is worth contrasting text simplification with the better established NLP task of automatic text summarisation. In some sense, the two tasks have diametrically opposed goals. Text simplification tries to preserve information content while reducing grammatical complexity. Summarisation aims to drastically reduce information content, retaining only the most significant information. Further, summarisation often results in an increase in grammatical complexity, as information is packed into sentences in the summary (McKeown et al., 1995).

Perhaps paradoxically, the two tasks also share a lot in common. Both summarisation and simplification involve transforming a text in some way. Further, for both tasks, the source and target texts are in the same language. I might therefore expect the architecture of a text simplification system to closely resemble the architectures used by text summarisation systems.

As described in section 1.4.1, early work on text simplification used an architecture with two stages— *analysis* and *transformation*. This corresponds to the architecture used by early summarisation systems (Luhn, 1958; Edmundson, 1964; Pollock and Zamora, 1975), where the analysis stage involved shallow surface level techniques like term frequency, cue phrases and sentence location and the transformation stage involved simple sentence extraction.

However, more recently, there has been a realisation that summarisation systems require three-stage architectures. Different authors refer to the the different stages differently; for example, *identification*, *interpretation* and *generation* (Hovy and Lin, 1999), *interpretation*, *transformation* and *generation* (Sparck Jones, 1999) and *analysis*, *transformation* and *synthesis* (Mani and Maybury, 1999). In the SUMMARIST system, Hovy and Lin (1999) used the middle stage to transform sentences using topic generalisations; for example, *John bought some vegetables, fruit, bread and milk*⟶*John bought some groceries*. They used WordNet as their knowledge source. However, despite their advocating full sentence planning in the generation stage, the SUMMARIST system is implemented to perform only simple sentence extraction. Similarly, Sparck Jones (1999) and Mani and Maybury (1999) argue in favour of a three-stage architecture, but do not offer an implementation of the same.

An example of a summarisation system with an involved generation stage that is in active use is MultiGen (Barzilay et al., 1999; Barzilay, 2003), which is part of Columbia University's NewsBlaster system for multi-document summarisation. This system goes beyond merely extracting sentences from news sources and stringing them together in a summary and performs *information fusion* across sentences. MultiGen clusters sentences (extracted from multiple news sources) that contain related information and attempts to fuse them into one sentence that contains the information common to the cluster. It does this by identifying a representative sentence and modifying it (by deletions and insertions) to include only information that is sufficiently common among the sentences in the cluster. Identifying information that is common across sentences extracted from different news reports requires an ability to recognise paraphrases; Barzilay (2003) describes an unsupervised approach for paraphrase acquisition. Deleting non-central information from a sentence is a similar task to that of sentence shortening (cf. section 1.3.2). Inserting textual units into a skeleton sentence is a sentence aggregation task. In addition to these, MultiGen also attempts to address issues of cohesion in the summary by ordering the sentences chronologically (chronological information is obtained from the date and time that a report is posted), while minimising the number of topic shifts (the level of a topic shift between two sentences is calculated by an information theoretic comparison of the sentence clusters that they represent). It is thus clear that the trend in summarisation systems is to move towards more elaborate transformation and regeneration stages. As I emphasise through this thesis, text simplification also requires a third stage (*regeneration*), to deal with the discourse-level implications of applying sentence-level syntactic transforms to text.

Text summarisation and simplification face similar problems in the analysis stage. While deep analysis would no doubt help both applications, limitations in applying current deep parsing technology to open domains result in the popularity of approaches based on shallow (though not surface) techniques. Both applications involve transforming texts at the scale of the sentences, clauses and phrases. These transforms lead to similar discourse level issues of text cohesion and coherence. In that sense, the regeneration stages of both applications can be expected to have significant overlap and I expect the approaches I present in this thesis for preserving cohesiveness of simplified text to be applicable to summarisation as well.

2.4.2 Natural Language Generation

Interestingly, early generation systems also used a two stage architecture— *document planning* and *linguistic realisation* (Thompson, 1977). The first stage handled issues like representing information in a domain, deciding what to say (content selection) and structuring the information to present according to a discourse model. The second stage converted intermediate abstract text specifications into surface text, handling issues of morphology and grammar.

In the 1990s, it started being recognised that many tasks that are important for generation do not categorise easily as either document planning or surface realisation. In some sense, most of these tasks lie in between the two stages. Recent natural language generation systems therefore tend to have a third stage, usually referred to as *microplanning*, that lies between the *document planning* and *surface realisation* stages (Reiter and Dale, 2000).

Interestingly, while document planning and surface realisation are irrelevant for syntactic simplification (the input to a text simplification system is a natural language text in which content has already been selected and structured, while its intermediate text specification is composed of sentences that have already been realised), the regeneration stage in my architecture for syntactic simplification corresponds closely to the microplanning stage of generation systems. Reiter and Dale (2000) include all my regeneration tasks (cue-word selection, sentence ordering, referring expression generation, determiner choice and pronominalisation) as tasks for the microplanner, arguing that all of them require knowledge not available at the document planning stage, but are not directly related with syntax and morphology driven surface realisation either. The aim of the microplanner is to generate coherent discourse, rather than select content or decide syntax, and this aim coincides with the aim of my regeneration component. Therefore, it is hardly surprising that my regeneration stage corresponds so closely to microplanners in generation systems.

3 *Analysis*

The purpose of the analysis module is to take in text and convert it into a representation that the transformation and regeneration modules can work with; hence the functions of the analysis module are derived from the requirements of these succeeding modules[12]. In section 2.1.1, I provided a specification of the tasks that the analysis module is required to perform. To recap, these tasks were sentence boundary detection, part-of-speech tagging, noun chunking, (limited) grammatical-function determination, clause and appositive identification and attachment (as needed for simplification rules) and third-person pronoun resolution.

As with any natural language processing application, a decision needs to be made on what depth of analysis is required. A parser could, in theory, be used for all the above stated tasks, with the exception of pronoun resolution. However, deeper analyses like full parses are less robust and computationally more expensive than shallower analyses like part-of-speech tagging and noun chunking. Even relatively shallow parsers like Briscoe and Carroll (1995) return full analyses for only 80% of sentences in newspaper text (as reported by the PSET project, section 1.4.2). And unfortunately, since sentences that need simplification tend to cause parsers problems due to their long length and high degree of ambiguity, it is likely that simplification will be useful for many of the sentences that the parsers fail on.

It is therefore worth considering shallower techniques for each of the tasks in my analysis module. In fact, as I demonstrate in this chapter, shallow techniques that are developed for specific tasks can perform as well as or even better than shallow parsers on those tasks. I use the LT Text Tokenization Toolkit (Grover et al., 2000; Mikheev et al., 1999) (described in section 1.7.2) to perform the initial analysis— segmenting text into sentences, annotating words with their part-of-speech tags and marking-up noun chunks. This guarantees an analysis for every sentence in a text with a computational complexity that is roughly linear in sentence length. In this chapter, I detail various techniques for solving the remaining tasks (grammatical-function determination, clause and appositive identification and attachment and third-person pronoun resolution), using part-of-speech tagged and noun-chunked text (with sentence boundaries marked) as a starting point. This shallow approach is feasible because I only need to identify a limited range of grammatical functions and clauses, and do not need the full GRs or full clause identification.

I present my salience-based pronoun-resolution algorithm in section 3.1. This includes a discussion of how the necessary grammatical functions can be extracted from chunked text by pattern matching (section 3.1.2). I present techniques for relative clause attachment in

[12] Parts of this chapter have been published previously in Siddharthan (2002b) and Siddharthan (2003b).

section 3.2 and then show how relative clause boundaries can be determined reliably using very shallow processing in section 3.3. In section 3.4, I show how appositive phrases can be identified and attached in a similar fashion. I describe my treatment of coordination and subordination in section 3.5. I end this chapter with a holistic evaluation of the analysis module in section 3.6.

3.1 Resolving Third-Person Pronouns

Pronoun resolution systems need to take a range of factors, both syntactic and semantic, into account. Most algorithms do this in stages, by first identifying possible antecedents, then applying a set of filters to rule out some of them and finally applying a decision procedure to select one of the remaining candidates. For example, salience-based algorithms first calculate salience scores for potential antecedents based on their syntactic roles and recency, then apply a set of semantic and syntactic filters to rule out potential antecedents and finally attach the pronoun to the most salient remaining potential antecedent.

Anaphora resolution systems based on salience models (Lappin and Leass, 1994; Kennedy and Boguraev, 1996) tend to use shallower syntactic analysis than those based on other discourse models like centering theory (Brennan et al., 1987; Tetreault, 1999); this makes them particularly attractive to me in my research.

There are pronoun resolution systems that do not form an explicit model of discourse. Mitkov (1998) calculated scores for potential antecedents only when resolving a pronoun. Though these scores were similar to salience scores, they were calculated on the fly when required and no discourse model was maintained as such. Hobbs (1986) used an algorithm that considered potential antecedents in a left to right order, starting with the current sentence and then moving back in the discourse one sentence at a time. This resulted in a preference for subjects that was similar to salience based approaches.

Another system that does not use an explicit model of discourse is that of Ge et al. (1998), who collapsed the distinction between *hard* agreement constraints and weaker syntactic criteria and used a probabilistic model to select an antecedent based on features derived from agreement values, grammatical roles, recency and repetition. They used a Bayesian approach to calculate the probability $p(a|p, f_1...f_n)$ that a is the antecedent of a pronoun p given the features f_{1-n}. The features they used were the head constituent above p, the type of the head constituent of a, the syntactic structures in which a and p appear (grammatical function), the distance between a and p, the number of times the referent of a is mentioned and the gender of p and a. Their pronoun resolution algorithm then involved maximising $P(a_i|p, f_{1-n})$ over all potential antecedents a_i.

Ge et al. (1998) used an unsupervised approach to learning gender information. They ran their algorithm without the gender feature on the entire Penn Wall Street Journal Treebank. By counting the number of times a noun was labelled as the antecedent of *he/his/him/himself*, *she/her/herself/hers* and *it/its/itself* by this purely syntactic pronoun-resolution algorithm, they managed to compute $p(m|w_i)$, $p(f|w_i)$ and $p(n|w_i)$ (the probabilities that a word w_i is male, female or neuter) for every word in the Penn Treebank. By bootstrapping, the gender information was used to improve the pronoun resolution algorithm, which was then used to calculate revised gender probabilities for words in the Penn Treebank.

Ge et al. (1998) used a small subset (containing 3975 sentences and 2477 pronouns) of the Penn WSJ Treebank that had been annotated with co-reference information for their experiments. Using 10-way cross-validation, they reported 82.9% of pronouns resoved correctly by their algorithm. Interestingly, they reported that removing the syntax features brought the accuracy down to 43%, while providing perfect gender information improved the accuracy to 89.3%. This suggests that both syntax and gender information are important to resolving pronouns. I will return to this point when describing my approach later in this section.

I use a salience-based pronoun-resolution algorithm in my implementation. This is partly because, as mentioned earlier, these algorithms are amenable to shallow implementations. The other reason is that I have found the salience-based model of attentional state to be useful for resolving not just third-person pronouns but also relative clause and appositive attachment (described later in sections 3.2 and 3.4.3). Unfortunately, the accuracy of pronoun-resolution systems based on salience, though exceeding 80% on restricted genre, appears to plateau at around 60-65% on unrestricted text (Barbu and Mitkov, 2001; Preiss, 2002). It appears that weights for various salience features, trained to give high performance on particular genre, need to be retrained to work on other genre. However, there remains a strong preference for antecedents that are subjects, and to a lesser extent direct objects, across genre. In section 3.1.2, I show how this crucial subject-object distinction can be made reliably using pattern matching on chunked text. This is a level of processing that is even shallower than that used by Kennedy and Boguraev (1996) (who use knowledge of subcategorisation frames of verbs) and guarantees an analysis for every sentence, with a computational complexity that is linear in sentence length.

Anaphora resolution algorithms need to fall back on more elaborate inference mechanisms when salience alone does not return a reliable answer. Unfortunately, knowledge-intensive approaches do not scale up well when attempts are made to apply them to unrestricted domains. I explore various shallow inference procedures that significantly boost results in section 3.1.4. I then describe my corpus in section 3.1.7 and evaluate my algorithm in section 3.1.9. But first, I describe my pronoun-resolution algorithm.

3.1.1 The Algorithm

My approach to third-person pronoun resolution (algorithm 3.1) closely follows other salience-based algorithms like Lappin and Leass (1994) and Kennedy and Boguraev (1996).

Algorithm 3.1 preprocesses the text (step 1) by annotating each noun phrase with information about agreement values and grammatical functions. It then considers each noun phrase from left to right, forming a new co-reference class for non-pronominal noun phrases (step 2(a)) and adding pronouns to existing co-reference classes (step 2(b)). At sentence boundaries, the algorithm halves the salience of each co-reference class (step 2(c)).

Algorithm 3.1 Resolving third-person pronouns

Resolve third-person pronouns

1. Identify all elementary NPs in the discourse window and associate the following features with them:
   - *type* : pronoun / common-noun / proper-noun
   - *agreement* : number, person, gender, animacy
   - *gfun* : subject / direct object / indirect object / oblique
2. Move through the discourse window from left to right. At each:
   (a) non-pronominal noun phrase, form a new co-reference class and initialise its salience value.
   (b) third-person pronoun, add it to the co-reference class with the highest salience value that satisfies all agreement and syntax restrictions. Update the salience value of this co-reference class.
   (c) sentence boundary, halve the salience value of each co-reference class.

3.1.2 Extracting GRs by Pattern Matching

Grammatical function is an important determinant of salience. As anaphora resolution algorithms have a strong subject preference, it is important that I am able to reliably differentiate between subjects and objects.

While most implementations of pronoun-resolution algorithms use some form of parser or information about subcategorisation frames of verbs to decide grammatical function, I do this using only pattern matching on noun-chunked text. I use an ordered sequence of five simple pattern matching rules to decide the grammatical function of noun chunks (these are the inner-most NPs, as marked-up by the LT TTT). In the following patterns, the superscript of noun chunk NP$_i$ gives its grammatical function:

1. Prep NP$_i^{obliq}$
2. NP$_i^{subj}$ [“, [^Verb]+,” | “Prep NP”]* Verb
3. Verb NP$_i^{dobj}$
4. Verb [NP]+ NP$_i^{iobj}$
5. Sent_Marker [^NP]* NP$_i^{subj}$

The first pattern (gfun=*oblique*) looks back for a preposition. The second (gfun=*subject*) looks ahead for a verb, jumping over appositives and prepositional phrases. This pattern will, for example, identify *Bailey Controls* as a subject in:

> [**Bailey Controls**]^{subj}, [based] in [Wickliffe], [Ohio], **makes**^{verb} [computerized industrial controls systems].

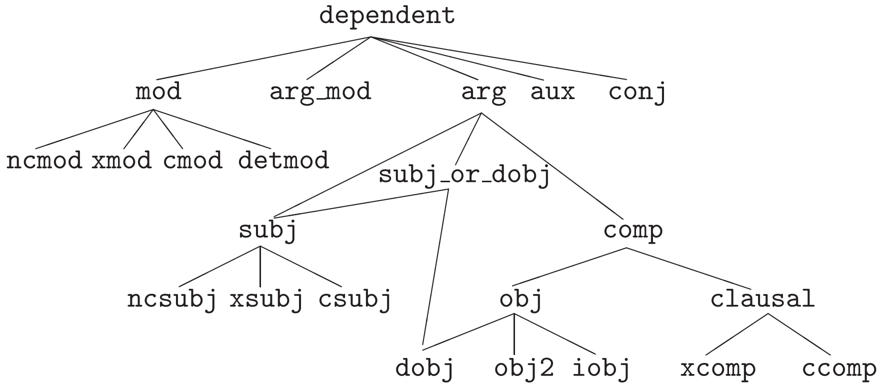


Figure 3.1. Grammatical relation hierarchy (from Briscoe et al. (2002))

The third (gfun=*direct obj*) and fourth (gfun=*indirect obj*) patterns look back for a verb. The fifth pattern marks the first noun phrase in the sentence as a subject. These patterns are applied strictly in order; for example, in:

> After the game, there were celebrations everywhere.

game has *gfun=obliq* because it matches pattern 1, even though it also matches pattern 5.

Preiss (2002) evaluated four parsers (Briscoe and Carroll (1993), Charniak (2000) and two versions of Collins (1997)) using the evaluation corpus for grammatical relations (Carroll et al., 1999a; Briscoe et al., 2002)[13]. The GR hierarchy is shown in figure 3.1 and an example from the evaluation corpus in figure 3.2. Preiss (2002) evaluated the parsers using three GRs—*subj*, *dobj* and *iobj*. As these are the relations that I am interested in, I tried to evaluate my approach using the same evaluation corpus, so that I could compare the results.

The main issue with performing this evaluation was that my patterns only identify the grammatical function of noun chunks. As figure 3.2 shows, the evaluation corpus consists of grammatical relations between nouns and verbs (for the subject and object relations that I am interested in). To make the evaluation meaningful, I had to modify my approach to recover the verb. I wrote a simple script to generate a grammatical relation from a grammatical function. For noun chunks with *gfun=dobj/iobj*, I extracted the most recent verb. For noun chunks with *gfun=subj*, I searched forwards for the head of the nearest verb group. I then generated GRs in the same format as those in the evaluation corpus.

I compare the performance of my approach with the results reported by Preiss (2002) in table 3.1. My approach resulted in low recall because I generated only one grammatical relation per noun chunk, even though a noun might be related to multiple verbs. My script also frequently found the wrong verb, especially when the grammatical function has been determined by pattern 5 (presented at the start of this section). Therefore, I also used the evaluation corpus to evaluate grammatical function, which is what I require for pronoun resolution. For evaluating grammatical function, I ignored the verb in the GRs and only compared the nouns. For example, if the gold standard contained the GRs

[13] The evaluation corpus for GRs is available at http://www.cogs.susx.ac.uk/lab/nlp/carroll/greval.html

Failure to do this will continue to place a disproportionate burden on Fulton taxpayers.

```
(xcomp to failure do)
(dobj do this)
(ncsubj continue failure)
(xcomp to continue place)
(ncsubj place failure)
(dobj place burden)
(ncmod burden disproportionate)
(iobj on place tax-payer)
(ncmod tax-payer Fulton)
(detmod burden a)
(aux continue will)
```

Figure 3.2. Example sentence and GRs from the Carroll et al. (1999a) evaluation corpus.

(these have been taken from the gold standard for illustration purposes, and do not come from one sentence):

```
(ncsubj elaborate jury)
(ncsubj place failure)
(ncsubj tell he)
(ncsubj provide measure)
(ncsubj enforce measure)
```

and my script generated the GRs:

```
(ncsubj did jury)
(ncsubj place failure)
(ncsubj tell he)
(ncsubj provide measure)
```

the precision and recall for grammatical relation (GR) and grammatical function (GF) extraction would be:

GR: $p = .75$ as three out of four generated relations feature in the gold standard.
$r = .60$ as three out of the five relations in the gold standard appear in the generated output.

GF: $p = 1.00$ as all four nouns in the generated GRs have been correctly identified as subjects.
$r = 1.00$ as all the four nouns in the gold standard have been correctly identified as subjects in the generated text.

Table 3.2 compares the accuracy of my approach for extracting grammatical relations and grammatical functions. Its performance on grammatical function is significantly higher for subjects, while there is very little difference for objects.

My approach identifies the object of any preposition as *oblique*, which results in very low recall for *iobj*. The *iobj* results for my algorithm in tables 3.1 and 3.2 are actually for the conflated *iobj/oblique* class; i.e. noun phrases that match pattern 1 are also labelled as

GR	Metric	B&C	Charniak	Collins 1	Collins 2	Me
subj	precision	.84	.91	.89	.90	**.69**
	recall	.88	.85	.80	.83	**.85**
	F-measure	.86	.88	.84	.86	**.76**
dobj	precision	.86	.82	.83	.83	**.81**
	recall	.84	.67	.62	.55	**.80**
	F-measure	.85	.74	.71	.66	**.80**
iobj	precision	.39	.60	.50	.50	**.15**
	recall	.84	.32	.32	.32	**.89**
	F-measure	.53	.41	.39	.39	**.26**

Table 3.1. Evaluation of grammatical relation extraction

Evaluation Criteria	**subj**			**dobj**			**iobj**		
	p	**r**	**f**	**p**	**r**	**f**	**p**	**r**	**f**
Grammatical Relations	.69	.85	.76	.81	.80	.80	.15	.89	.26
Grammatical Function	.90	.95	.92	.82	.81	.81	.16	.93	.27

Table 3.2. Evaluation of grammatical function extraction

iobj. This results in high recall and low precision. The inability to differentiate *iobj*s from *oblique* references is not a problem for pronoun resolution as the Lappin and Leass (1994) salience function uses the same weights for oblique and indirect object emphasis.

On the other hand, an important class of errors (accounting for 20% of the mislabelled *dobj*s) my algorithm makes is that of labelling temporal adjuncts as objects; for example, in *The judge said Friday that...* I take corrective measures for this in my pronoun resolution algorithm, by reducing the salience of hyponyms of the WordNet classes *time period* and *time unit* that appear in the object position.

My results indicate that noun chunks can be classified as subjects and direct objects reliably without using a parser. This is significant, because my approach guarantees an analysis for every sentence, with a complexity that is linear in sentence length.

3.1.3 *Agreement Features*

I use the four standard agreement features, for *number*, *person*, *gender* and *animacy*. I implement the features as lists of allowed values:

1. *number* = (s)ingular, (p)lural
2. *person* = (f)irst, (s)econd, (t)hird
3. *gender* = (m)ale, (f)emale, (n)euter
4. *animacy* = (a)nimate, (i)nanimate

This allows me to under-specify features when I have inadequate information. Having separate *animacy* and *gender* features allows me to handle companies and animals in an elegant way. For a company, I set *gender*=$\{n\}$ and *animacy*=$\{a\}$. For an animal, I set *gender*=$\{m/f,n\}$ and *animacy*=$\{a\}$. Then, for example, the pronoun *it* can refer to something with *gender*=$\{n\}$ and *animacy*=$\{a\}$ (like a company or animal) or something with *animacy*=$\{i\}$. However, *he* can only refer to something with *gender*=$\{m\}$ and *animacy*=$\{a\}$ (an animal but not a company).

I also implement an additional *speaker-quote* agreement feature. This enforces two restrictions: firstly, third person pronouns within quotes cannot co-refer with the speaker of the quote and secondly, pronouns that are speakers of quotes cannot co-refer with noun phrases (apart from first person pronouns) within the quote.

3.1.4 Inferring Agreement Values

Of the four standard agreement features — *number*, *person*, *gender* and *animacy*, values for the first two are available from the POS tagger; however, the tagger does not provide gender and animacy information. To get the most out of my agreement filters, I need to infer as much agreement information as possible. Ge et al. (1998) present an unsupervised approach to learning gender information from a corpus. I take an alternative approach to the problem. In edited text, animacy and gender information for a potential antecedent is usually available in some form elsewhere in the text, usually in other references to the same referent. I try and retrieve this information using shallow inference mechanisms. I run through the set of noun phrases in iterations that:

1. Look for keywords in the NP.
2. Try to co-refer the NP with another NP.
3. Collect information about the head noun in WordNet.
4. Infer from appositives and existential constructs.
5. Make use of any reliable subcategorisation frames for the verb.

In each iteration, I only consider noun phrases for which we are still looking for some agreement information (*animacy* or *gender*).

In the first iteration, I look for keywords in an NP; for example, key words like *Inc.*, *Lmt.*, *PLC.* and *Corp.* suggest that the noun phrase is a company (*gender*=$\{n\}$ and *animacy*=$\{a\}$) and titles like *Mrs.* and *Ms.* suggest that the noun phrase is a female person (*gender*=$\{f\}$ and *animacy*=$\{a\}$). I use the following list of keywords:

> Inc, Ltd, Co, Corp, PLC, Mr, Lord, Earl, Duke, King, Emperor, Sir, Rev, Mrs, Ms, Miss, Lady, Queen, Empress, Duchess, Dr, Prof, Minister, Secretary, President

In the second iteration, I try and co-refer an NP with an NP for which I have the required information. For example, consider the text:

> **Pierre Vinken**^{x}, 61 years old, will join the board as a nonexecutive director Nov. 29. **Mr. Vinken**^{y} is chairman of Elsevier N.V., the Dutch publishing group.

I can find agreement values for x if I can co-refer it to y, which the first iteration has dealt with. I try and perform this co-reference operation in two steps, that look for people and companies respectively. To check if an unrecognised (animacy or gender flag not set) noun phrase is a person, I search forwards for a noun phrase with the same head noun that has already been recognised as a person. This follows the intuition that if a noun phrase $X_1 ... X_n\ Y$ is a person, it is likely that a reference further in the discourse is of the form *Title* Y (for example, *Mrs* Y). To check if an unrecognised noun phrase is a company, I search backwards. This follows the intuition that a company $X_1 ... X_n$ might have been introduced into the discourse as $X_1 ... X_n\ Y$, where Y is an acronym like *co, corp, inc, ltd, plc* or even a hyponym of the WordNet class *group/organisation* like *Association, Institute, Company, University, School...*

The first two iterations largely deal with proper nouns, a particularly troublesome class. The third iteration deals with common nouns and involves a look-up of the head noun in WordNet. If the head noun is a hypernym of *human*, *animal* or *organisation* I set *animacy*=$\{a\}$, otherwise I set *animacy*=$\{i\}$. Gender information is sometimes available for humans in WordNet; for example if the head noun is *son*, *woman*, *widow* or *spinster*. WordNet also recognises some place names, particularly countries and cities.

The fourth iteration makes use of information contained in appositives and copula constructs; for example, consider the examples:

> **J.P. Bolduc**^{x}, **vice chairman**^{y} of W.R. Grace Co., was elected a director.
>
> **Finmeccanica**^{x} is **an Italian state-owned holding company**^{y} with interests in the mechanical engineering industry.

I assign *animacy*=$\{a\}$ to x using the WordNet class of the head noun of y (*chairman* and *company*). I also set *gender*=$\{n\}$ for *Finmeccanica* and rule out *gender*=$\{n\}$ for *J.P. Bolduc*.

The fifth iteration makes use of any reliable subcategorisation frames for verbs. For example, the subject of verbs like *said*, *reported*, *stated* are assigned *animacy*=$\{a\}$.

3.1.5 Syntax Filters

Syntactic filters are required to rule out antecedents that violate binding constraints. I use a fairly simple syntax filter for reflexive pronouns (pronouns ending in *self* or *selves*, for example, *themselves*). This filter marks the region of the sentence between the reflexive pronoun and the most recent subject and ensures that the last member of the co-reference class lies within this marked region. For example, in:

> It was seven o'clock of a very warm evening in the Seeonee hills when *Father Wolf woke up from his day's rest, scratched* **himself**, yawned, and spread out his paws one after the other to get rid of the sleepy feeling in their tips.

the antecedent of *himself* is constrained to lie in the italicised region.

It is trickier to define the binding constraints on personal pronouns (like *they*, *she* or *it*) . For example, it is acceptable for *him* to co-refer with *John* in:

> John slammed the door behind him.

Salience Factor	**L&L Weight**
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect Object / Oblique emphasis	40
Head Noun emphasis	80

Table 3.3. Salience factors and weights (Lappin and Leass, 1994)

but not in:

> John slammed the door on him.

I use a relaxed filter that does not rule out personal pronouns with *gfun=iobj/oblique* co-referring with the subject. The filter does however prevent them co-referring with other objects of the same verb. For example, in:

> The car had a trailer behind it.

it can refer to *car*, but not *trailer*. My syntactic filter also prevents personal pronouns with *gfun=dobj* co-referring with the subject of the same verb (implemented by finding the most recent noun phrase with *gfun=subj*). So, for example, in:

> The grease on the chain protects it from rust.

it can co-refer with *chain* (*gfun=oblique*), but not with *grease* (*gfun=subj*).

I do not place any binding constraints on adjectival (possessive) pronouns like *his* or *their*.

3.1.6 Salience

I use the following Lappin and Leass (1994) salience features shown in figure 3.3. I also consider *possessives*, giving them a weight equal to the weight of their enclosing NP minus ten. The additional features I consider are the number of members in the co-reference class, the WordNet category of the co-reference class and noun phrase recency (distance of the potential antecedent measured in noun phrases).

3.1.7 The Corpus

Due to the lack of a standardised evaluation corpus for pronoun resolution, I have constructed an annotated corpus[14], the contents of which are described in table 3.4. The training and test corpora contain some genres in common (articles from the news, sports and guest column sections of one British and one American daily). The literature component of the training corpus consists of Beatrix Potter, H.H. Munro, Rudyard Kipling

[14] The corpus has been annotated by me, due to various impracticalities of independent annotation. This appears to be the standard procedure in the pronoun-resolution field, and I am unaware of any independently annotated corpora for pronoun resolution.

and Anna Sewell. The literature component of the test corpus consists of Aesop, Lewis Carroll and Agatha Christie. In addition, I have included some genre in the test set that I have not trained on, specifically travelogues (from the Lonely Planet guide) and medical articles.

I expect that this corpus will not overlap with corpora traditionally used in NLP that algorithms might have been trained on, and hence can be useful to other researchers as an independent evaluation corpus. My annotation marks sentences and noun phrases and assigns each noun phrase an index and an optional co-reference index; for example, in:

> (S1 (NP Mr Gilchrist 93) denied (NP-PRP he 94#93) was scare-mongering.
>),

the pronoun *he* has index 94 and co-refers with the noun phrase with index 93.

Pronouns in the corpus are co-referenced with the most recent antecedent. However, earlier antecedents can be recovered for evaluation purposes by following the co-reference chains backward. Pronouns with no antecedent in the discourse are given the co-reference index *#-1*. Plural pronouns that have more than one noun phrase as antecedents are, for the moment, given the co-reference index *#-2*. In future, they could be dealt with using multiple #s.

3.1.8 Methodology

For an evaluation to be meaningful, it is essential that the test data is unseen until the training has been completed. To ensure this, I constructed and annotated my test corpus after I had finished training my algorithm.

I used the training phase to determine the weights for the salience features, as well as to decide the number of WordNet senses to consider and the order in which to use my inference rules. As my aim was to build a genre-independent system, I needed to make sure I did not over-train on my data. I did this by trying to ensure that the training improved results on all the training genre individually, not just the whole corpus collectively. I used eight-fold cross-validation in the training phase; i.e. I trained on seven of my training genre and tested on the remaining one, using each of the eight genre for testing once.

As one of my aims was to observe how different parameters affected pronoun-resolution, I trained the algorithm by perturbing the parameters (salience weights, number of WordNet senses, and the order in which to use inference rules) by hand, in many iterations, till I achieved a configuration I was happy with.

I found that altering the original Lappin and Leass (1994) weights (figure 3.3) in different ways gave improved performance on some genre, but also resulted in worse performance on other genre. For genre-independent performance, the exact salience weights were not significant, as long as there was a strong subject preference. I therefore stuck with the original Lappin and Leass (1994) weights.

As I did not perform word sense disambiguation before looking up WordNet for animacy information, I had to decide how many senses to consider in WordNet. I found that considering only the first sense in WordNet gave poor results as it frequently provided animacy information for the wrong sense. On the other hand, considering three or more

Genre / Corpus	**Training Set**		**Test Set**	
/ Pronoun Type	**3^{rd} Person**	**Relative**	**3^{rd} Person**	**Relative**
Guardian News	93	33	81	24
Guardian Sports	99	25	105	22
Guardian Opinion	93	24	88	20
New York Times News	117	35	122	41
New York Times Sports	94	15	93	28
New York Times Opinion	92	25	111	35
Literature	231	33	216	11
Computer Manuals	89	42	-	-
Lonely Planet Travelogues	-	-	93	27
Medical Articles	-	-	70	23
Total	**908**	**230**	**979**	**231**

Table 3.4. Number of 3^{rd} person and relative pronouns in my corpus

senses was futile, as they assigned all possible animacy values to most nouns. I found that the optimal results were therefore obtained when considering only the first two senses in WordNet.

The optimal ordering for the inference rules is the one presented in section 3.1.4.

I now discuss two different evaluation measures for my training phase. My *gold standard* is marked-up with chains of co-references and I have two options. Suppose my algorithm has resolved the pronouns as below:

> Although Hindley^{1}'s own plans are still in place, police sources say they may have to be revised. "There will be no big send-off, " said one officer^{2}. Feelings about her$^{3\#2}$ still run very high so all arrangements have to be carefully worked out. Just 12 people had been invited to attend the service including her$^{4\#3}$ mother.

I could treat the pronoun *her*$^{4\#3}$ as correctly resolved as it co-refers correctly with *her*^{3}. As salience decreases very fast with distance, the salience of a class tends to be dictated by its most recent member. By verifying only the most recent antecedent, I am evaluating how well salience is working. In future, I refer to the evaluation on the most recent antecedent as *Eval-Salience*.

However, if (as above) the most recent antecedent is a pronoun (*her*$^{3\#2}$), I should chain back all the way to decide if the pronoun has been resolved correctly. In this example my algorithm has resolved *her*$^{4\#3}$ incorrectly to *officer*^{2}. Ultimately, this is what I am interested in, and from now on, I refer to this "absolute" evaluation as *Eval-Absolute. Eval-Salience* is an indicator of how well my algorithm can perform. *Eval-Absolute* measures how well it does. The difference is a measure of how far errors propagate.

It turns out that optimising my algorithm for the *Eval-Salience* measure in the training phase leads to better generalisation and performance in the unseen genre in the training corpus. This is because training on *Eval-Absolute* results in a model that optimises itself for instances of pronouns that (purely by luck) happen to propagate a long distance in the training set. This can happen at the expense of learning patterns that would help it resolve other common instances. Training on *Eval-Salience* results in each pronoun in the training set being treated equally and this leads to a better generalisation ability.

3.1.9 Evaluation

I present my results for third person pronouns in table 3.5. The results for the basic algorithm on my corpus are comparable to those reported by Barbu and Mitkov (2001) and Preiss (2002) for completely different corpora. Barbu and Mitkov (2001) report that the salience-based approach of Kennedy and Boguraev (1996) resolves 61.6% of pronouns in a corpus of computer manuals correctly. On the same corpus, CogNIAC (Baldwin, 1997), used by the PSET project, resolves only 49.7% of pronouns correctly. Preiss (2002) reports that the Lappin and Leass (1994) algorithm resolves between 61% and 64% of pronouns in a subset of the British National Corpus (BNC) correctly, depending on the parser that is used for the analysis.

There is a big improvement in the performance of my algorithm when I use WordNet to obtain agreement values (section 3.1.3). There is a further improvement when I infer agreement values for agreement features (3.1.4) and enforce *speaker-quote* agreement (section 3.1.3). The fact that I report better results on the unseen test corpus suggests that I have not over-trained my system. It is interesting to note that the *Eval-Salience* measure appears to stay reasonably constant across data sets. However, the *Eval-Absolute* measure can vary wildly, from *Eval-Salience* in the best case when errors do not propagate at all, to 20% below *Eval-Salience* when they propagate far. This suggests that traditional evaluations of pronoun resolution algorithms on small corpora can involve a fair bit of luck. The fact that finding the immediate antecedent is easier than finding the absolute antecedent is useful to us. This is because there are applications where I only require the immediate antecedent. I discuss this further in section 5.4 in the chapter on regeneration.

3.1.10 A Note on the Pleonastic 'It'

My pronoun-resolution algorithm does not have a filter for detecting pleonastic or event-denoting occurrences of the pronoun *it*. The results in table 3.5 only consider pronouns that have antecedents in the text. The rationale behind this is that in newspaper text (the genre that I am interested in simplifying), very few instances of *it* have an antecedent (only 15% in the Guardian news reports in my corpora). Most cases are either pleonastic (63% in the Guardian news reports in my corpora; for example, in *it was necessary to give the public a specific warning*) or event-denoting (22% in the Guardian news reports in my corpora; for example, in *she was very adept at telling you what she thought you wanted to hear, if she thought it would bring her closer to release*). Canning (2002) notes that the style book for the Sunderland Echo contains the following advice for using *it* anaphorically: "*Use 'it' sparingly and ensure that it is close to the noun to which it refers. Even then it can produce ambiguity.*" The fact that 85% of *it*s in Guardian news reports do not have antecedents suggests that the Guardian might have a similar editorial policy.

In this thesis I require pronoun resolution in the regeneration module (refer to section 5.4) in order to replace pronouns with their antecedent noun phrases. I do not attempt pronoun replacement for the pronoun *it* due to its predominantly pleonastic use

Genre / Corpus	**Training Corpus**			**Test Corpus**		
/ Algorithm	**Base Algo**	**+ WordNet**	**+ Inference**	**Base Algo**	**+ WordNet**	**+ Inference**
Guardian Opinion	.60 / .65	.79 / .81	.80 / .84	.61 / .73	.69 / .77	.83 / .85
Guardian News	.58 / .64	.77 / .79	.80 / .81	.61 / .69	.56 / .77	.60 / .78
Guardian Sports	.57 / .68	.53 / .74	.80 / .85	.60 / .71	.71 / .79	.84 / .87
NY Times Opinion	.60 / .76	.65 / .77	.81 / .88	.56 / .65	.72 / .79	.85 / .88
NY Times News	.53 / .64	.68 / .77	.82 / .86	.68 / .75	.75 / .79	.84 / .85
NY Times Sports	.70 / .76	.77 / .83	.69 / .75	.62 / .70	.71 / .82	.80 / .84
Literature	.61 / .75	.67 / .80	.73 / .84	.55 / .62	.68 / .71	.74 / .84
Computer Manuals	.66 / .72	.72 / .76	.74 / .78	-	-	-
Travelogues	-	-	-	.66 / .73	.77 / .79	.84 / .87
Medical Articles	-	-	-	.54 / .72	.65 / .83	.89 / .90
Average	**.61 / .71**	**.69 / .79**	**.76 / .82**	**.60 / .70**	**.69 / .79**	**.79 / .85**

Table 3.5. Results for third person pronouns — Accuracy is reported as *Eval-Absolute / Eval-Salience*

(Canning (2002) also does not perform pronoun replacement for *it* in the PSET project, for the same reason). I therefore do not require a pleonastic filter for my application, simplifying news reports. However, there are other genre where the use of *it* is predominantly anaphoric, and a filter for detecting pleonastic occurrences of *it* might be required for performing text simplification on those genre. An implementation of a pleonastic filter is described by Lappin and Leass (1994), who evaluated their pronoun resolution algorithm on one such genre—computer manuals.

3.2 Deciding Relative Clause Attachment

Relative clause attachment is an interesting problem that has traditionally been approached in a parsing framework. However, determining what a relative pronoun refers to is not a problem that can always be solved in a syntactic framework; in particular, parsers like Briscoe and Carroll (1995) now treat non-restrictive relative clauses as text adjuncts, following the analysis in Nunberg (1990). The parsing community has explored probabilistic approaches to structural disambiguation, but the literature has focused on prepositional phrase attachment (Clark and Weir, 2000; Collins and Brooks, 1995; Ratnaparkhi, 1998), rather than relative clause attachment, largely because PP-attachment is a pervasive form of ambiguity, but perhaps also because there exist standard training and test data for evaluating PP-attachment. This leaves the relative clause attachment decisions to anaphora resolution algorithms. However, existing anaphora resolution algorithms do not address relative clause attachment.

In this section I treat relative clause attachment as an anaphora resolution problem and provide a resolution mechanism for relative pronouns based on salience, agreement and syntactic filters. Before describing my approach, I summarise the technique employed by Clark and Weir (2000) for structural disambiguation. Though Clark and Weir (2000) only evaluate their technique on PP-attachment, they claim that it is useful for deciding relative clause attachment as well. They introduce their approach with the example:

> Fred awarded a prize for the dog that ran the fastest.

They argue that the knowledge that *dog*, rather than *prize*, is often the subject of *run* can be used to decide in favour of local attachment in the example above. However, attempts at lexicalising attachment decisions using probabilities of nouns being the subjects of verbs result in models with vast numbers of parameters and the resultant sparse data problems at the training stage. Clark and Weir (2000) describe a method of reducing the number of parameters by calculating probabilities for classes of nouns rather than individual nouns. In the training stage, they pass counts for individual nouns up the WordNet hierarchy. As an example, if the training corpus contains *eat chicken*, the count can be passed up from the word *chicken* to one of its WordNet hypernyms—<*meat*>, <*food*>,...,<*entity*>. The problem then is to work out how far up the WordNet hierarchy the count can be passed. *Eat* <*food*> is a suitable generalisation, but *Eat* <*entity*> is obviously an over-generalisation. Clark and Weir (2000) describe how statistical significance tests can be used to decide the appropriate level of generalisation and demonstrate how class-based statistics can be learnt for structural disambiguation. However, they only evaluate their approach on the PP-attachment problem, so it is unclear how useful it is for resolving

Pronoun /	**Agreement**		
Agreement	**number**	**gender**	**animacy**
who	from following verb	{m,f,n}	{a}
which	from following verb	{n}	{a,i}
that	from following verb	{m,f,n}	{a,i}

Table 3.6. Agreement values for relative pronouns

relative clause attachment. Also, as their model remains lexicalised on the verb, data sparsity issues still arise for infrequent verbs.

I present my approach to relative pronoun resolution below, using the anaphora resolution framework of agreement and syntax filters and salience functions. My approach is also class based, and relies on WordNet classes for nouns. However, it is now lexicalised over the relative pronoun (*who*, *which* or *that*) rather than the verb, which means it is less affected by issues of sparse data.

3.2.1 Agreement Filter

The most important feature for determining relative clause attachment is animacy. I make a distinction between *who* and *which* clauses. According to Quirk et al. (1985), the relative pronoun *who* is used to refer to something with *personality* and *which* to something without. In terms of the WordNet hierarchy (Miller et al., 1993), *who* can only refer to hyponyms of the following classes— *humans*, *groups(organisations)* or *animals*, while *which* cannot refer to *humans*. There are no animacy restrictions on *that*. I encode these restrictions as agreement values for the relative pronouns as shown in table 3.6. These agreement values allow *who* to refer to people, companies and animals, but nothing inanimate; *which* to refer to companies, animals and inanimate objects, but not people; and *that* to refer to any noun phrase. The values for the number-agreement feature are taken from the verb in the relative clause. The part of speech tags *VB* and *VBZ* set *number*={*s*} and the part of speech tag *VBP* sets *number*={*p*}. For all other verbs, the default of *number*={*s,p*} is used.

3.2.2 Syntactic Filter

The antecedent of a relative pronoun is usually only separated from it by prepositional phrases or appositives; for example, in the sentences below:

> One man who is likely to reap the benefits is Vaino Heikkinen^1, aged 67, a farmer in Lieksa, 10km from the Soviet border, who$^{\#1}$ claims a Finnish record for shooting 36 bears since 1948.
>
> ‘The pace of life was slower in those days,’ says 51-year-old Cathy Tinsall^2 from South London, who$^{\#2}$ had five children, three of them boys.

My syntactic filter rules out any potential antecedent that is separated from the relative pronoun by any other category. This filter can be too restrictive. If no antecedent is found,

Pronoun /	**Training Corpus**		**Test Corpus**	
Algorithm	**Baseline***	**Salience**	**Baseline***	**Salience**
who	.82	.98	.87	.98
which	.85	.97	.78	.86
that	.81	.85	.93	.96
Average	**.82**	**.94**	**.86**	**.94**

* Always attach locally

Table 3.7. Accuracy results for salience-based relative pronoun resolution

I do away with the syntactic filter completely and try again.

3.2.3 *Salience*

I use the same salience function as for third person pronouns; however, I weight it according to the relative pronoun and the animacy of the co-reference class under consideration.

For *who*, I increase the salience of potential antecedents that are people (*anim*={*a*} and *gend*={*m,f*}). This is because to refer to a company or animal in a context where a potential antecedent is a person, the author can use *which* instead of *who*. Hence, in an ambiguous situation, *who* is more likely to refer to a person than an organisation or animal.

For *which*, I increase the salience of potential antecedents that are organisations or animals (*anim*={*a*} and *gen*={*n*}). This is a genre-specific weighting and arises because organisations and animals get referred to more often than inanimate nouns in my corpus.

3.2.4 *Evaluation*

I present an evaluation of my salience-based relative-pronoun resolution algorithm (on the corpus described in table 3.4) in table 3.7. I report a 10% improvement over the local attachment baseline.

An analysis of my training corpus showed that 51% of relative clause attachments were unambiguous (in the sense that my syntax filter returned exactly one potential antecedent for the relative pronoun). Therefore, results on only ambiguous cases is ˜88%, compared to the corresponding baseline for the training corpus of ˜64%. My algorithm therefore gives a significant improvement (˜24%) over the baseline for resolving ambiguous cases.

The ambiguous cases fell into two main types. The main cause of ambiguity (accounting for 70% of the ambiguous cases) involved deciding local vs wide attachment when the noun phrase preceding the relative clause has the structure `NP1 Prep NP2`. The second type (accounting for 26% of the ambiguous cases) involved picking the right noun phrase in the presence of appositives. The remaining 4% of ambiguous cases are those for which my syntactic filter ruled out every noun phrase as an antecedent.

0:	Target (wide attachment)	16:	`NP2` is a *person*
1:	Target (local attachment)	17:	`NP2` is a *group*
2:	Restrictive Clause	18:	`NP2` is an *animal*
3:	`NP1` is a *person*	19:	`NP2` is a *possession*
4:	`NP1` is a *group*	20:	`NP2` is an *entity*
5:	`NP1` is an *animal*	21:	`NP2` is an *act*
6:	`NP1` is a *possession*	22:	`NP2` is an *abstraction*
7:	`NP1` is an *entity*	23:	`NP2` has no WordNet class
8:	`NP1` is an *act*	24:	`NP2` is a proper noun
9:	`NP1` is an *abstraction*	25:	`NP2` contains a definite determiner
10:	`NP1` has no WordNet class	26:	`NP2` has no determiner
11:	`NP1` is a proper noun	27:	`Verb` selects for singular subject
12:	`NP1` contains a definite determiner	28:	`Verb` selects for plural subject
13:	`NP1` has no determiner	29:	`NP1` is singular
14:	`Prep` favours local attachment	30:	`NP2` is singular
15:	`Prep` favours wide attachment		

Table 3.8. List of binary features for deciding relative clause attachment

3.2.5 A Machine Learning Approach to Relative Clause Attachment

The use of salience for making decisions on relative clause attachment is a new approach to the problem. It is therefore worth comparing it with the two obvious alternatives—statistical parsing and machine learning. I now describe a machine learning approach to relative clause attachment. I restrict myself to the case of *who* and *which* relative clauses that are preceded by the structure `NP1 Prep NP2`.

I define the binary features in table 3.8 for each instance of a *who* or *which* clause (either restrictive or non-restrictive) that is preceded by the pattern `NP1 Prep NP2`. An example is then a vector of the indexes of the features that are present in any particular sentence. I used the *SNoW* machine learning package (Carlson et al., 1999) to train a network to decide between local(1) and wide(0) attachment using the WINNOW algorithm. Since I required a larger corpus for training my network, I used parse trees from the Penn Treebank for my experiments.

As mentioned earlier, the most important feature for determining relative clause attachment is animacy. Features 3-10 and 16-23 in table 3.8 classify `NP1` and `NP2` according to the WordNet classes of their head nouns.

I included features for prepositions that the network could make use of when `NP1` or `NP2` did not have WordNet classes; proper nouns (that could be people, organisations or locations) are very common as arguments to prepositions. Lexicalisation over prepositions (having the presence/absence of each preposition as a separate feature) was impractical due to data sparsity problems. I therefore assumed that prepositions only influence attachment indirectly, through their preferences for the agency of their arguments.

I classified the subject and object of 15000 occurrences of prepositions in the WSJ Treebank (in any context, not just preceding relative clauses) according to their WordNet classes. I introduced two features (14 and 15) for prepositions.

Prep	$\mathbf{P}_{who}$	$\mathbf{P}_{which}$	**Prep**	$\mathbf{P}_{who}$	$\mathbf{P}_{which}$	**Prep**	$\mathbf{P}_{who}$	$\mathbf{P}_{which}$
about	.58	.43	against	.53	.47	among	.62	.42
as	.57	.43	at	.46	.57	before	.51	.52
between	.75	.41	by	.63	.43	during	.44	.56
from	.62	.53	for	.55	.50	in	.52	.52
like	.39	.54	near	.50	.50	of	.52	.52
on	.62	.49	over	.61	.50	to	.66	.61
under	.34	.54	with	.52	.51	without	.37	.54

Table 3.9. Probability of the preposition selecting for local attachment (for *who* and *which* clauses)— Derived from the Penn WSJ Treebank.

For *who* clauses, if the probability of the preposition's object being *human*, *group (organisation)* or *animal* is greater than that of the preposition's subject, then the preposition selects for local attachment and feature 14 is set, otherwise feature 15 is set. For *which* clauses, if the probability of the preposition's object not being *human* is greater than the probability of the preposition's subject not being *human*, then feature 14 is set, otherwise feature 15 is set. Table 3.9 gives the probability that the preposition selects for local attachment for some common prepositions. For probabilities greater than 0.5 in table 3.9, features 14 is set and for the rest, feature 15 is set.

The other features I use are for number agreement with the verb in the relative clause (features 27-30), whether the clause is restrictive (feature 2) and whether the noun phrases are definite (features 12-13, 25-26).

The accuracy of the machine learning approach for deciding attachment for *who* and *which* clauses when the preceding noun phrase has the structure `NP1 Prep NP2` is shown in table 3.10. Unfortunately, the limited number of examples available meant that I could not create an unseen test set. Instead I used five-fold cross-validation on the 248 examples for *who* relative clauses preceded by `NP1 Prep NP2`, dividing them into four sets of 50 and one set of 48. An experiment was run with each of the sets as test data (and the other four as training data). The results in table 3.10 are an average of the results of these five experiments. I used nine-fold cross-validation on the 466 examples for *which* clauses, dividing them into eight sets of 50 and one set of 66.

For *who* clauses, the machine learning approach gave results that were roughly 25% better than the local attachment baseline. *Which* clause attachments were not learnt as well as *who* clause attachments. The WordNet hierarchy was obviously useful when exactly one of `NP1` and `NP2` was *human*. In the majority of cases, however, neither was *human* and, as the second baseline suggests, the prepositions did not provide much of a clue either, so the network had very little to go on.

3.2.6 Interpreting these Results

Psycholinguistic studies suggest that when agreement values cannot rule out either `N1` or `N2`, adult native speakers of English tend to associate the relative clause with `NP2` rather than `NP1`, while for many other languages, including Spanish, German, French and Greek, adult native speakers show a preference for `NP1` (Cuetos and Mitchell, 1988; Gilboy et al.,

Pronoun	Data Set	Size	Baseline1	Baseline2	Winnow
who	Training Set	˜200	.67	.73	.92
who	Test Set	˜50	.67	.73	.91
which	Training Set	˜400	.70	.63	.77
which	Test Set	˜50	.70	.63	.77

Baseline1: Always attach locally
Baseline2: Attach according to the preposition's preferences

Table 3.10. Accuracy results for the machine learning approach to relative clause attachment

Pronoun	Winnow	Salience	B&C	Baseline[1]
who	.91	.88	.69[2]	.67
which	.77	.75	-	.70

[1] Baseline: Always attach locally
[2] Recall = .62

Table 3.11. Comparison of my salience-based approach to relative-pronoun resolution with my machine-learning approach and the Briscoe and Carroll (1995) parser.

1995; Fernandez, 2000). This preference for NP2 attachment in English is explained by the locality principle of *recency*, which prefers attachment to the most recently processed phrase. Table 3.10 shows that there is a preference in edited text for local attachment, with ˜69% of clauses attaching to NP2.

There are also studies that suggest that for genuinely ambiguous cases, adult speakers' attachment preferences are influenced by the type of preposition (showing, a preference for NP1 attachment for complex NPs joined by *of*, and a preference for NP2 attachment for noun phrases joined by *with*), though children appear to disambiguate purely on the basis of structure (Felser et al., To appear; Felser et al., 2003). Table 3.10 confirms that making attachment decisions purely based on the preposition is also an effective strategy, giving an accuracy of ˜66%. Interestingly, table 3.10 suggests that deciding attachment purely on the basis of structure gives better results for *which clauses*, while deciding attachment purely based on the preposition gives better results for *who* clauses.

Figure 3.11 compares the performance of the machine-learning approach, the salience approach and the Briscoe and Carroll (1995) parser on relative clause attachment (only cases with NP1 Prep NP2 ambiguity) using Penn WSJ Treebank data. The treebank data was converted to plain text. The results for the Briscoe and Carroll (1995) parser were computed from the parse trees it generated from the plain text. The results for the salience-based approach were computed from the output of my system (after chunking the plain text using the LT TTT and pronoun resolution).

As mentioned earlier, the Briscoe and Carroll (1995) parser attaches non-restrictive relative clauses to the root node of the parse tree as an adjunct, following the treatment of Nunberg (1990). Unfortunately, almost all the Wall Street Journal *which* clauses are non-restrictive (American English prefers *that* to *which* for the restrictive case). Hence the evaluation for the Briscoe and Carroll (1995) parser on *which clauses* was meaningless. This was also the main reason for loss of recall on *who* clauses, though there were also a few sentences that did not return parses.

There are a couple of points to consider when analysing the results in table 3.11. The first is that the machine learning approach has been specifically trained on this data set, while the parser and salience-based approach have not. The second is that the features used by the machine learner have been extracted from the *perfect* parses in the treebank. Taking these considerations into account, the salience-based approach performs creditably in the comparison. Further, as described in section 3.2.4, the salience-based approach can handle other kinds of ambiguities, like appositives, and is also easy and efficient to incorporate into my analysis module.

Both the machine learning and the salience-based approaches perform better than the baseline and the Briscoe and Carroll (1995) parser. This indicates that relative clause attachment is not a purely syntactic phenomenon, and issues like animacy, prepositional preferences and even attentional state can be useful in its resolution.

3.3 Deciding Clause Boundaries

I now consider the issue of deciding relative clause boundaries. I consider the non-restrictive and restrictive cases separately.

3.3.1 *Non-Restrictive Relative Clauses*

Determining where a relative clause ends is not always trivial. Non-restrictive relative clauses can extend to the end of the sentence or end with a comma. However, there might be commas internal to the clause so that at each comma after the clause starts, a decision needs to be made on whether the clause ends or not. I devised a set of heuristics for making this decision based on a manual examination of 290 non-restrictive *who* clauses and 846 non-restrictive *which* clauses in my training set derived from the Penn WSJ Treebank. These heuristics are encoded in algorithm 3.2.

Step 4 is required for jumping over parentheticals like *reports Stephen Labaton of The Times* in:

> Now that company is being sued by investors, [$_{RC}$who, reports Stephen Labaton of The Times, claim that management defrauded them of millions].

Most ambiguous commas were followed by either noun groups (15%) or verb groups (67%). All appositives attached locally within the clause (step 5(a)). This is because when a relative clause and an appositive attach to the same noun phrase, the appositive always precedes the relative clause. The verb groups always ended the clause unless they were past participle, present participle or gerund in which case they acted like appositives and attached locally (step 5(b)). Step 5(c) checks for commas that conjoin adverbs or adjectives. Step 5(d)i is designed to handle structures like:

> For Jan Stenbeck, who went to Harvard Business School, and who worked at Morgan Stanley in New York, chasing the America's Cup had been a longtime dream.

Algorithm 3.2 Deciding non-restrictive relative clause boundaries

```
Decide-Non-Restrictive-RC-Boundaries
  1. LET $n$ be the number of commas between “, {who|which}” and the end of the
     sentence (</S1>) or enclosing clause (</SIMP-...>).
  2. IF $n = 0$ THEN clause extends till the end of sentence
  3. IF $n > 0$ THEN a decision needs to be made at each comma as follows:
  4. IF the relative pronoun is immediately followed by a comma THEN Jump to the
     token after the next comma
  5. FOR each comma (scanning from left to right) DO
       (a) IF followed by an appositive (appositive determination is described in section
           3.4.2) THEN INTERNAL comma
       (b) IF followed by a verb group THEN
             i. IF the verb has POS “VB{N|G}” THEN INTERNAL comma
       (c) IF an implicit conjunction of adjectives or adverbs like “JJ, JJ” or “RB, RB”
           THEN INTERNAL clause
       (d) IF it is a Pronoun_X clause where Pronoun_X={who|which} THEN
             i. IF “, CC Pronoun_X” THEN INTERNAL clause and DELETE “Pronoun_X”
            ii. IF “, {who|which|that}” THEN INTERNAL comma
  6. ELSE by default end clause on first comma
```

and marks it up; deleting the second occurrence of *who* to give:

> For Jan Stenbeck, [$_{RC}$who went to Harvard Business School, and worked at Morgan Stanley in New York], chasing the America's Cup had been a longtime dream.

Step 5(d)ii follows because it is unlikely that two consecutive relative clauses attach to the same noun phrase without an intervening conjunction and it is much more likely that the second one attaches locally. It is also implausible that a different relative pronoun (like *which* or *that*) attaches to the same noun phrase as a *who* clause or vice-versa. If step 5 cannot make a decision, the default (step 6) is to end the clause at the comma.

The WSJ Treebank contains too few instances of non-restrictive *that* clauses to generalise over. Further, most of the instances that are present attach to clauses rather than noun phrases. It was therefore decided to not simplify non-restrictive *that* clauses.

3.3.2 Restrictive Relative Clauses

I use a similar algorithm for restrictive relative clauses. However, I can no longer rely on punctuation to mark the end of a clause. The procedure I use to mark restrictive relative clauses is shown in algorithm 3.3. The main difference from algorithm 3.2 is that I now need to check for an end of clause not only at punctuation, but also at each verb group and relative pronoun.

Algorithm 3.3 Deciding restrictive relative clause boundaries

```
Decide-Restrictive-RC-Boundaries
  1. IF the relative pronoun is immediately followed by a comma THEN Jump to the
     token after the next comma
  2. Jump forwards past one verb group and one noun group.
  3. IF a complementiser was encountered after the verb group, or the verb group con-
     tained a saying verb THEN jump ahead past the next verb group as well.
  4. FOR each comma, colon, semicolon, verb group or relative pronoun (processing in a
     left to right order) DO
       (a) IF colon or semicolon or end of enclosing clause (</SIMP...>), THEN END
           CLAUSE
       (b) IF a comma followed by an appositive (appositive determination is described
           in section 3.4.2) THEN INTERNAL comma
       (c) IF a comma followed by a verb group THEN
             i. IF the verb has POS "VB{N|G}" THEN INTERNAL comma
       (d) IF a comma that is an implicit conjunction of adjectives or adverbs like "JJ,
           JJ" or "RB, RB" THEN INTERNAL clause
       (e) IF we are inside a Pronoun_X relative clause where Pro-
           noun_X={who|which|that} THEN
             i. IF " CC Pronoun_X" THEN INTERNAL clause and DELETE "Pronoun_X"
            ii. IF ", {who|which|that}" THEN INTERNAL comma
           iii. IF "{who|which|that}" THEN INTERNAL comma
            iv. Recursively find the end of the embedded clause
  5. IF previous token is a conjunction or a subject pronoun (I, they, he, we, she) THEN
     INTERNAL comma
  6. ELSE by default end clause
```

Step 1 deals with clause-initial parentheticals. Step 2, which skips over a verb group and a noun group, marks the minimum relative clause. If a *saying* verb or a complementiser is encountered, I need to extend the relative clause by another verb and noun group (step 3). I then end the clause if I encounter a colon or semicolon (step 4(a)). If I encounter another relative pronoun, I need to recursively find the end of that clause (step 4(e)iv). If step 4 does not decide the issue, I look at the previous token in step 5. If the previous

Data Set /	**Non-Restrictive**			**Restrictive**	
Clause Type	**Size**	**Accuracy**[1]	**Accuracy**[2]	**Size**	**Accuracy**[1]
Training	1036	.99	.97	494	.91
Test (*who*)	236	.98	.97	292	.86
Test (*which*)	696	.97	.94	27	.89
Test (*that*)	-	-	-	320	.89

[1] Accuracy for all clauses
[2] Accuracy for only ambiguous clauses

Table 3.12. Evaluation of clause boundary algorithm on the Penn WSJ Treebank

token is a conjunction or a subject pronoun, it suggests that the clause hasn't ended yet. Otherwise, the default (step 6) is to end the clause.

3.3.3 *Evaluation*

I performed two evaluations of my algorithm. The first evaluation was on the Penn WSJ Treebank corpus. The results are shown in table 3.12. Non-restrictive clauses are labelled ambiguous if there is at least one comma between the relative pronoun and the end of the sentence. The second evaluation was on the test data for the Computational Natural Language Learning Workshop (CoNLL-2001) on clause identification (Daelemans and Zajac, 2001) at ACL-2001. I compared my algorithm against the best performing clause identification system at CoNLL-2001 (Carreras and Màrquez, 2001) and the Briscoe and Carroll (1995) parser. The results are shown in figure 3.13. This comparison, against a system tackling the harder task of identifying all clauses in text and a statistical parser illustrates the point that disambiguation algorithms aimed at a specific tasks can perform better on that task than more general purpose approaches. The workshop provided training and test sets and the output of six systems on the test set are downloadable at the website. The test set contained ˜100 non-restrictive relative clauses that did not end unambiguously in a full stop and ˜200 restrictive relative clauses.

3.4 Marking up Appositives

3.4.1 *What is Apposition*

Quirk et al. (1985) identify three conditions that define apposition:

1. Each of the appositives can be separately omitted without affecting the acceptability of the sentence.
2. Each fulfils the same syntactic role in the resultant sentences.
3. There is no difference between the original sentence and either of the resultant sentences in extra-linguistic reference.

For example, if the appositives are omitted from:

Mr. Vinken is chairman of *Elsevier N.V., the Dutch publishing group.*,

we obtain two sentences:

Data Set / Algorithms	B&C	C&M	Me
Non-Restrictive Relative Clauses	.77[1]	.81	.96
Restrictive Relative Clauses	.67[2]	.76	.89

B&C: Briscoe and Carroll (1995) parser [1] recall of .85 [2] recall of .95
C&M: Carreras and Màrquez (2001) clause identifier

Table 3.13. Evaluation of clause boundary algorithm on CoNLL'01 Task

(a) Mr. Vinken is chairman of Elsevier N.V.
(b) Mr. Vinken is chairman of the Dutch publishing group.

Both sentences are acceptable, the syntactic role of the appositive in both is the same, and as *Elsevier N.V* and *the Dutch publishing group* are co-referential in the original sentence, we can assume their reference to be the same in both sentences.

Quirk et al. (1985) call apposition that satisfies all three conditions *full apposition. Partial apposition*, which can violate any or all of the three conditions, is hard to define. An example from Quirk et al. (1985) is:

> *Norman Jones, at that time a student,* wrote several best-sellers.

Omitting appositives, we obtain the two sentences:

(a) Norman Jones wrote several best-sellers.
(b) At that time a student wrote several best-sellers.

Condition 2 is not satisfied by this example, as *at that time a student* is not a constituent in sentence (b).

Quirk et al. (1985) also classify apposition in other ways, like *strict/weak* (in weak apposition, the appositives have different syntactic categories) and *non-restrictive/restrictive* (restrictive apposition does not contain punctuation, for example, ***The utter fool John*** *insisted on going there*).

3.4.2 Identifying Appositive Boundaries

I only mark-up strict non-restrictive appositives for simplification. This includes some cases of partial apposition, for example:

> There were more than 100 workers trapped in the coal mine in *Huaibei, 420 miles south of Beijing.*

but not others that violate the strictness criterion, for example:

> *Norman Jones, at that time a student,* wrote several best-sellers.

I identify as an elementary appositive (`appos_e`), constructs that match the following pattern:

, NP ['Prep NP']* [RC$_{rest}$]? [,|EOS]

Data Set	Size	Identification
Training	270	.90
Test	513	.88

Table 3.14. Accuracy results for appositive identification using the Penn WSJ Treebank

This pattern matches a comma followed by a noun phrase that is followed by zero or more prepositional phrases and zero or one restrictive relative clause. The pattern ends in either a comma or an end-of-sentence marker. Examples of appositives identified by my pattern `appos_e` are:

> Lorillard Inc., [$_{appos}$ *[the unit] of [New York-based Loews Corp.] [$_{RC_{rest}}$ that makes Kent cigarettes*]], stopped using crocidolite in its Micronite cigarette filters in 1956.

> “ There’s no question that some of those workers and managers contracted asbestos-related diseases, ” said Darrell Phillips, [$_{appos}$ *[vice president] of [human resources] for [Hollingsworth & Vose]*].

To avoid fragmenting the text too much, I do not recursively simplify appositives and only mark-up one large appositive; for example, in:

> Larry Birns , [$_{appos}$ *director of the Washington-based Council on Hemispheric Affairs , a liberal research group*], said that Latin American countries would be “ profoundly disappointed ” if Canada were to follow the U.S. lead in the OAS .

I treat *director of the Washington-based Council on Hemispheric Affairs , a liberal research group* as one appositive. Thus I mark as an appositive (`appos`), the longest sequence of one or more simple appositives; i.e. the longest string that matches the pattern:

```
appos = [appos_e]+
```

In addition, I only mark-up for simplification appositives that are longer than two words. This is again to prevent too much fragmentation of the text, as well as to avoid problems due to place names in constructs like *the workers at the West Groton, Mass., paper factory.* I also perform a check for coordinated noun phrases; I scan ahead from the end of the appositive I have identified, till I reach a verb or end-of-sentence marker. If I encounter an *and* or *or*, I reject my analysis of the appositive. This stops me making wrong analyses like:

> Their talks would include human rights, [$_{appos}$ regional disputes, relations with allies], economic cooperation and joint efforts to fight narcotics.

I present my results for appositive identification on Penn WSJ Treebank data in table 3.14. Apposition is not marked-up explicitly in the treebank. The evaluation was therefore done on the basis of bracketing. An appositive (X), as determined by my program using the pattern `appos`, was marked as being correct if all the following conditions were satisfied:

1. X was a noun phrase and surrounded by punctuation in the treebank.
2. The immediate enclosing bracketing marked a noun phrase in the treebank.
3. X was the right-most phrase in the enclosing bracketing.

For example, in the following extract from the treebank:

> ...) (PP (PREP than) (NP (NP (NP (DET the) (JJ common) (NN kind)) (PP (PREP of) (NP (NN asbestos))) (, ,) (NP (NN chrysotile)) (, ,)) (VP (VBN found) (NP...

(NP (NN chrysotile)) is a noun phrase surrounded by punctuation and is immediately enclosed by the noun phrase:

> (NP (NP (DET the) (JJ common) (NN kind)) (PP (PREP of) (NP (NN asbestos))) (, ,) (NP (NN chrysotile)) (, ,))

It is also the right-most phrase in the above noun phrase and therefore an appositive

This immediately enclosing bracketing is also used to evaluate appositive attachment in section 3.4.3; the identified appositive attaches to the left-most entity in the enclosing brackets, in this case, to *(NP (DET the) (JJ common) (NN kind))*.

Most of the errors in identifying appositives could be traced back to incorrect part-of-speech tagging; for example, in:

> 'Smokers have rights too,' says Al Ries, [$_{appos}$ chairman of Trout & Ries Inc., a Greenwich, Conn.], marketing$_{vbg}$ [strategy firm].

marketing is tagged as a verb which leads to incorrect noun chunking and hence incorrect appositive identification.

3.4.3 Deciding Appositive Attachment

I decide appositive phrase attachment in the same manner as relative clause attachment. I resolve the head noun phrase in the appositive to the most salient noun phrase that agrees with it in number, animacy and gender, subject to the syntactic constraint below.

Syntactic Filter

I use a very restrictive filter that ensures that the noun phrase that an appositive attaches to can only be separated from it by prepositional phrases; for example, in the sentence:

> Preliminary tallies by the Industry Ministry showed [another trade deficit]^{1} in October, *[the fifth monthly setback]*$^{2\#1}$ *in a year*, casting a cloud on South Korea 's export-oriented economy.

Data Set/Algorithm	Size	Baseline*	Salience
Training	62	.79	.84
Test	205	.80	.87

* Baseline: Always attach locally.

Table 3.15. Accuracy results for ambiguous appositive attachment

Only 11% of appositives in the WSJ Treebank had attachment ambiguities. Further, the local attachment baseline for the ambiguous cases was as high as 80%. This meant that the local attachment baseline gave an overall accuracy of 97.8%. Table 3.15 compares my salience based approach with the local attachment baseline for 267 ambiguous instances in the WSJ Treebank. I used the same salience function and agreement and syntax filters as for relative clause attachment and an examination of the training set suggested that these did not need to be changed.

3.5 Marking-up Conjoined Clauses

My transformation stage simplifies coordinated clauses as well as subordinated and correlated clauses. My analysis stage handles both *prefix* and *infix* conjunctions. In this section, the patterns `Clause`n match the longest strings that have a subject and a verb and don't have crossing brackets with any previously marked-up clauses.

3.5.1 *Prefix Conjunctions*

Subordinated clauses with a prefix conjunction match the following pattern:

`CC Clause`$_1$`, Clause`$_2$`.`

A marked-up example is:

> [$_{CC}$ Although] [$_{Clause_1}$ both India and Pakistan announced partial troop withdrawals along the border], [$_{Clause_2}$they both left their forces in Kashmir intact].

The issues involved in marking-up subordinated clauses with a prefix conjunction are similar to those of determining non-restrictive clause boundaries. At each comma, I need to decide whether or not to end the first clause and start the second. I reuse the same algorithm (algorithm 3.2) with an additional check that both clauses contain a verb and subject. The subordinating conjunctions in prefix position that I mark-up for simplification are *though*, *although*, *when*, *if*, *since*, *as* and *because*.

I also mark up the correlative *if...then* construct that matches the patterns:

`If Clause`$_1$`, then Clause`$_2$`.`

`If Clause`$_1$` then Clause`$_2$`.`

A marked-up example is:

> [$_{CC_1}$ If] [$_{Clause_1}$ people have got in place proper effective safety measures], [$_{CC_2}$ then] [$_{Clause_2}$ naturally we are pleased about that].

Data Set	Size	Accuracy
Training	100	.96
Test	200	.94

Table 3.16. Accuracy results for conjoined clause identification

3.5.2 Infix Conjunctions

Coordinated and subordinated clauses with infix conjunctions match the patterns:

```
Clause_1 CC Clause_2.
Clause_1, CC Clause_2.
```

Marked-up examples are:

> $[_{Clause_1}$ I have been involved with badgers for 24 years] $[_{CC}$ and] $[_{Clause_2}$ I have never heard of anything like this].

> $[_{Clause_1}$ Labor has complained that the budget favors settlers over the poor], $[_{CC}$ but] $[_{Clause_2}$Mr. Sharon has said he would dismiss anyone from his government who opposed his plan].

The coordinating conjunctions that I handle are *and*, *or* and *but*. The subordinating conjunctions in infix position that I mark-up for simplification are *though*, *although*, *because*, *since*, *as*, *before*, *after* and *when*. This list of conjunctions was determined by manually examining sentences containing conjunctions in the WSJ Treebank and selecting the conjunctions where the two clauses could be separated without compromising meaning. Conjunctions that occurred less than 10 times in the treebank were excluded, as I was unable to satisfy myself that they could be simplified reliably.

I only mark-up infix conjunctions if both $Clause_1$ and $Clause_2$ contain a verb and subject. In particular, I do not simplify coordinated verb phrases. The reason for this is that coordinated VPs often occur within other constructs and simplifying them usually results in fragmenting the text too much.

I present my results for conjoined-clause identification in table 3.16. I used the WSJ Treebank for the experiment. This is an evaluation of whether what I have marked-up are actually clauses or not, using the WSJ Treebank as the gold standard. Some examples of errors are:

> Last March, after attending a teaching seminar in Washington, Mrs. Yeargin says she returned to Greenville two days $[_{before-CL}$ before annual testing feeling that she hadn't prepared her low-ability geography students adequately].

> The average maturity for funds open only to institutions, considered by some to be a stronger indicator $[_{because-CL}$ because those managers watch the market closely, reached a high point for the year – 33 days].

3.6 A Holistic Evaluation

I now perform an evaluation of my entire analysis module. The reasons for this evaluation are two-fold. Firstly, it is required so that I can see how the errors in attachment and boundary determination combine with each other. I expect this evaluation to give me an indication of what proportion of sentences I can expect to simplify correctly. Secondly, many of the evaluations in this chapter have had to be carried out on the WSJ Treebank. This evaluation, on a corpus of Guardian Newspaper text, is likely to indicate how well my algorithms perform on a different genre of edited text.

I ran twelve news reports (containing 263 sentences) from the Guardian newspaper through my analysis module and performed an evaluation on all the constructs that were marked-up for simplification. As this evaluation involved subjectivity (there is no gold standard like the Penn WSJ Treebank to compare against), I used two native-English speakers for the annotation task. The guidelines that I gave my annotators are attached in appendix A.1.

There were 203 decisions (both attachment and identification) that needed to be made. There were 105 sentences that had at least one construct marked-up to be simplified.

The two independent annotators disagreed on only 9 decisions out of 203. This gives an inter-annotator agreement of 96% ($\kappa = 0.78$)[15]. There were 18 decisions spread across 17 sentences which both independent annotators marked incorrect, and 9 decisions spread across an additional 7 sentences which only one annotator marked incorrect. In other words, 87% of decisions were made correctly according to both annotators and 91% of decisions were made correctly according to at least one annotator. 77% of the 105 the simplifiable sentences contained no analysis errors according to both annotators and 84% were error-free according to at least one independent annotator.

When I used myself as the third annotator, 75% of simplifiable sentences were error-free according to all three annotators, 79% of simplifiable sentences were error-free according to two out of three annotators and 87% of simplifiable sentences were error-free according to at least one annotator. The inter-annotator agreement for the three annotators was now 93% ($\kappa = 0.76$).

3.7 Discussion

In this chapter I have explored the use of shallow salience-based discourse models coupled with knowledge sources like WordNet for a variety of tasks. I have shown that

[15] κ (kappa) is a measure of inter-annotator agreement over and above what might be expected by pure chance (See Siegel and Castellan (1988) for a description of the formula and Carletta (1996) for its use in NLP). The formula for κ is:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

The numerator measures the difference between the proportion $P(A)$ of times the annotators agree and the proportion $P(E)$ of times they would be expected to agree by chance. The denominator measures the difference between perfect agreement 1 and chance agreement $P(E)$. $\kappa = 1$ if there is perfect agreement between annotators, $\kappa = 0$ if the annotators agree only as much as you would expect by chance, $\kappa < 0$ if the annotators agree less than predicted by chance.

shallow inference procedures used with a shallow discourse model can give good results on third-person pronoun resolution, even without using a parser. These results are not surprising, as syntactic constraints on pronoun resolution are largely indicatory and semantic constraints are much more important.

A more interesting aspect of this chapter is my demonstration that the combination of shallow discourse models and shallow semantic inference is effective not just for resolving third-person pronouns but also for making attachment decisions for relative clauses and appositives. These are tasks that have traditionally been performed in a syntactic framework. It is significant that shallow processing at the discourse and semantic levels can outperform syntactic approaches on relative clause and appositive attachment. Indeed, this combination of discourse and semantics appear to perform as well on relative clause attachment as a machine learning approach using 31 features. It is not entirely clear why salience is useful for resolving appositive and relative clause attachment. These attachment decisions involve balancing two competing locality principles— *recency*, which favours local attachment, and *predicate proximity*, which favours wide attachment. It is possible that the salience function (which combines numerical scores for recency and grammatical relation) succeeds in balancing the two competing locality principles. It is, however, evident that most of the work in disambiguation is done by the agreement features; in particular, the animacy feature.

My results also suggest that shallow solutions tailored to specific syntactic problems can achieve performance on those problems that equal, or even exceed, that of more sophisticated general purpose models. For example, simple pattern matching techniques based on local context are sufficient to decide between three or four grammatical relations, though it should be emphasised that methods that shallow cannot scale up to deciding between the other GRs in figure 3.1. Simple algorithms based on the local context described by part of speech tags, like the ones presented for deciding relative clause boundaries, can be better at disambiguation than both sophisticated wide coverage parsers and general purpose clause boundary determination algorithms. This is quite understandable; as statistical models for parsing are trained using an evaluation criteria that involves many syntactic constructs, it is quite plausible that they are not optimised for my specific tasks.

Finally, the results presented in this chapter show that I cannot hope to perform syntactic simplification *perfectly*. There will always be some sentences that get simplified incorrectly due to errors in the analysis stage, both in attaching and in determining the boundaries of clauses and phrases. The question then arises— how accurate does my analysis need to be for the system to be useful? The experiments with aphasics and the deaf described in the introduction provide some answers. My system can decide relative clause boundaries and attachment with an accuracy of over 85%, compared to the 25-60% reported for readers in tables 1.1 and 1.2. This suggests that my simplified sentences might be easier to understand for many. Whether or not an entire text will be easier to comprehend when simplified will depend on how easy it is to link together the meanings of individual simplified sentences. This will depend on how well I can preserve the cohesion of the original text; a topic I address in chapter 5 on *regeneration*.

4 *Transformation*

The transformation module is where the actual syntactic simplification occurs in my architecture. As the analysis module has already marked-up the constructs to be simplified and the regeneration module handles the discourse aspects of simplification (as described in chapter 2 on *architecture*), the functions of the transformation module are quite straightforward.

The primary function of the transformation module is to apply syntactic-simplification rules to the analysed text. The second function is to invoke the regeneration module when required. This chapter presents my set of simplification rules and details the interaction between the transformation and regeneration modules. Most of this interaction just involves the transformation module providing the regeneration module with the information it requires for preserving text cohesion. As I show in section 4.2, this information can be encoded succinctly in the form of rhetorical relations.

There are, however, two issues whose resolution requires more involved cooperation between the transformation and regeneration modules— deciding the order in which to use the simplification rules and ordering the simplified sentences. It is possible that there is more than one construct that can be simplified in a sentence. My transforms result in splitting sentences into two; hence, for example, applying three transforms to a sentence will result in four sentences. This raises the issue of how to order the simplified sentences. In my modular architecture, the transformation stage deals with only sentence level syntactic transformations and all discourse level decisions are taken in the regeneration stage. However, while sentence ordering is really a regeneration issue (with consequences for text cohesion), individual transforms can place constraints on both the ordering of the transformed sentences and the ordering of sentences generated by further transformation of the original sentence. This makes the order of application of simplification rules dependent on the sentence-order decisions resulting from previous simplifications. The central issue in this chapter is of how to resolve the intertwined issues of transform and sentence ordering to, for example, allow:

> Mr. Anthony, who runs an employment agency, decries program trading, but he isn't sure it should be strictly regulated.

to be simplified to

> Mr. Anthony runs an employment agency. Mr. Anthony decries program trading. But he isn't sure it should be strictly regulated.

but not to:

> Mr. Anthony decries program trading. Mr. Anthony runs an employment agency. But he isn't sure it should be strictly regulated.

In my architecture, individual simplification rules in the transformation module provide the regeneration module with information about the rhetorical relations that hold between the simplified sentences. The regeneration module converts this information into explicit constraints on sentence order (and also introduces other constraints on sentence order arising from centering theory) and resolves them, passing back constraints on future sentence-ordering to the transformation module. I describe this process in detail in sections 4.2–4.4. The individual transforms are carried out using hand-crafted simplification rules. I describe these rules in section 4.1.

4.1 Simplification Rules

My transformation module uses seven hand-crafted syntactic simplification rules (rules 4.1–4.7 in the discussion below). There are three rules for conjunction and two rules each for relative clauses and apposition.

4.1.1 *Conjoined Clauses*

The transformation rule for prefix subordination is:

$$(4.1)\quad \texttt{CC}_n\ [_{Clause_n 1}\ \texttt{X}]\texttt{,}\ [_{Clause_n 2}\ \texttt{Y}]\texttt{.} \longrightarrow \begin{array}{ll} (a) & \texttt{X.} \\ (b) & \texttt{Y.} \end{array}$$

where the conjunction $\texttt{CC}_n$ matches one of *though*, *although*, *when* and *because*. As an example, this rule splits:

> $[_{CCm}$ Although$]$ $[_{Clause_m 1}$ both India and Pakistan announced troop withdrawals along the border$]$, $[_{Clause_m 2}$ they both left their forces in Kashmir intact$]$.

into:

> (a) Both India and Pakistan announced troop withdrawals along the border.
> (b) {But}[16] they both left their forces in Kashmir intact.

The rule for the correlative *if...then* and the subordinative *if* construct is:

$$(4.2)\quad [_m\ \texttt{If}][_{Clause_m 1}\ \texttt{X}]\ [\texttt{then}|\texttt{,}]\ [_{Clause_m 2}\ \texttt{Y}]\texttt{.} \longrightarrow \begin{array}{ll} (a) & \texttt{X.} \\ (b) & \texttt{Y.} \end{array}$$

As an example, this rule splits:

[16] The cue-word *but* is not introduced by rule 4.1; rather, it is introduced by the regeneration module on the basis of the rhetorical relation between the conjoined clauses (refer to section 4.2). In the examples in this chapter, curly brackets denote the fact that they contain words that are introduced by the regeneration stage. These are only shown to make the examples look realistic.

> [$_{CC_n}$ If] [$_{Clause_n1}$ people have got in place proper effective safety measures], [$_{CC_n}$ then] [$_{Clause_n}$ naturally we are pleased about that].

into:

> (a) {Suppose} people have got in place proper effective safety measures.
> (b) {Then} naturally we are pleased about that.

The rule for infix coordination and subordination is:

$$(4.3)\qquad \texttt{[}_{Clause_n1}\ \texttt{X]}\ \texttt{[,]?}\ \ \texttt{[}_{n}\ \texttt{CC]}\ \texttt{[}_{Clause_n2}\ \texttt{Y]]} \longrightarrow \begin{array}{ll}(a) & \texttt{X.}\\ (b) & \texttt{Y.}\end{array}$$

where the conjunction `CC` matches one of *though*, *although*, *but*, *and*, *because*, *since*, *as*, *before*, *after* and *when*. For example, this rule splits:

> [$_{Clause_n1}$ I have been involved with badgers for 24 years] [$_n$ and] [$_{Clause_n2}$ I have never heard of anything like this].

into:

> (a) I have been involved with badgers for 24 years.
> (b) {And} I have never heard of anything like this.

and:

> [$_{Clause_n1}$ Labor has complained that the budget favors settlers over the poor], [$_n$ but] [$_{Clause_n2}$ Mr. Sharon has said he would dismiss anyone from his government who opposed his plan].

into:

> (a) Labor has complained that the budget favors settlers over the poor.
> (b) {But} Mr. Sharon has said he would dismiss anyone from his government who opposed his plan.

4.1.2 Relative Clauses

The transformation rules for relative clauses are:

$$(4.4)\qquad \texttt{V}\ \texttt{W}^{x}_{NP}\ \texttt{X}\ \texttt{[}_{RC\_n}\ \texttt{RELPR}^{\#x}\ \texttt{Y]}\ \texttt{Z.}\!\longrightarrow \begin{array}{ll}(a) & \texttt{V W X Z.}\\ (b) & \texttt{W Y.}\end{array}$$

$$(4.5)\qquad \texttt{V}\ \texttt{W}^{x}_{NP}\ \texttt{X}\ \texttt{[}_{RC\_n}\ \texttt{RELPR}^{\#x}\ \texttt{Y].}\!\longrightarrow \begin{array}{ll}(a) & \texttt{V W X.}\\ (b) & \texttt{W Y.}\end{array}$$

These rules state that if, in my analysed text, a relative clause `RELPR Y` attaches to a noun phrase `W`, then I can extract `W Y` into a new sentence. In the case of non-restrictive clauses, the enclosing commas implicit in `X` and `Z` are removed. These rules can simplify:

> [Garret Boone]^1, [$_{RC_n}$ who$^{\#1}$ teaches art at Earlham College], calls the new structure “just an ugly bridge” and one that blocks the view of a new park below.

to:

(a) Garret Boone calls the new structure "just an ugly bridge" and one that blocks the view of a new park below.
(b) Garret Boone teaches art at Earlham College.

and:

'The pace of life was slower in those days,' says [51-year-old Cathy Tinsall]^1 from South London, [$_{RC_n}$ who$^{\#1}$ had five children, three of them boys].

to:

(a) 'The pace of life was slower in those days,' says 51-year-old Cathy Tinsall from South London.
(b) Cathy Tinsall had five children, three of them boys.

Before applying the simplification rules for relative clauses, I perform a check on the noun phrases they attach to. To avoid performing the simplification if the clause attaches to a partitive, I make sure that if the clause attaches to the pattern `NP1 of NP2`, then `NP1` is not a numerical attribute (like *number*, *percentage*, *dozens* etc.). This is to avoid simplifying constructs like *the number of people who...* I also do not perform the simplification if the clause attaches to the noun phrase *those*. This is to avoid simplifying constructs like *those of us who...* or *those who...*

4.1.3 *Appositive Phrases*

The transformation rules for appositives are:

$$(4.6)\quad \texttt{U}\ \texttt{V}^{x}_{NP}\ \texttt{W,}\ [_{appos\_n}\ \texttt{X}^{\#x}\ \texttt{Y}],\ \texttt{Z.} \longrightarrow \begin{array}{ll} (a) & \texttt{U V W Z.} \\ (b) & \texttt{V Aux X Y.} \end{array}$$

$$(4.7)\quad \texttt{U}\ \texttt{V}^{x}_{NP}\ \texttt{W,}\ [_{appos\_n}\ \texttt{X}^{\#x}\ \texttt{Y}]. \longrightarrow \begin{array}{ll} (a) & \texttt{U V W.} \\ (b) & \texttt{V Aux X Y.} \end{array}$$

These rules state that if, in my analysed text, an appositive `X Y` attaches to a noun phrase `V`, then I can extract `V Aux X Y` into a new sentence. The auxiliary verb `Aux` is one of *is*, *was*, *are* and *were* and is determined from the tense of the main clause and by whether `V` is singular or plural. These transforms can be used to simplify, for example:

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

to:

(a) Pierre Vinken will join the board as a nonexecutive director Nov. 29.
(b) Pierre Vinken is 61 years old.

and:

"There's no question that some of those workers and managers contracted asbestos-related diseases," said Darrell Phillips, vice president of human resources for Hollingsworth & Vose.

to:

(a) "There's no question that some of those workers and managers contracted asbestos-related diseases," said Darrell Phillips.
(b) Darrell Phillips was vice president of human resources for Hollingsworth & Vose.

Conjunctions	**Rhetorical Relation**
although, though, whereas, but, however	(a, Concession, b)
or, or else	(a, Anti-Conditional, b)
if, if...then...	(a, Condition, b)
because	(a, Justify, b)
`X`	(a, `X`, b)

Table 4.1. Rhetorical relations triggered by conjunctions

4.2 The Interface between Transformation and Regeneration

There are five issues that the regeneration module needs to resolve— cue-word selection, sentence order, referring expression generation, determiner choice and pronominal use. The first four are transform-specific, and need to be addressed immediately when a transform is performed. Pronominal use can be resolved as a post-process, in a transform-independent manner (details in chapter 5). For the regeneration module to resolve the first four issues, it is sufficient that it receives the following input from the transformation stage:

1. *(a, RR, b)*: The rhetorical relation *RR* that holds between the two simplified sentences *a* and *b* generated by the transform
2. *n*: The index of the noun phrase for which to generate the referring expression

The detailed discussion of why this specification is sufficient is postponed to chapter 5, when I address these regeneration issues. But in brief, cue-word selection, sentence order and determiner choice can be decided from the rhetorical relation, and the referring expression generator requires only the index of the noun phrase. I now make explicit the rhetorical relations that I use.

4.2.1 *The List of Rhetorical Relations Used*

Conjunctions act as cue words that can define the rhetorical relation (introduced in section 1.6.3) between the conjoined clauses. Table 4.1 shows the rhetorical relation associated with each subordinating conjunction that I simplify. In each entry, *a* is the nucleus and *b* is the satellite of the relation (*a* and *b* are the simplified sentences generated by rules 4.1–4.3). The final row in table 4.1 is a default that arises because rhetorical structure theory is in some cases not suited for my purposes (A discussion follows in section 4.2.2). For example, RST provides the rhetorical relation *circumstance* where the satellite clause provides an interpretive context of situation or time. However, I need to be able to distinguish between *when*, *before* and *after* clauses, all of which have the *circumstance* relation with their nucleus. I therefore use my own relations *(a, when, b)*, *(a, before, b)* and *(a, after, b)* for the conjunctions *when*, *before* and *after*. There are also cases of ambiguous conjunctions that can signal more than one rhetorical relation. For example, the conjunctions *as* and *since* can indicate either a *justify* or a *circumstance* relation. As my analysis module does not disambiguate rhetorical relations, I define my own relations *(a, as, b)* and *(a, since, b)* that capture the underspecified rhetorical relation.

I also need to adapt RST to offer a treatment of relative clauses and appositives. RST provides an *elaboration* relation, but the original theory does not use it for non-restrictive relative clauses. The problem is that a relative clause has a relationship with the noun phrase it attaches to, and in RST, that noun phrase does not qualify as a text span. This problem is generally overcome by labelling non-restrictive relative clauses and appositives as parenthetical units (Marcu, 1997; Marcu, 2000). But restrictive relative clauses do not qualify as parentheticals and are left without a treatment, which seems unreasonable. I continue to use the *parenthetical* relation to relate non-restrictive relative clauses and appositives to noun phrases. In addition, I use an *identification* relation to relate restrictive relative clauses to noun phrases. To motivate this relation, consider the restrictive relative clause in:

> The man $[_{RC}$ who had brought it in for an estimate] then returned to collect it.

The relative clause serves to *identify* one man from the larger set of men. The relation is not strictly parenthetical, because if the clause is omitted:

> The man then returned to collect it.,

it is likely that the reader can no longer unambiguously identify the referent of *the man*.

4.2.2 A Note on My Use of RST

As discussed in the last section, I have adapted the broad framework of RST to suit my requirements. I now provide a short discussion of how my adaptation of RST differs from its original formulation, and how my goals in this thesis relate to the goals of RST.

An important difference is that my adaptation only considers lexically signalled relations. As described in section 1.6.3 of the introduction, RST postulates that rhetorical relations need not be signalled lexically. Indeed, less than half the rhetorical relations in naturally occurring text are lexically signalled. Therefore my adaptation only deals with a subset of RST.

Also, RST was proposed as a model of conjunctive cohesion, and did not allow for referential relations. The *identification* relation that I introduced to relate a restrictive relative clause to a noun phrase is an example of a referential relation. This addition of a referential relation was required in order to offer a unified treatment of restrictive and non-restrictive relative clauses. This was important to me for my simplification task, but was obviously not an issue for the original RST, given its differing goals.

These differences in flavour between my relations and traditional RST means that my goals and techniques for analysing text differ from traditional approaches to *RST-parsing*. RST-parsing (Marcu, 1997; Marcu, 2000) is a much harder problem than that tackled in this section. It involves the identification of rhetorical relations that may or may not be signalled linguistically, the disambiguation of ambiguous linguistic cues (for example, the words *since* and *as* can signal either a *justify* or a *circumstance* relation) and the creation of a RST tree that spans the entire text. Marcu (1997) provided the following algorithm for rhetorical parsing:

1. Determine the set D of all discourse markers (linguistic cues) and the set U_T of elementary textual units in the text T.
2. Hypothesise a set of relations R between the elements of U_T.
3. Use a constraint satisfaction procedure to determine all the discourse trees of T.
4. Assign a weight to each of the discourse trees and determine the tree(s) with the maximal weight.

Comparing steps 1 and 2 of this algorithm with my approach, I use a very small set of discourse markers D, that consists of ten conjunctions and three relative pronouns (in contrast, Marcu (1997) considers over 450 discourse markers). My task of hypothesising relations between textual units (that are determined by the clause boundary routines described in chapter 3) is made easy because I define a one-to-one relationship between linguistic cues and relations. This is possible because I allow for underspecified relations (like *since* and *as*) that connect textual units linked by ambiguous linguistic cues.

My application does not require the construction of a RST tree. Therefore, I do not need to carry out steps 3 and 4. Indeed, my application (which simplifies sentences one at a time) only requires me to identify rhetorical relations between textual units within the same sentence.

In summary, while I retain the spirit of RST in using it to formalise cohesive relations in text, I deviate slightly by including a referential relation. I also make my RST-based analysis easier by restricting the number of linguistic cues and by postulating underspecified relations that do away with the need for disambiguation when hypothesising relations.

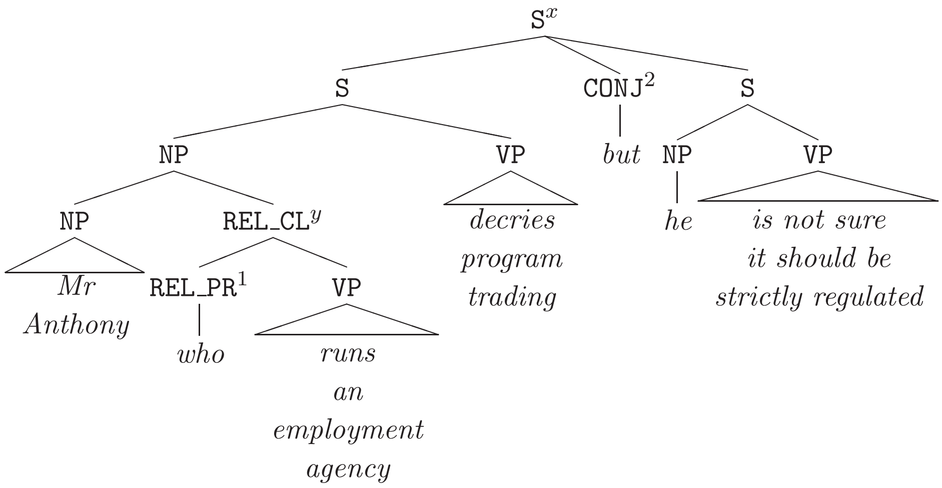
4.3 Deciding Transformation Order

Having discussed the use of RST to interface between the transformation and regeneration stages, I now return to the function of the transformation module—to decide the order in which to apply transforms. I use three examples to motivate my approach to ordering transforms. I also use these examples to illustrate that individual transforms can constrain sentence-order not just for the sentences generated by the transform, but also during further recursive simplification of these sentences. My algorithm for recursively applying transforms (presented in section 4.4) passes constraints on sentence-order down the recursion using constraint sets (also detailed in section 4.4). I use this section to flag the kinds of constraints that are required.

The easiest way to deal with multiple transforms is to apply them in the order in which the constructs occur in the sentence; that is, from left to right. If I use cue-words to mark occurrences, this corresponds to a depth first traversal of the corresponding parse tree[17].

This is illustrated in the example in figure 4.1, where the ordering (1,2) corresponds to both depth first search on the parse tree and left-to-right search on cue words in the

[17] The parse trees shown in figures 4.1–4.3 are for illustration purposes and do not correspond to the output of my analysis module. My analysed text does not consist of parse trees; however, it does contain a partial tree structure as some marked-up constructs can be embedded within others.



Original Sentence:

> Mr. Anthony, who[1] runs an employment agency, decries program trading, but[2] he isn't sure it should be strictly regulated.

Simplified Sentences:

(a) Mr. Anthony decries program trading.

(b) Mr. Anthony runs an employment agency.

(c) But he isn't sure it should be strictly regulated.

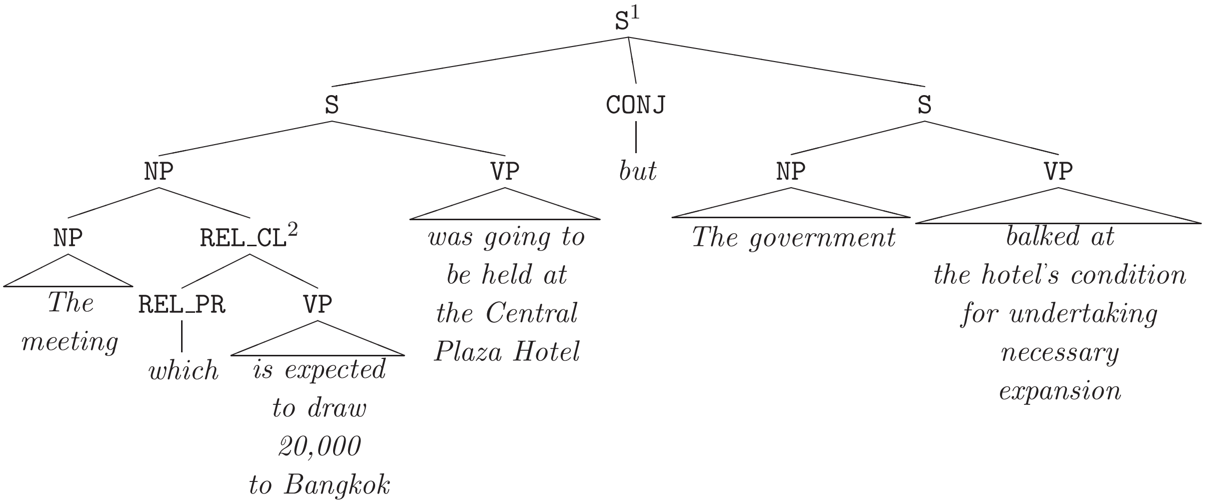
Figure 4.1. Left-to-right simplification and depth-first tree traversal

sentence. This ordering (1,2) is not satisfactory because transform 2 places constraints on the ordering of sentences generated by transform 1 and should therefore be performed first. The conjunction *but* in the original sentence relates two clauses *Mr. Anthony decries program trading* and *he isn't sure it should be strictly regulated.* To maintain the original *concession* relation, the simplified sentence (a) has to be immediately before (c). This constrains the position of sentence (b) generated by transform 1. Hence transform 2 needs to be performed first.

A top-down left-to-right search on *rules* as opposed to *cue words* allows me to place all the constraints I require and results in the optimal clause ordering. In figure 4.1, it results in the desired ordering (x,y) of transforms. Top-down left-to-right search on parse trees also corresponds to processing transforms in a left-right order on my analysed text; only I order transforms by the first clause involved, rather than the cue-word. This is illustrated in figure 4.2, where the order of applying transforms is (1, 2).

4.3.1 Sentence Ordering by Constraint Satisfaction

The examples in figures 4.1 and 4.2 illustrate that individual simplification rules can introduce constraints on the final ordering of simplified sentences. When the simplification rules are applied in a top-down manner, it is possible to resolve sentence-ordering constraints locally, rather than globally. Consider the example in figure 4.1. Global sentence


S1
S
CONJ
S
NP
VP
but
NP
VP
NP
REL_CL2
was going to be held at the Central Plaza Hotel
The government
balked at the hotel's condition for undertaking necessary expansion
The meeting
REL_PR
VP
which
is expected to draw 20,000 to Bangkok

Original Sentence:

> [$_1$ The meeting, [$_2$ which is expected to draw 20,000 to Bangkok], was going to be held at the Central Plaza Hotel], but$_1$ [$_1$ the government balked at the hotel's conditions for undertaking necessary expansion].

Figure 4.2. Top-down left-to-right search on rules

ordering would involved deciding the relative order of the three sentences:

1. Mr. Anthony decries program trading.
2. Mr. Anthony runs an employment agency.
3. But he isn't sure it should be strictly regulated.

On the other hand, if sentence ordering decisions were made locally using a top-down transform order, two smaller decisions would be required— ordering the sentences generated by the first transform (that simplifies the *but* clause using rule 4.3):

(a) Mr. Anthony, who runs an employment agency, decries program trading.

(b) But he isn't sure it should be strictly regulated.

and then ordering the sentences generated by the second transform (that simplifies the relative clause using rule 4.4):

(aa) Mr. Anthony decries program trading.

(ab) Mr. Anthony runs an employment agency.

Deciding sentence order locally has the advantage of greatly pruning the search space of possible sentence orders. This results in a more efficient implementation than global sentence ordering. When using the local approach to sentence ordering, a decision needs to be made at every transform application on the optimal order of the two generated simplified sentences. In this thesis, I formulate this as a constraint satisfaction problem.

I discuss the nature of the constraints that decide sentence order in detail in section 5.2.1. In this section, I only formalise the constraint satisfaction problem in general terms, and introduce the notation used in the algorithm for the transformation module (algorithm 4.1 in the next section).

A constraint satisfaction problem (Hentenryck, 1989) is defined by:

1. A set of variables $X_1, X_2, ..., X_n$.
2. For each variable X_i, a finite domain D_i of possible values.
3. A set of constraints C on the values of the variables (for example, if X_i are integers, the constraints could be of the form $X_1 < X_3$ or $X_3 > X_4$ or $X6 = 0$).

A solution to the problem assigns to each variable X_i a value from its domain D_i such that all the constraints are respected. It is possible that a constraint satisfaction problem has multiple solutions, exactly one solution or no solution. In order to select from amongst multiple solutions, the problem definition can be extended to allow for *hard* and *soft* constraints. Then, a solution would assign each variable a value from its domain such that all the hard constraints are respected, and the number of soft constraints respected is maximised.

I treat local sentence ordering as a constraint satisfaction problem where the variables represent the positions of the simplified sentences in the regenerated text and the constraints are expressed in terms of the possible orderings of the two sentences generated by a transform. These constraints arise from RST, as well as from considerations of referential cohesion and connectedness. The details are presented in section 5.2.1, but to illustrate the constraints that arise from RST, consider simplifying the *concession* relation in:

> Mr. Anthony, who runs an employment agency, decries program trading, but he isn't sure it should be strictly regulated.
>
> ↓
>
> (a) Mr. Anthony, who runs an employment agency, decries program trading.
>
> (b) But he isn't sure it should be strictly regulated.

The nucleus of the concession relation (signalled by *but*) should immediately precede the satellite. The precedence is enforced by the constraint $a < b$. To enforce the immediacy, constraints need to be passed down the recursion, so that when recursively simplifying (a):

> (a) Mr. Anthony, who runs an employment agency, decries program trading.
>
> ↓
>
> (aa) Mr. Anthony decries program trading.
> (ab) Mr. Anthony runs an employment agency.

the nucleus sentence (aa) is forced to be last. This can be achieved by passing down the constraint *nucleus is last* when recursively simplifying (a). The final sentence order is then constrained to be:

> Mr. Anthony runs an employment agency. Mr. Anthony decries program trading. But he isn't sure it should be strictly regulated.

The algorithm for recursively applying transforms is described in the next chapter, while the task of sentence ordering is described in section 5.2.1.

4.4 The Algorithm for Transformation Module

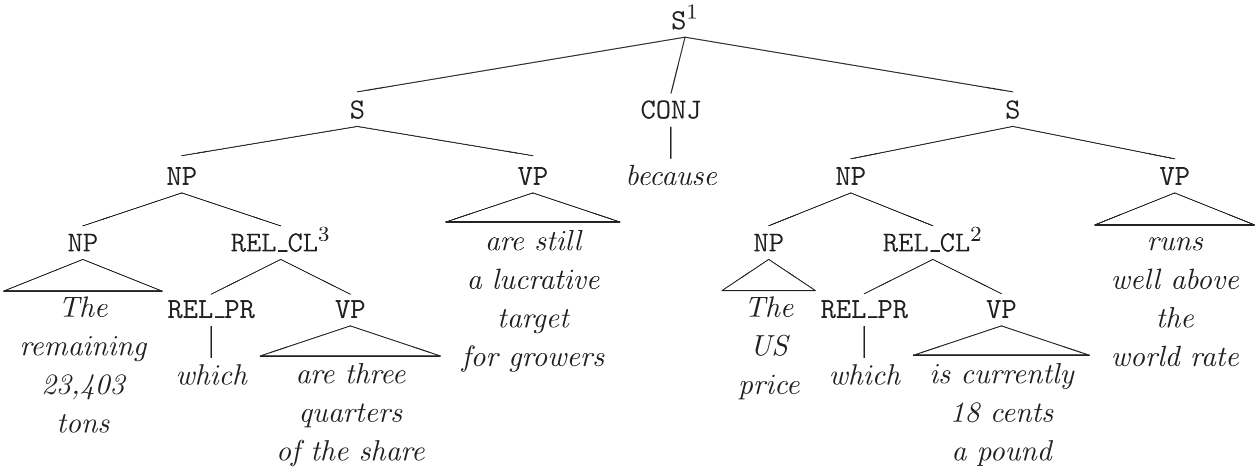
Returning to the issue of transform ordering, both depth-first and top-down left-to-right tree traversal methods are inadequate in practice. The problem arises when a simplification rule results in reversing the original clause order. When this happens, the right branch of the tree needs to be processed before the left branch. This is illustrated in figure 4.3, where depth first search gives the ordering (3,1,2), top-down left-to-right search gives the order (1,3,2) and the required order is (1,2,3). In this example, clause order is reversed because my regeneration stage converts a *(a, justify, b)* relation to a *(b, consequence, a)* relation, in order to be able to use the cue-word *so* (discussed in section 5.2.1).

I now present my algorithm for the transformation stage, that applies transforms in the required order for the examples in figures 4.1-4.3. Algorithm 4.1 takes as input the output of the analysis stage, and outputs the correctly ordered simplified sentences to the output stream. The algorithm works by maintaining a *queue* of simplified-sentence/constraint-set pairs. Step 2(a) is the iterative part of the algorithm that considers each sentence from the input stream in turn. Step 2(b) is the recursive part of the algorithm that transforms a sentence and sends the simplified sentences to the output stream.

In the recursive step, I take the first sentence/constraint-set pair (S, C) in the queue (step iii.A.) and apply a transform R (step iii.B.) to get the simplified sentences S_a and S_b. At this point C contains the constraints that have been passed down from previously applied transforms. The regeneration module is now invoked (step iii.C.). The regeneration module (apart from addressing cue-word selection, referring expression generation and determiner choice) uses the inherited constraints C, new constraints from the rhetorical relation between S_a and S_b, and local cohesion constraints from centering theory to determine the sentence order for S_a and S_b. It also passes down the constraints to C_a and C_b.

If the sentence-ordering constraints cannot be resolved, the transform is not performed, and the original pair (S, C) is pushed back onto the Queue with the mark-up for the failed transform removed (step iii. D.). Otherwise, the sentence/constraint-set pairs (S_a, C_a) and (S_b, C_b) are pushed onto the front of the queue in the correct order (steps iii.E. and iii.F.).

This approach decides sentence order in a top-down manner, such that the queue always contains simplified sentences in the right order. Hence, when the base step of the recursion (ii.A.) is invoked (when the first sentence S in the queue cannot be simplified further),


S1
S
CONJ
S
NP
VP
because
NP
VP
NP
REL_CL3
are still a lucrative target for growers
NP
REL_CL2
runs well above the world rate
The remaining 23,403 tons
REL_PR
VP
The US price
REL_PR
VP
which
are three quarters of the share
which
is currently 18 cents a pound

Original Sentence

[1 The remaining 23,403 tons, [3 which are three quarters of the share], are still a lucrative target for growers because the U.S. price, [2 which is currently 18 cents a pound], runs well above the world rate].

Desired Transformation Sequence

1. The U.S. price, which is currently 18 cents a pound, runs well above the world rate. So, the remaining 23,403 tons, which are three quarters of the share, are still a lucrative target for growers.

2. The U.S. price is currently 18 cents a pound. The U.S. price runs well above the world rate. So, the remaining 23,403 tons, which are three quarters of the share, are still a lucrative target for growers.

3. The U.S. price is currently 18 cents a pound. The U.S. price runs well above the world rate. So, the remaining 23,403 tons are still a lucrative target for growers. The 23,403 tons are three quarters of the share.

Figure 4.3. Inadequacy of top-down left-to-right processing

the sentence S can be sent straight to the output stream.

The algorithm thus ensures that sentences are sent to the output stream in the correct order by optimising constraints locally at each recursive step, rather than by performing a global constraint optimisation at the base step. This procedure has two advantages over global optimisation. Firstly, it cannot result in a situation where clauses that were related in the original sentence are separated by large distances in the transformed text. Secondly, it provides an easy escape route when a transform can't be performed because of conflicting constraints that cannot be resolved simultaneously; in that eventuality, the solution is to not perform that transform. This allows the remaining transforms to be carried out.

Figure 4.4 schematically shows how the transformation stage (algorithm 4.1) interacts with the regeneration stage. Transform-specific issues are resolved by a call to the regeneration stage during transform application. Other discourse level issues like anaphoric

Algorithm 4.1 Transforming sentences recursively

```
Transform_Recursively(Input_Stream, Output_Stream)
  1. Initialise Queue to be empty
  2. WHILE there are sentences left in Input_Stream DO a-b
     (a) IF Queue is empty THEN
         i. PUSH next sentence from Input_Stream onto Queue
         ii. Initialise the associated constraint-set to be empty
     (b) WHILE Queue is not empty DO i-iii
         i. Consider the first sentence/constraint-set pair $(S, C)$ in the queue
         ii. IF $S$ can't be simplified THEN
             A. POP $(S, C)$
             B. Send $S$ to Output_Stream and discard $C$
             C. Fix future anaphoric links (figure 4.4)
         iii. ELSE
             A. POP $(S, C)$
             B. Apply the first (by top-down left-to-right search) simplification rule $R$
                to $S$, obtaining sentences $S_a$ and $S_b$
             C. Invoke the regeneration module to address transform-specific regeneration
                issues (figure 4.4). In particular, it returns the sentence order
                (either (a,b), (b,a) or fail) and passes down the new constraints to $C_a$
                and $C_b$.
             D. IF the sentence order is fail, THEN remove the mark-up for the failed
                transform in $S$ and PUSH $(S, C)$ back onto the Queue.
             E. IF the sentence order is (a,b) THEN PUSH $(S_b, C_b)$ and $(S_a, C_a)$ onto
                Queue in that order.
             F. IF the sentence order is (b,a) THEN PUSH $(S_a, C_a)$ and $(S_b, C_b)$ onto
                Queue in that order.
```

structure are dealt with as a post-process.

I now illustrate, using the example in figure 4.3, how constraints are passed down during the recursion. I start the trace at step 2(b), with the queue containing the original sentence S with an empty constraint set C:

queue:

1. $S = [^l$ The remaining 23,403 tons, $[^m$ which are three quarters of the share], are still a lucrative target for growers because the U.S. price, $[^n$ which is currently 18 cents a pound], runs well above the world rate].
   $C = []$

The first pair (S, C) is popped off the queue (step iii.A.) and the transform is applied

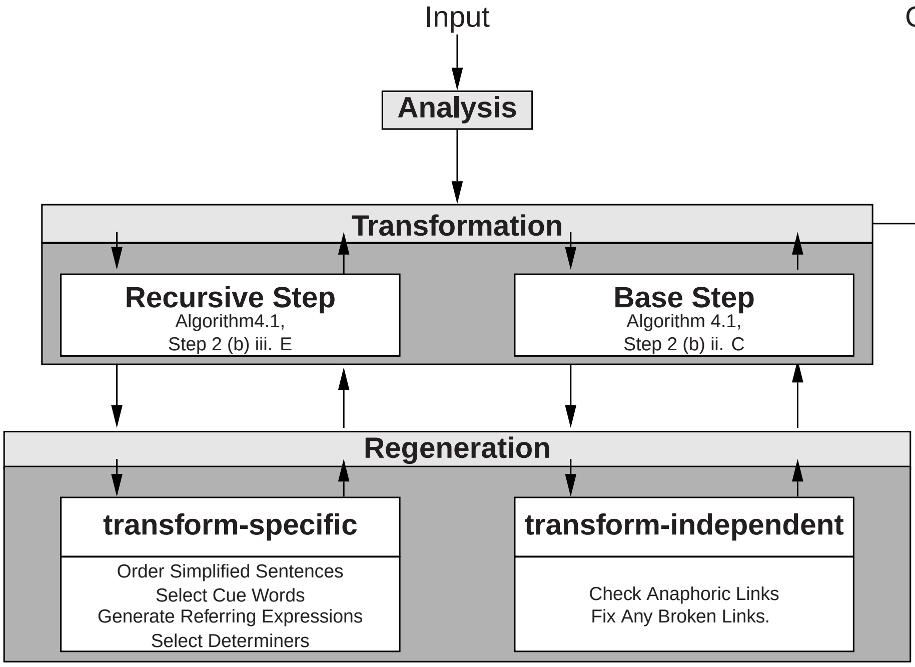


Figure 4.4. The interaction between the transformation and regeneration stages

(step iii.B.):

Transform 1: infix subordination (because) rule 4.3

(a) The remaining 23,403 tons, [^{m} which are three quarters of the share], are still a lucrative target for growers.

(b) The U.S. price, [^{n} which is currently 18 cents a pound], runs well above the world rate.

The regeneration module is then invoked (step iii.C.), which introduces the appropriate cue-word, returns the sentence order (b,a) and passes down the constraints:

C_a={nucleus is first}
C_b={nucleus is last}

The pairs (a, C_a) and (b, C_b) are now pushed onto the front of the queue in that order (step iii.F.), so that the contents of the queue are now:

queue:

1. $S =$ The U.S. price, [^{n} which is currently 18 cents a pound], runs well above the world rate.
   $C = \{$nucleus is last$\}$

2. $S =$ So the remaining 23,403 tons, [^{m} which are three quarters of the share], are still a lucrative target for growers.
   $C = \{$nucleus is first$\}$

I now recurse step 2(b), popping sentence/set pair #1 (S, C) off the queue (step iii.A.), and then applying the step iii.B.:

Transform 2: non-restrictive relative clause rule 4.4

(a) The U.S. price runs well above the world rate.

(b) The U.S. price is currently 18 cents a pound.

In step iii.C., the regeneration module is invoked with C={nucleus is last}. It returns the sentence order (b,a) and passes down the constraints:

C_a={nucleus is last, soft: nucleus is first}
C_b={soft: nucleus is last}

The pairs (a, C_a) and (b, C_b) are now pushed onto the front of the queue in that order (step iii.F.), so that the contents of the queue are now:

queue:

1. $S =$ The U.S. price is currently 18 cents a pound.
   $C =$ {soft: nucleus is last}

2. $S =$ This price runs well above the world rate.
   $C =$ {nucleus is last, soft: nucleus is first}

3. $S =$ So the remaining 23,403 tons, [^{m} which are three quarters of the share], are still a lucrative target for growers.
   $C =$ {nucleus is first}

I now recurse 2(b) again, this time reaching the base case (step ii.A.) and popping sentence 1 from the queue to the output stream. The new queue is now:

queue:

1. $S =$ This price runs well above the world rate.
   $C =$ {nucleus is last, soft: nucleus is first}

2. $S =$ So the remaining 23,403 tons, [^{m} which are three quarters of the share], are still a lucrative target for growers.
   $C =$ {nucleus is first}

and the output stream contains:

output stream:

1. The U.S. price is currently 18 cents a pound.

I recurse 2(b) again, again reaching the base case (step ii.A.) and popping sentence 1 from the queue to the output stream. The new queue is now:

queue:

1. $S =$ So the remaining 23,403 tons, [^{m} which are three quarters of the share], are still a lucrative target for growers.
   $C =$ {nucleus is first}

and the output stream contains:

output stream:

1. The U.S. price is currently 18 cents a pound.
2. This price runs well above the world rate.

The recursion follows a similar route for the remaining sentence on the queue, and at the end of the recursion, the output stream contains:

output stream:

1. The U.S. price is currently 18 cents a pound.
2. This price runs well above the world rate.
3. So the remaining 23,403 tons are still a lucrative target for growers.
4. These 23,403 tons are three quarters of the share.

For the sentences in figures 4.1 and 4.2, my algorithm finds the following sentence orders:

> Mr. Anthony, who runs an employment agency, decries program trading, but he isn't sure it should be strictly regulated.
>
> ↓
>
> Mr. Anthony runs an employment agency. Mr. Anthony decries program trading. But, he isn't sure it should be strictly regulated.

and:

> The meeting, which is expected to draw 20,000 to Bangkok, was going to be held at the Central Plaza Hotel, but the government balked at the hotel's conditions for undertaking necessary expansion.
>
> ↓
>
> The meeting is expected to draw 20,000 to Bangkok. This meeting was going to be held at the Central Plaza Hotel. But the government balked at the hotel's conditions for undertaking necessary expansion.

In these examples, each sentence ordering decision was controlled by a *hard* constraint. In cases where there are no hard constraints (for example, if there are two or three relative clauses and appositives in a sentence), the regeneration stage uses coherence checks along with the soft constraints. This is discussed further in the next chapter (refer to section 5.2.1 on sentence ordering).

5 *Regeneration*

As I have emphasised through this thesis, there are various discourse-level issues that arise when carrying out sentence-level syntactic simplification of the kind described in chapter 4[18]. If these discourse implications are not taken into account, the rewriting could result in a loss of cohesion, making the text harder to read, or even alter its intended meaning; in either case, making the text harder to comprehend. My architecture therefore uses, in addition to the *analysis* and *transformation* stages, a third stage—*regeneration*, that I describe in this chapter.

5.1 Issues of Cohesion and Texture

As outlined in section 1.5, my theory of text simplification splits the regeneration task into the separate issues of preserving conjunctive and anaphoric cohesion. Conjunctive cohesion is addressed using the framework of rhetorical structure theory, while anaphoric cohesion is addressed using a model of attentional state (salience or centering). Figure 5.1 schematically shows how various regeneration issues influence text cohesion.

5.1.1 Conjunctive Cohesion

In section 5.2, I show how the regeneration issues of sentence ordering, cue-word selection and determiner choice can be resolved so as to minimise the adverse affect of text

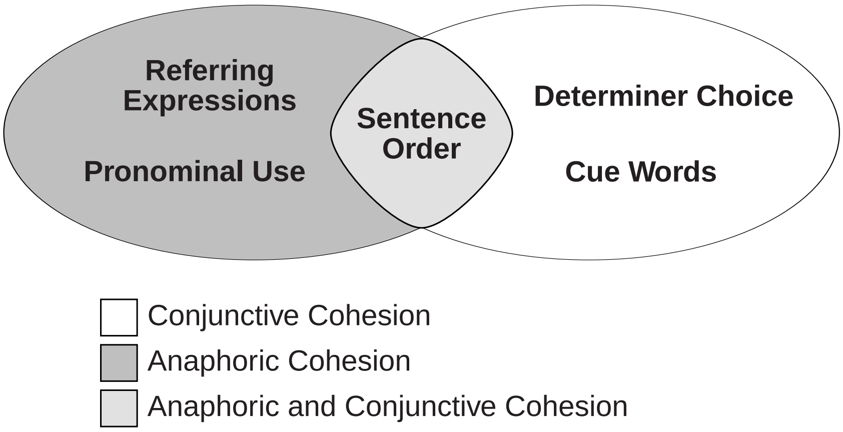


Figure 5.1. Regeneration issues and text cohesion

[18] Parts of this chapter have been published in Siddharthan and Copestake (2002), Siddharthan (2003a), Siddharthan and Copestake (2004) and Siddharthan (To appear).

simplification on conjunctive cohesion. For example, to simplify 5.1(a) to 5.1(b) below, I introduce a new cue word (*so*) and reverse the original clause ordering:

(5.1) a. The "unengageable" element of the welfare population is rising because the city is playing reclassification games.

b. The city is playing reclassification games. So the "unengageable" element of the welfare population is rising.

In addition to rhetorical structure, my approach to sentence-ordering also considers issues of connectedness, as information presented in a disjointed manner, or frequent changes in focus, can make a text difficult to read.

5.1.2 Anaphoric Cohesion

I discuss issues of anaphoric cohesion in section 5.3 (on the use of referring expressions) and section 5.4 (on the use of pronouns). As an illustration of the use of referring expressions, consider example 5.2 below:

(5.2) a. A former ceremonial officer from Derby, who was at the heart of Whitehall's patronage machinery, says there is a general review of the state of the honours list every five years or so.

b. A former ceremonial officer from Derby was at the heart of Whitehall's patronage machinery. **This former ceremonial officer from Derby** says there is a general review of the state of the honours list every five years or so.

b'. A former ceremonial officer from Derby was at the heart of Whitehall's patronage machinery. **This former officer** says there is a general review of the state of the honours list every five years or so.

My rules for simplifying relative clauses and appositive phrases involve the duplication of a noun phrase. The second instance needs to be a referring expression. As illustrated above, I have a choice of referring expressions. Reproducing the entire noun phrase (as in 5.2(b)) can make the text stilted. Further, including too much information in referring expressions can cause unwanted conversational implicatures. For example, 5.2(b) might suggest to the reader that the fact that the officer is from Derby is important to the interpretation of the discourse. It is therefore preferable to use a shorter referring expression, as illustrated in 5.2(b'). I describe my approach to generating referring expressions in section 5.3.

The other important mechanism for reference is pronominalisation. Unfortunately, text-rewriting operations can make the original pronominalisation unacceptable. This is because syntactic transforms can result in discourse referents getting introduced in different orders, with different grammatical functions, and this can make it hard for a reader to correctly resolve pronouns further in the text. For example, if 5.3(a) is naively simplified to 5.3(b), the pronoun *he* becomes difficult to resolve correctly:

(5.3) a. Dr. Knudson found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent, who had necessarily had the disease. Under a microscope he could actually see that a bit of chromosome 13 was missing.

b. Dr. Knudson found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent. This parent had necessarily had the disease. Under a microscope **he** could actually see that a bit of chromosome 13 was missing.

The problem arises because the attentional state at the pronoun *he* has been altered by the simplification process, resulting in the *parent* becoming more salient than *Dr. Knudson*. I discuss techniques for detecting and fixing broken pronominal links in section 5.4.

5.2 Preserving Rhetorical Relations

I now describe how the issues of sentence ordering, cue-word selection and determiner choice can be resolved in a manner that maintains conjunctive cohesion and connectedness.

5.2.1 *Sentence Order*

As described in chapter 4, the sentence-ordering algorithm interacts closely with the transform-ordering algorithm in the transformation stage. When there is more than one construct that can be simplified in a sentence, the transformation stage applies simplification-rules recursively on the sentence, in a top-down manner. Consider:

> $[_m$ Mr. Anthony^1, $[_n$ who$^{\#1}$ runs an employment agency], decries program trading],$[_m$ but he isn't sure it should be strictly regulated].

The top-down rule application leads to the conjunction (construct m) being simplified first, generating the two sentences:

(a) Mr. Anthony^1, $[^n$who$^{\#1}$ runs an employment agency], decries program trading.
(b) {But} he isn't sure it should be strictly regulated.

The sentence-ordering algorithm is called by the transformation stage after each application of a simplification rule. Its role is to decide between the orderings *(a,b)* and *(b,a)* of the two sentences a and b generated by the simplification rule and to constrain the ordering of sentences generated by the recursive simplification of a and b. In this example, it needs to constrain the possible orderings of the sentences aa and ab generated by transform n:

(aa) Mr. Anthony decries program trading.
(ab) Mr. Anthony runs an employment agency.

The sentence-ordering algorithm receives two inputs:

1. A triplet (a, RR, b) of the simplified sentences a and b and the rhetorical relation RR between them.

2. A set C of inherited constraints on sentence order.

It then forms new constraints from the rhetorical relation RR, adds these to the set C of inherited constraints and finds the optimal sentence order. It then initialises the constraint sets C_a and C_b for the simplified sentences a and b. These constraints are then passed down the recursion, as described in section 4.4.

I now describe the constraints that different rhetorical relations RR (described in section 4.2.1) add to the sets C, C_a and C_b. With the exception of three (*justify*, *parenthetical* and *identification*), every rhetorical relation introduces the following constraints:

1. *In C:* a precedes b
2. *In C_a:* the nucleus is last
3. *In C_b:* the nucleus is first

The first constraint is required in order to enforce the correct rhetorical relation between the two simplified sentences. The other two constraints arise because this rhetorical relation held between particular clauses in the original sentence; hence if the simplified sentences a and b get further simplified, it is necessary to enforce the continued adjacency of those clauses. In the example above,

> Mr. Anthony, who runs an employment agency, decries program trading, but he isn't sure it should be strictly regulated.

was simplified twice to give, first:

> (a) Mr. Anthony, who runs an employment agency, decries program trading.
> (b) But he isn't sure it should be strictly regulated.

and then:

> (aa') Mr. Anthony decries program trading.
> (ab') Mr. Anthony runs an employment agency.
> (b') But he isn't sure it should be strictly regulated.

The first constraint introduced by the *but* transform (RR=*concession*) enforces the ordering $a<b$. The second constraint enforces the ordering *aa'*>*ab'* which ensures that the *concession* relation continues to hold between *Mr. Anthony decries program trading* and *he isn't sure it should be strictly regulated.* These constraints ensure that the text is simplified to:

> Mr. Anthony runs an employment agency. Mr. Anthony decries program trading. But he isn't sure it should be strictly regulated.

and not the misleading:

> Mr. Anthony decries program trading. Mr. Anthony runs an employment agency. But he isn't sure it should be strictly regulated.

An exception to these constraints is when RR = *justify.* In this case, the constraints are:

1. *In C:* b precedes a

2. *In C_a:* the nucleus is first
3. *In C_b:* the nucleus is last

This is because I transform the *justify* relation into a *consequence* relation (refer to section 5.2.2 for the rationale) and the *consequence* clause has to be second; for example, I simplify:

> The remaining 23,403 tons are still a lucrative target for growers because the U.S. price runs well above the world rate.

to:

> The U.S. price runs well above the world rate. So the remaining 23,403 tons are still a lucrative target for growers.

The constraints presented thus far are all *hard*; they have to hold in the final sentence order. In contrast, when *RR=parenthetical*, the constraints introduced are *soft*. Parentheticals contain information that is not central to the discourse. This means that there is some flexibility as to where they can be positioned. The sole constraint introduced by parentheticals is:

1. *In C:* soft: a precedes b

This constraint arises because parentheticals (non-restrictive relative clauses and appositives) tend to provide additional information about the noun phrase they attach to. This additional information is better presented in the second sentence. This is a soft constraint; disregarding it causes a change from an elaborative to a more narrative style, but does not make the text misleading or nonsensical; for example, in isolation, 5.4(b') is only marginally (if at all) less acceptable than 5.4(b) below:

(5.4) a. Garret Boone, who teaches art at Earlham College, calls the new structure "just an ugly bridge" and one that blocks the view of a new park below.

b. Garret Boone calls the new structure "just an ugly bridge" and one that blocks the view of a new park below. Garret Boone teaches art at Earlham College.

b'. Garret Boone teaches art at Earlham College. Garret Boone calls the new structure "just an ugly bridge" and one that blocks the view of a new park below.

The final relation that needs to be considered is *RR=identification*, which holds between a restrictive relative clause and the noun phrase it attaches to. The constraint introduced by this relation is:

1. *In C:* soft: b precedes a

This constraint arises because it is preferable to identify the referent of the noun phrase before it is used in the main clause. This constraint encourages the sentence:

> The man who had brought it in for an estimate returned to collect it.

to be simplified as:

> A man had brought it in for an estimate. This man returned to collect it.

The soft constraints introduced by *parenthetical* or *identification* relations can be violated either to enforce a hard constraint or to improve text connectedness.

I now present my algorithm for deciding sentence order. Algorithm 5.1 receives a constraint set C, the simplified sentences a and b and the rhetorical relation RR between them as input from the transformation stage. The algorithm first makes the constraint sets for a and b inherit the constraints from previous transforms that are present in C (step 1). It then uses the rhetorical relation RR to update the constraint sets C, C_a and C_b (step 2) as described previously in this section.

The algorithm then scans the constraint set C for hard constraints (steps 3 and 4). If there are conflicting hard constraints, it returns an error code and the transformation stage aborts that transform. In the case where there is a hard constraint present and there is no conflict, the algorithm returns the order specified by the hard constraint.

In the case where there are no hard constraints to guide sentence order, the algorithm considers issues of connectedness. There are two cases when these issues decide sentence order. The first (step 5) is when the simplified sentences have the form a = `X Y.` and b = `Y Z.` In this case, the sentence order `X Y. Y Z.` (a, b) is judged to be more connected than the order `Y Z. X Y.` (b, a); for example, the ordering (b) is judged more connected than (b') in:

(5.5) a. They will remain on a lower-priority list that includes 17 other countries.

b. (1) They will remain on a lower-priority list. (2) This list includes 17 other countries.

b'. (1) A lower-priority list includes 17 other countries. (2) They will remain on this list.

This can be justified using centering theory. The main assumption is that in the original sentence (a), it is unlikely that the backward-looking center $C_b(a)$ is contained within a relative clause and so $C_b(a)$ is most likely to be the referent of *they*. In that case, the sentence-ordering (b) consists of one center-continuation transition (to sentence 1) and one center-retaining transition (to sentence 2). On the other hand, the sentence-ordering (b') involves a center-shift to sentence 1 and is therefore more disruptive (refer to section 1.6.1).

While centering theory can be used to justify my sentence-ordering decisions, using it to actually make them is impractical, as that would involve having to make a wide range of co-reference decisions. For example, the surrounding text for example 5.5 above is:

> These three countries^{1} aren't completely off the hook, though. They$^{\#1}$ will remain on a lower-priority list^{2} that includes 17 other countries^{3}. Those countries$^{\#3}$ – including Japan, Italy, Canada, Greece and Spain – are still of some concern to the U.S. but are deemed to pose less-serious problems for American patent and copyright owners than those on the "priority" list$^{\#2}$.

Algorithm 5.1 Deciding sentence order

```
Decide-Sentence-Order((a,RR,b),C)
  1. Initialise $C_a$ and $C_b$ to the constraints in $C$
  2. Process $RR$ and update $C$, $C_a$ and $C_b$ (as described earlier in the section)
  3. IF constraint set $C$ contains hard constraints ($a<b$ or $a$ is first or $b$ is last) THEN
     (a) IF there are no conflicting hard constraints THEN RETURN $(a, b)$ and $C_a$ and $C_b$
         ELSE RETURN fail
  4. IF constraint set $C$ contains hard constraints ($b<a$ or $b$ is first or $a$ is last) THEN
     (a) IF there are no conflicting hard constraints THEN RETURN $(b, a)$ and $C_a$ and $C_b$
         ELSE RETURN fail
  5. IF $a = XY.$ and $b = YZ.$ THEN
     (a) Add the constraint soft: nucleus is last to $C_a$ and soft: nucleus is first to $C_b$
     (b) RETURN $(a, b)$ and $C_a$ and $C_b$
  6. IF $a$ can be simplified further or IF constraint set $C$ contains soft constraints ($b < a$
     or $b$ is first or $a$ is second) and no conflicting constraints THEN
     (a) Add the constraint soft: nucleus is first to $C_a$ and soft: nucleus is last to $C_b$
     (b) RETURN $(b, a)$ and $C_a$ and $C_b$
  7. By default:
     (a) Add the constraint soft: nucleus is last to $C_a$ and soft: nucleus is first to $C_b$
     (b) RETURN $(a, b)$ and $C_a$ and $C_b$
```

Finding the backward-looking centers for this example would require co-referencing not just pronouns (like *they*) but also definite references (like *those countries* and *the "priority" list*).

Text can also lose its connectedness if clauses that were adjacent in the original sentence get separated by an intervening sentence. This can happen if sentence *a* contains another construct to be simplified; for example, consider the sentence:

(5.6) a. The agency, ***which*** *is funded through insurance premiums from employers*, insures pension benefits for some 30 million private-sector workers who take part in single-employer pension plans.

that contains two relative clauses. When applying the first transform, the following sentences are generated:

(a) The agency insures pension benefits for some 30 million private-sector workers ***who*** *take part in single-employer pension plans.*

(b) The agency is funded through insurance premiums from employers.

Rhetorical Relation	**Cue-Words**
Concession	but
Anti-Conditional	or
Condition	suppose...then
Justify (→Consequence)	so
And	and
`X`	This `Aux X`

Table 5.1. Cue words that are introduced when simplifying various conjoined clauses.

In this case sentence (a) can be simplified further. If the order (a, b) is returned by the first transform, there are two possibilities for the final sentence ordering:

(5.6) b'. The agency insures pension benefits for some 30 million private-sector workers. These workers take part in single-employer pension plans. The agency is funded through insurance premiums from employers.

b". These workers take part in single-employer pension plans. The agency insures pension benefits for some 30 million private-sector workers. The agency is funded through insurance premiums from employers.

If the first transform returns the order (b, a), it leads to the final sentence ordering:

(5.6) b. The agency is funded through insurance premiums from employers. The agency insures pension benefits for some 30 million private-sector workers. These workers take part in single-employer pension plans.

Again, centering theory can be used to reason that 5.6(b) is preferable to both 5.6(b') and 5.6(b"). Step 6 returns the ordering (b, a) if a can be simplified further, or if there are non-conflicting soft constraints that suggest that order. Otherwise, by default, the order with the nucleus first (a, b) is returned (step 7).

5.2.2 Cue-Word Selection

To preserve the rhetorical relation between conjoined clauses that have been simplified into separate sentences, it is necessary to introduce new cue-words to signal the relation. As described in section 4.2, cue-word selection is resolved using an input from the transformation stage of the form *(a, RR, b)*, where *RR* is the rhetorical relation connecting the two simplified sentences *a* and *b*.

I have a choice of cue-words available for signalling some relations. Williams et al. (2003) conducted experiments on learner readers that showed faster reading times when simple cue-words like *so* and *but* were used instead of other widely used cue-words like *therefore*, *hence* or *however*. Williams et al. (2003) also reported that the presence of punctuation along with the cue-word resulted in faster reading times. I therefore restrict myself to using simple cue-words like *so* for the *consequence* relation and *but* for the *concession* relation and also include punctuation wherever possible.

Table 5.1 gives a list of rhetorical relations and the corresponding cue-words that my algorithm introduces. Every *concession* relation results in a sentence-initial *but* in the second sentence:

(5.7) a. **Though** all these politicians avow their respect for genuine cases, it's the tritest lip service.

b. All these politicians avow their respect for genuine cases. **But**, it's the tritest lip service.

(5.8) a. Teachers often "teach the test" as Mrs. Yeargin did, **although** most are never caught.

b. Teachers often "teach the test" as Mrs. Yeargin did. **But**, most are never caught.

I convert the *justify* relation to a *consequence* relation in order to use the simple cue-word *so*. This also results in reversing the original clause order (refer to section 5.2.1 on sentence-ordering). An example is:

(5.9) a. The federal government suspended sales of U.S. savings bonds **because** Congress hasn't lifted the ceiling on government debt.

b. Congress hasn't lifted the ceiling on government debt. **So**, the federal government suspended sales of U.S. savings bonds.

In section 4.2, I introduced my own rhetorical relations like *when*, *before*, *after*, *and*, *since* and *as* (either due to a lack of granularity in the original Mann and Thompson (1988) relations or due to difficulties with disambiguating the rhetorical relation). For each of these rhetorical relations `X` (with the exception of *and*), I introduce the cue-words *This* `Aux X`. The auxiliary verb `Aux` is either *is* or *was* and is determined from the tense of the nucleus clause; for example, in:

(5.10) a. Kenya was the scene of a major terrorist attack on August 7 1998, **when** a car bomb blast outside the US embassy in Nairobi killed 219 people.

b. Kenya was the scene of a major terrorist attack on August 7 1998. **This was when** a car bomb blast outside the US embassy in Nairobi killed 219 people.

(5.11) a. A more recent novel, " Norwegian Wood ", has sold more than four million copies **since** Kodansha published it in 1987.

b. A more recent novel, " Norwegian Wood ", has sold more than four million copies. **This is since** Kodansha published it in 1987.

(5.12) a. But Sony ultimately took a lesson from the American management books and fired Mr. Katzenstein, **after** he committed the social crime of making an appointment to see the venerable Akio Morita, founder of Sony.

b. But Sony ultimately took a lesson from the American management books and fired Mr. Katzenstein. **This was after** he committed the social crime of making an appointment to see the venerable Akio Morita, founder of Sony.

5.2.3 Determiner Choice

Simplifying relative clauses and appositives results in the duplication of a noun phrase. I need to use a referring expression the second time, a topic I discuss in section 5.3. I also need to decide on what determiners to use. This decision depends on the rhetorical relation between the extracted clause or phrase and the noun phrase it attaches to.

In the non-restrictive case (for either appositives or relative clauses), the rhetorical relation is *RR=parenthetical*. The only constraint here is that there should be a definite determiner in the referring expression. I use *this* or *these* depending on the whether the noun phrase is singular or plural; for example, in:

(5.13) a. A former ceremonial officer, who was at the heart of Whitehall's patronage machinery, said there should be a review of the honours list.

b. A former ceremonial officer said there should be a review of the honours list. **This** officer was at the heart of Whitehall's patronage machinery.

When simplifying restrictive clauses, the rhetorical relations is that of *identification*—identifying a member (or some members) from a larger set. To preserve this, I require an indefinite determiner (*a* or *some*) in the noun phrase that the clause attaches to. This has the effect of introducing the member(s) of the larger set into the discourse:

(5.14) a. The man who had brought it in for an estimate returned to collect it.

b. **A** man had brought it in for an estimate. **This** man returned to collect it.

The indefinite article is not introduced if the noun phrase contains a numerical attribute; for example, in:

(5.15) a. He was involved in two conversions which turned out to be crucial.

b. He was involved in two conversions. **These** conversions turned out to be crucial.

The referring expression contains a definite determiner for the restrictive case as well.

I do introduce or change the determiner in either the original noun phrase or the referring expression if the head noun is a proper noun or if there is an adjectival (possessive) pronoun present (for example, in *his latest book*).

5.2.4 Evaluation

Evaluating issues of conjunctive cohesion is non-trivial. Unlike the evaluations in chapter 3 on *analysis*, there are no gold standards like the Penn WSJ Treebank to compare against. Therefore, the only way to evaluate these regeneration issues is by means of human judgements. There is, however, a fair bit of subjectivity involved in making judgements on issues such as optimal sentence-order or cue-word and determiner selection. And, since neither of the previous attempts at syntactic simplification (described in section 1.4) considered issues of conjunctive cohesion, there is no precedent for evaluation that I can follow.

In section 6.1 of the chapter on *results*, I present an evaluation of the correctness of the simplified sentences generated by my program. In that evaluation, I use three human judges to each evaluate three aspects of the simplified sentences—grammaticality, semantic parity and coherence. In order to evaluate how well by program preserves conjunctive cohesion, I summarise the results for coherence (for details, refer to section 6.1).

The judges were presented with 95 sentences from Guardian news reports and the simplified sentences that my program generated from them. They were asked to judge coherence on a scale of 0 to 3 (0 indicating meaning change, 1 indicating major disruptions in coherence, 2 indicating a minor reduction in coherence and 3 indicating no loss of coherence). For 40% of the 95 examples, all the judges scored 3. However, there was very little agreement between judges for the remaining 60%.

As an indication of how coherent the simplified sentences were, the average of the scores of all the judges over all the examples was 2.43. The average of the three judges' scores was above 2 for 73% of the cases and below 1 for only 8% of the cases.

To try and pin the errors on particular algorithms in my simplification system, I asked two of the judges to revise the simplified sentences (for cases where they had scored less than 3) if they could think up a more cohesive output. Most of the revisions the judges made involved increasing the use of pronouns; for example, the output:

> Argentina's former president was Carlos Menem. Argentina's former president was last night on the brink of throwing in the towel on his re-election bid...

was rewritten by one judge as:

> Argentina's former president was Carlos Menem. He was last night on the brink of throwing in the towel on his re-election bid...

This indicates that simplified text can be difficult to read for people with high reading ages. However, though the lack of pronominalisation makes the text less cohesive, it might still be beneficial to people who have difficulty resolving pronouns.

Among the revisions that could be used to evaluate the algorithms in this section, the two judges (on average) changed sentence order 3 times, cue-words 4 times, auxiliary verbs (*is* to *was* and vice-versa) 4 times and determiners once. However, most of the revisions were of a more semantic nature, and generated sentences that would be beyond the scope of my program. For example, the sentence:

> An anaesthetist who murdered his girlfriend with a Kalashnikov souvenir of his days as an SAS trooper, was struck off the medical register yesterday, five years later.

got simplified by my program to:

> A anaesthetist, was struck off the medical register yesterday, five years later. This anaesthetist murdered his girlfriend with a Kalashnikov souvenir of his days as an SAS trooper.

This was then revised by one judge to:

> An anaesthetist was struck off the medical register yesterday. Five years earlier he murdered his girlfriend with a Kalashnikov souvenir of his days as an SAS trooper.

and by the other judge to:

> A anaesthetist, was struck off the medical register yesterday. This anaesthetist murdered his girlfriend with a Kalashnikov souvenir of his days as an SAS trooper. This happened five years ago.

There were also instances where a judge marked the output as incoherent, but could not think of a coherent way to rewrite it. For example, the sentence:

> The hardliners, who have blocked attempts at reform by President Mohammad Khatami and his allies, have drawn a different lesson from the Iraq conflict.

was simplified by my program to:

> The hardliners have drawn a different lesson from the Iraq conflict. These hardliners have blocked attempts at reform by President Mohammad Khatami and his allies.

One judge decided that it was not possible to preserve the subtleties of the original, and despite giving it a low coherence score, did not offer a revision.

To summarise, an average score of 2.43 suggested that for most of the sentences, the loss in coherence was minor. However, when there was a loss in coherence, it tended to arise from subtleties at the semantic level. This meant that most of the revisions suggested by the judges required more involved rewrites than could be achieved by manipulating sentence order, determiners, cue-words or tense.

5.2.5 A Comparison with Constraint Based Text Planning

As discussed in section 5.2.1, I formulate the sentence ordering task as a constraint satisfaction problem (cf. section 4.3.1). The constraint satisfaction approach has previously been used in planning text structure in natural language generation. I now compare the use of constraints in text generation from rhetorical structure (using the ICONOCLAST (Power, 2000) project as a case study) with my use of constraints for preserving rhetorical structure during text regeneration.

A key issue in natural language generation is the realisation of a discourse structure, represented as a RST tree, by a text structure, in which the content of the discourse structure is divided into sentences, paragraphs, itemised lists and other textual units. In general, there are many possible text structures that can realise a discourse structure; the task is to enumerate them and select the best candidate. Power (2000) described how this task could be formalised as a constraint satisfaction problem. The rules of text formation (for example, that sentences should not contain paragraphs) were formalised as hard constraints. The potential solutions (text structures that correctly realise a rhetorical structure) were then enumerated by solving these constraints. In order to further constrain the solution, Power (2000) included a set of soft stylistic constraints; for example, that single sentence paragraphs are undesirable.

Power (2000) assigned four variables (TEXT-LEVEL, INDENT, ORDER, CONNECTIVE) to each node of the rhetorical structure tree. TEXT-LEVEL was an integer between 0 and 4 that denoted:

0: text phrase

1: text clause

2: text sentence

3: paragraph

4: section

INDENT was the level of indentation of the text and took integer values $(0, 1, 2...)$. ORDER was an integer less than N, the number of sister nodes. CONNECTIVE was a linguistic cue (for example, *however*, *since* or *consequently*).

A solution then involved assigning values to these four variables at each node in the rhetorical structure tree, without violating any hard constraints. Some constraints arose from the desired structure of the text; for example, the root node should have a higher TEXT-LEVEL than its daughters, sister nodes should have identical TEXT-LEVELs and sister nodes should have different ORDERs. In addition, the choice of the discourse connective could impose further constraints. For example, if the *cause* relation was expressed by CONNECTIVE=*consequently*, the satellite had to have a lower ORDER than the nucleus and the TEXT-LEVEL values had to be greater than zero. In addition, it was possible to constrain the solution using various stylistic soft constraints; for example, imposing TEXT-LEVEL$\neq 1$ results in sentences without semi-colons, imposing ORDER=1 on the satellite node of a relation results in a style where the nucleus is always presented first and the constraint that when TEXT-LEVEL=2 there is at least one sister node present prevents paragraphs that contain only one sentence.

The ICONOCLAST approach to text structuring is not dissimilar to that described in this thesis in sections 4.3.1 and 5.2.1. However, my approach only requires me to consider text-sentences (TEXT-LEVEL=2). Further, I do not consider typographic features like indentation. On the other hand, Power (2000) do not offer an account of relative clauses or apposition and only consider relations that can be realised by a conjunction. In offering a treatment of relative clauses and apposition, I have in this thesis used the constraint satisfaction approach to combine constraints arising from considerations of referential cohesion and text connectedness (modelled by centering theory) with those arising from considerations of conjunctive cohesion (modelled by RST).

5.3 Generating Referring Expressions

The previous section dealt with the issue of preserving conjunctive cohesion. I now turn my attention to issues of anaphoric cohesion. In this section, I consider the use of referring expressions as an anaphoric device. Then, in section 5.4, I consider issues relating to pronominalisation in rewritten text.

When splitting a sentence into two by dis-embedding a relative clause, I need to provide the dis-embedded clause with a subject. The referent noun phrase hence gets duplicated, occurring once in each simplified sentence. This phenomenon also occurs when simplifying appositives. I now need to generate a referring expression the second time, as duplicating the whole noun phrase can make the text stilted and cause unwanted conversational implicatures. For example, contrast 5.16(b) with 5.16(c):

(5.16) a. 'The pace of life was slower in those days,' says 51-year-old Cathy Tinsall, *who had five children*.

b. 'The pace of life was slower in those days,' says 51-year-old Cathy Tinsall. Cathy Tinsall had five children.

c. 'The pace of life was slower in those days,' says 51-year-old Cathy Tinsall. 51-year-old Cathy Tinsall had five children.

5.16(c), apart from sounding stilted, emphasises Cathy Tinsall's age. This might, for example, inadvertently suggest to the reader that the relationship between her age and her having five children is important. In general, including too much information in the referring expression can convey unwanted and possibly wrong conversational implicatures.

Referring-expression generation is an important aspect of natural-language generation. When a definite noun phrase is used in text to refer to an entity, it needs to contain enough information to help the reader correctly identify the referent. This can be achieved by including either adjectives (attributes of the referent) or prepositional phrases (relations between the referent and other entities) in the referring expression. The referring-expression problem is then that of finding the shortest description that succeeds in differentiating the referent entity from all other entities in context.

In section 5.3.1, I describe the *incremental algorithm* (Reiter and Dale, 1992) for selecting attributes and describe various problems with it. These problems are shared by other existing approaches to attribute selection, which make similar assumptions. I then present my algorithm for attribute selection in section 5.3.2 and discuss how it overcomes the drawbacks of previous approaches. I discuss existing approaches to selecting relations in section 5.3.5 and present my approach to relational descriptions in section 5.3.6. I then present a corpus-based evaluation of my algorithm in section 5.3.9.

5.3.1 The Background to Attribute Selection

The *incremental algorithm* (Reiter and Dale, 1992) is the most widely discussed attribute selection algorithm. It takes as input the entity (e) that needs to be referred to and a *contrast set* (C) of *distractors* (other entities that could be confused with the intended referent). Entities are represented as attribute value matrices (AVMs). The algorithm also takes as input a `*preferred-attributes*` list that contains, in order of preference, the attributes that human writers use to reference objects. For the example in their paper (that deals with entities like *the small black dog, the white cat...*), the preference might be [colour, size, shape, ...]. The algorithm then keeps adding attributes from `*preferred-attributes*` that rule out at least one entity in the contrast set to the referring set until all the entities in the contrast set have been ruled out.

It is instructive to look at how the incremental algorithm works. Consider an example where a *large brown dog* needs to be referred to. The contrast set contains a *large black dog*. These are represented by the AVMs shown below:

$$e = \begin{bmatrix} \texttt{type} & \textit{dog} \\ \texttt{size} & \textit{large} \\ \texttt{colour} & \textit{brown} \end{bmatrix} \qquad\qquad C = \left\{ \begin{bmatrix} \texttt{type} & \textit{dog} \\ \texttt{size} & \textit{large} \\ \texttt{colour} & \textit{black} \end{bmatrix} \right\}$$

Assuming that the `*preferred-attributes*` list is `[size, colour, ...]`, the algorithm would first compare the values of the `size` attribute (both *large*), disregard that attribute as not being discriminating, compare the values of the `colour` attribute and return *the brown dog*.

Unfortunately, the incremental algorithm is unsuitable for open domains because it assumes the following:

1. A classification scheme for attributes exists
2. The values that attributes take are mutually exclusive
3. Linguistic realisations of attributes are unambiguous

All these assumptions are violated when I move from generation in a very restricted domain to generation or regeneration in an open domain. Adjective classification is a hard problem and there is no sensible classification scheme that can be used when dealing with an open domain like newspaper text. Even if I had such a scheme, I would not be able to assume the mutual exclusivity of values; for example, I might end up comparing [`size` *big*] with [`size` *large*] or [`colour` *dark*] with [`colour` *black*]. Further, selecting attributes at the semantic level is risky because their linguistic realisation might be ambiguous and most of the common adjectives are polysemous (See example 1 in section 5.3.2).

My alternative algorithm measures the relatedness of adjectives, rather than deciding if two of them are the same or not (section 5.3.2). It works at the level of words, not their semantic labels. Further, it treats discriminating power as only one criteria for selecting attributes and allows for the easy incorporation of other considerations like reference modification and the reader's comprehension skills (section 5.3.4).

5.3.2 *My Approach*

In order to quantify discriminating power, I define the following three quotients:

Similarity Quotient (SQ)

I define *similarity* as transitive synonymy. The idea is that a synonym of a synonym is a synonym, and the level of synonymy between two adjectives depends on how many times I have to chain through WordNet synonymy lists (refer to section 1.7.1 for an overview of

WordNet) to get from one to the other. Suppose I need to find a referring expression for e_0. For each adjective a_j describing e_0, I calculate a similarity quotient SQ_j by initialising it to 0, forming a set of WordNet synonyms S_1 of a_j, forming a synonymy set S_2 containing all the WordNet synonyms of all the adjectives in S_1 and forming S_3 from S_2 similarly. Now for each adjective describing any distractor, I increment SQ_j by 4 if it is present in S_1, by 2 if it is present in S_2, and by 1 if it is present in S_3. SQ_j now measures how similar a_j is to other adjectives describing distractors.

Contrastive Quotient (CQ)

Similarly, I define *contrastive* as transitive antonymy. I form the set C_1 of strict WordNet antonyms of a_j, C_2 of strict WordNet antonyms of members of S_1 and WordNet synonyms of members of C_1 and C_3 similarly from S_2 and C_2. I now initialise CQ_j to zero and for each adjective describing each distractor, add $w =\in \{4, 2, 1\}$ to CQ_j, depending on whether it is a member of C_1, C_2 or C_3. CQ_j now measures how contrastive a_j is to other adjectives describing distractors.

Discriminating Quotient (DQ)

An attribute that has a high value of SQ has bad discriminating power. An attribute that has a high value of CQ has good discriminating power. I can now define the Discriminating Quotient (DQ) as $DQ = CQ - SQ$. This gives me an order (decreasing DQs) in which to incorporate attributes. I demonstrate my approach with two examples.

Example 1

Suppose I need to refer to e_1 when the contrast set C contains e_2 in:

$$e_1 = \begin{bmatrix} \texttt{type} & \textit{president} \\ \texttt{age} & \textit{old} \\ \texttt{tenure} & \textit{current} \end{bmatrix} \qquad C = \left\{ e_2 = \begin{bmatrix} \texttt{type} & \textit{president} \\ \texttt{age} & \textit{young} \\ \texttt{tenure} & \textit{past} \end{bmatrix} \right\}$$

If I followed the strict typing system used by previous algorithms, to refer to e_1 I would compare the `age` attributes and rule out e_2 and generate *the old president*. This expression is ambiguous since *old* can also mean *previous*. Models that select attributes at the semantic level will run into trouble when their linguistic realisations are ambiguous. In contrast, my algorithm successfully picks *the current president* as *current* has a higher DQ than *old*:

attribute	**distractor**	**CQ**	**SQ**	**DQ**
old	e2{young, past}	4	4	0
current	e2{young, past}	2	0	2

In this example, *current* is a WordNet synonym of *present*, which is a WordNet antonym of *past*. *Old* is a WordNet antonym of *young* and a WordNet synonym of *past*.

Example 2

Assume I have four dogs in context: `e1`(a large brown dog), `e2`(a small black dog), `e3`(a tiny white dog) and `e4`(a big dark dog). To refer to `e4`, for each of its attributes, I calculate the three quotients with respect to `e1`,`e2` and `e3`:

attribute	**distractor**	**CQ**	**SQ**	**DQ**
big	`e1`{large, brown}	0	4	-4
big	`e2`{small, black}	4	0	4
big	`e3`{tiny, white}	1	0	1
big	TOTAL	5	4	1
dark	`e1`{large, brown}	0	0	0
dark	`e2`{small, black}	1	4	-3
dark	`e3`{tiny, white}	2	1	1
dark	TOTAL	3	5	-2

Overall, *big* has a higher discriminating power (1) than *dark* (-2). I therefore pick *big* and rule out all the distractors that *big* has a positive *DQ* for (in this case, `e2` and `e3`). `e1` is the only distractor left. And I need to pick *dark* because *big* has a negative *DQ* for `e1` and *dark* doesn't.

If I had to refer to `e3`, I would end up with simply *the white dog* as *white* has a higher overall *DQ* (6) than *tiny* (1) and rules out `e2` and `e4`. As *white*'s *DQ* with the only remaining distractor `e1` is non-negative, I can assume it to be sufficiently discriminating:

attribute	**distractor**	**CQ**	**SQ**	**DQ**
tiny	`e1`{large, brown}	1	0	1
tiny	`e2`{small, black}	0	1	-1
tiny	`e4`{big, dark}	1	0	1
tiny	TOTAL	2	1	1
white	`e1`{large, brown}	0	0	0
white	`e2`{small, black}	4	0	4
white	`e4`{big, dark}	2	0	2
white	TOTAL	6	0	6

5.3.3 *Justifying my Algorithm*

The psycholinguistic justification for the incremental algorithm hinges on two premises:

1. Humans build up referring expressions incrementally
2. There is a preferred order in which humans select attributes (e.g., colour>shape>size...)

My algorithm is also incremental. However, there is a subtle departure from premise 2. I assume that speakers pick out attributes that are distinctive in context. Averaged over contexts, some attributes have more discriminating power than others (largely because of the way people visualise entities) and premise 2 is an approximation to my approach.

Incremental Algorithm	My Algorithm	Optimal Algorithm
$O(nN)$	$O(n^2N)$	$O(n2^N)$

Table 5.2. The computational complexity of the incremental algorithm (Reiter and Dale, 1992), my algorithm and an optimal algorithm (such as Reiter, 1990).

I now quantify the extra effort I am making to identify attributes that "stand out" *in a given context*. Let N be the maximum number of entities in the contrast set and n be the maximum number of attributes per entity. Table 5.2 compares the computational complexity of an optimal algorithm (such as Reiter (1990)), my algorithm and the incremental algorithm.

Both the incremental algorithm and my algorithm are linear in the number of entities N. This is because neither algorithm allows backtracking; an attribute, once selected, cannot be discarded. In contrast, an optimal search requires $O(2^N)$ comparisons. As my algorithm compares each attribute of the discourse referent with every attribute of every distractor, it is quadratic in n. The incremental algorithm, that compares each attribute of the discourse referent with only one attribute per distractor, is linear in n. This increase in complexity from n to n^2 is insignificant for my domain, as noun phrases in news reports rarely contain more than two or three adjectives.

5.3.4 A Few Extensions

Previous work on generating referring expressions has focused on selecting attributes that help to uniquely identify an entity in the presence of other entities. Discriminating power is, however, only one of many considerations. A major advantage of my approach is that it is easy to incorporate considerations other than discriminating power into the attribute selection process. I discuss three of them below.

Reference Modifying Attributes

The analysis thus far has assumed that all attributes modify the referent rather than the reference to the referent. However, for example, if `e1` is *an alleged murderer*, the attribute *alleged* modifies the reference *murderer* rather than the referent `e1` and referring to `e1` as *the murderer* would be factually incorrect. I can handle reference modifying adjectives trivially by adding a large positive weight to their *DQ*s. This will have the effect of forcing them to be selected in the referring expression.

Reader's Comprehension Skills

I can specify a user-dependent *DQ* cut-off for inclusion of adjectives. For example, for very low reading age readers, I could include every adjective with a non-negative *DQ*. Increasing the cut-off would result in fewer adjectives being included. Alternatively, I could weight *DQ*s according to how common the adjective is. This can be measured using frequency counts on a corpus. The main intuition I use is that uncommon adjectives have more discriminating power than common adjectives. However, they are also more likely to be incomprehensible to people with low reading ages. If I give uncommon adjectives higher weights, I will end up with referring expressions containing fewer, though harder

Entity	**Distractors**
first half-free Soviet *vote*	fair *elections* in the GDR
military construction *bill*	fiscal *measure*
copper *consumption*	declining *use*
cunning *ploy*	public education *gambit*
steep *fall* in currency	*drop* in market stock
permanent *insurance*	death benefit *coverage*

Table 5.3. Examples of distractors from newspaper text

to understand, adjectives. This is ideal for readers with high reading ages. On the other hand, if I flip the weights, so that common adjectives get higher weights, I will end up with referring expressions containing many simple adjectives. This is ideal for people with low reading ages.

Incorporating Salience

The incremental algorithm assumes the availability of a contrast set of distractors and does not provide an algorithm for constructing and updating it. The contrast set, in general, needs to take context into account, though Dale (1992) suggests that for some domains (for example, constructing cooking recipes) the total number of entities is so small that the entire global entity set can be used as the contrast set.

Krahmer and Theune (2002) provide a counter-example, suggesting that if they were reporting a dog show with a hundred dogs and used the global entity set for the contrast set, even if they were talking about one particular dog, they would always have to refer to it by its full description (e.g. large black male long-haired sausage dog). So in general, a *context set* is required, rather than a *global entity set*, in order to reduce the number of distractors that need to be distinguished from the referent.

Krahmer and Theune (2002) proposed an extension to the incremental algorithm which treated the context set as a combination of a discourse domain and a salience function. Their algorithm for deciding salience combined the centering theory approach of Grosz et al. (1995) and the focusing theory approach of Hajicova (1993).

Incorporating salience into my algorithm is trivial. In section 5.3.2, I described how to compute the quotients SQ and CQ for an attribute. This was done by adding an amount $w \in \{4, 2, 1\}$ to the relevant quotient each time a distractor's attribute was discovered in a synonym or antonym set. I can incorporate salience by weighting w with the salience of the distractor whose attribute I am considering. This will result in attributes with high discriminating power with regard to more salient distractors getting selected first in the incremental process. However, for the evaluation in section 5.3.9, I do not consider salience. This is because my input is newspaper articles and I have found empirically that there are rarely more than three distractors.

To form the contrast set for NP_o, I identify all the noun phrases in a discourse window of four sentences and select potential distractors among them. I consider two cases separately.

If NP_o is indefinite, it is being newly introduced into the discourse and any noun phrase previously mentioned within the discourse window that has a similar lexical head (determined by WordNet synonym sets) is a distractor.

If NP_o is definite, I want to exclude any noun phrases that it co-refers with from my contrast set. If the attribute set of a previously mentioned noun phrase with similar lexical head (NP_i) is a superset of the attribute set of NP_o, I assume that NP_o co-refers with NP_i and exclude NP_i from the contrast set. Any other noun phrase previously mentioned within the discourse window that has a similar lexical head is a distractor.

Irrespective of whether NP_o is definite or indefinite, I exclude any noun phrase NP_j that appears in the window after NP_o whose attribute set is a subset of NP_o's.

Table 5.3 gives some examples of distractors that my program found during the evaluation (section 5.3.9).

5.3.5 The Background to Selecting Relations

Semantically, *attributes* describe an entity (eg. *the small grey dog*) and *relations* relate an entity to other entities (eg. *the dog in the big bin*). Relations are troublesome because in relating an entity e_o to e_1, a referring expression needs to be (recursively) generated for e_1. The incremental algorithm does not consider relations and the referring expression is constructed out of only attributes. It is difficult to imagine how relational descriptions can be incorporated in the incremental framework of the Reiter and Dale (1992) algorithm, where the order of incorporation of modifiers is predetermined according to a classification system. The Dale and Haddock (1991) algorithm allows for relational descriptions but involves exponential global search. An important difference between the Reiter and Dale (1992) incremental algorithm and my incremental approach is that my approach computes the order in which attributes are incorporated on the fly, by quantifying their utility through the quotient DQ. This makes it easy for me to extend my algorithm to handle relations because I can compute DQs for relations in much the same way as I did for attributes. I present my treatment of relations in section 5.3.6.

An interesting approach to relational descriptions is provided by Krahmer et al. (2003), who model the problem as a graph with nodes corresponding to entities, edges from a node to itself representing attributes and edges between nodes representing relations. Generating a referring expression then corresponds to identifying a subgraph that uniquely matches a node. A feature of this approach is the unified treatment of relations and attributes, both of which are represented as graph edges. Figure 5.2 shows a scene with two dogs (`d1` and `d2`) and a bin (`b1`) and the graph describing this scene. Figure 5.3 shows the minimal referring expression for `d1`.

It needs to be emphasised, however, that graphs are only a representation (an alternative to AVMs) and are not an algorithm. The algorithm for generating the referring expression is actually the subgraph-matching routine, which still faces the problems described in section 5.3.1 when comparing edges, as these edges have semantic labels that are equivalent to the attributes and relations in the AVM formalism. Further, the process of selecting edges is not incremental when edges between nodes (relations) are allowed, hence the subgraph-matching algorithm has exponential complexity.

Another problem with the (Krahmer et al., 2003) approach is that the comparison is purely structural. A purely structural comparison will work if attributes are adjectives.

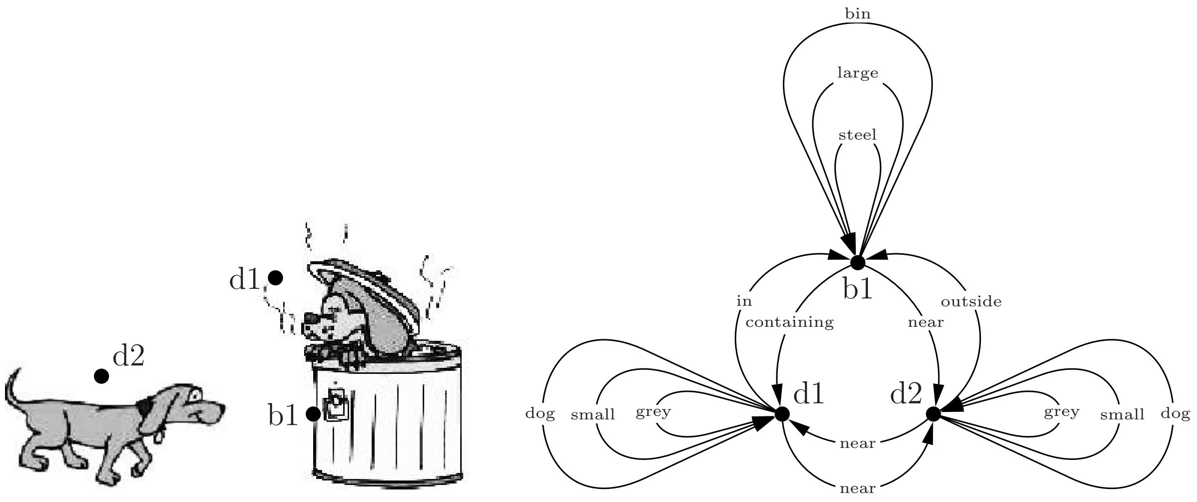


Figure 5.2. Graph representation of two dogs and a bin

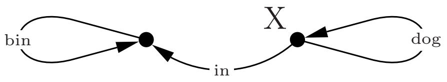
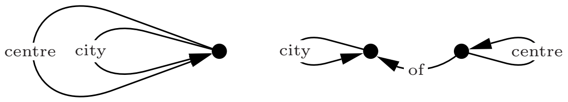


Figure 5.3. Minimal subgraph uniquely matching `d1`

However, many nominals introduced through relations can also be introduced attributively. Purely structural matching cannot detect the similarity between, for example:

1. the city centre / the centre of the city
2. the IBM president / the president of IBM
3. a London company / a company from London
4. a love song / a song about love

The graph structures for the first example are:



I therefore need to compare nominals irrespective of whether they appear attributively or relationally. This is an involved task and I discuss it in section 5.3.8 as an extension to my algorithm.

Incorporating relations in a referring expression is more expensive (in terms of length) than incorporating attributes as their linguistic realisation is a phrase rather than a word. This is modelled implicitly in the graph representation, where an attribute is represented as a single edge while a relation is represented as three. However, treating attributes and relations in a unified framework might not be appropriate because they often serve different discourse functions.

Attributes are usually used to *identify* an entity while relations, in most cases, serve to *locate* an entity. This needs to be taken into account when generating a referring expression. For example, in a newspaper article, the main purpose of a referring expression is to uniquely reference an entity. I therefore want the shortest description, irrespective of how many attributes and relations it contains. However, if I were generating instructions for using a piece of machinery, I might want to include both attributes and relations; so, to instruct the user to switch on the power, I might say "switch on the red button on the top-left corner". This would help the user locate the switch ("top-left corner") and identify it ("red"). If I were helping a chef find the cooking salt in a kitchen, I might want to use only relations because the chef knows what salt looks like. The twelve word long phrase "The salt behind the corn flakes on the shelf above the fridge" is, in this context, preferable to the shorter "the fine white crystals", even if both expressions uniquely identify the salt.

A general purpose approach to generating referring expressions has to be flexible enough to permit a discourse plan to dictate what kind of referring expression it requires. This is an important criteria for me when designing my approach.

5.3.6 My Approach to Relations

Suppose I need to compute the three quotients for the relation $[prep_o\ e_o]$. I consider each entity e_i in the contrast set in turn. If e_i does not have a $prep_o$ relation then the relation is useful and I increment CQ by 4. If e_i has a $prep_o$ relation then two cases arise. If the object of e_i's $prep_o$ relation is e_o then I increment SQ by 4. If it is not e_o, the relation is useful and I increment CQ by 4. This is an efficient non-recursive way of computing the quotients CQ and SQ for relations. I now discuss how to calculate DQ. For attributes, I defined $DQ = CQ - SQ$. However, as the linguistic realisation of a relation is a phrase and not a word, I would like to normalise the discriminating power of a relation with the length of its linguistic realisation. Calculating the length involves recursively generating referring expressions for the object of the preposition, an expensive task that I want to avoid unless I am actually using that relation in the final referring expression. I therefore initially approximate the length as follows. The realisation of a relation $[prep_o\ e_o]$ consists of $prep_o$, a determiner and the referring expression for e_o. If none of e_o's distractors have a $prep_o$ relation then I only require the head noun of the object in the referring expression and $length = 3$. If n distractors contain a $prep_o$ relation with a non-e_o object, I set $length = 3 + n$. This is an approximation to the length of the realisation of the relation that assumes one extra word per distractor. I now define $DQ = (CQ - SQ)/length$.

If the discourse plan requires the algorithm to preferentially select relations or attributes, I can add a positive amount α to their DQs. So the final formula is $DQ = (CQ - SQ)/length + \alpha$, where $length = 1$ for attributes and by default $\alpha = 0$ for both relations and attributes.

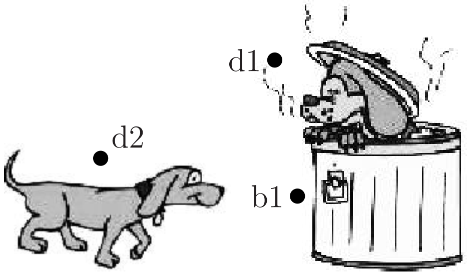
5.3.7 The Complete Algorithm

Algorithm 5.2 generates a referring expression for *Entity*. As it recurses, it keeps track of entities it has used up in order to avoid entering loops like *the dog in the bin containing the dog in the bin....* To generate a referring expression for an entity, the algorithm calculates the *DQ*s for all its attributes and approximates the *DQ*s for all its relations (step 2). It then forms the **preferred** list (step 3) and constructs the referring expression by adding elements from **preferred** till the contrast set is empty (step 4).

Algorithm 5.2 Generating referring expressions

```
Generate-Referring-Expression(Entity, ContrastSet, UsedEntities)
1. IF $ContrastSet = []$ THEN RETURN $\{Entity.head\}$
2. Calculate $CQ$, $SQ$ and $DQ$ for each attribute and relation of Entity (as in Sec 5.3.2
   and 5.3.6)
3. Let *preferred* be the list of attributes/ relations sorted in decreasing order of DQs.
   FOR each element (Mod) of *preferred* DO steps 4, 5 and 6:
4. IF $ContrastSet = []$ THEN RETURN $RefExp \cup \{Entity.head\}$
5. IF Mod is an Attribute THEN
   (a) LET $RefExp = \{Mod\} \cup RefExp$
   (b) Remove from ContrastSet, any entities Mod rules out
6. IF Mod is a Relation $[prep_i\ e_i]$ THEN
   (a) IF $e_i \in UsedEntities$ THEN
         i. Set $DQ = -\infty$
        ii. Move Mod to the end of the *preferred* list
       ELSE
         i. LET ContrastSet2 be the set of non-$e_i$ entities that are the objects of $prep_i$
            relations in members of ContrastSet
        ii. LET $RE$ = generate-referring-exp($e_i$,
                ContrastSet2, $\{e_i\} \cup UsedEntities$)
       iii. recalculate $DQ$ using $length = 2 + length(RE)$
        iv. IF position in *preferred* is lowered THEN re-sort *preferred*
            ELSE
             ($\alpha$) SET $RefExp = RefExp \cup$
                   $\{[prep_i|determiner|RE]\}$
             ($\beta$) Remove from ContrastSet, any
                   entities that Mod rules out
7. RETURN $RefExp \cup \{Entity.head\}$
```

This is straightforward for attributes (step 5). For relations (step 6), it needs to recursively generate the prepositional phrase first. It checks that it hasn't entered a loop (step 6a), generates a new contrast set for the object of the relation (step 6(a)i), recursively



$$d1 = \begin{bmatrix} \texttt{head} & \textit{dog} \\ \texttt{attrib} & [\textit{small}, \\ & \textit{grey}] \\ \texttt{in} & \textit{b1} \\ \texttt{near} & \textit{d2} \end{bmatrix} \quad d2 = \begin{bmatrix} \texttt{head} & \textit{dog} \\ \texttt{attrib} & [\textit{small}, \\ & \textit{grey}] \\ \texttt{outside} & \textit{b1} \\ \texttt{near} & \textit{d1} \end{bmatrix} \quad b1 = \begin{bmatrix} \texttt{head} & \textit{bin} \\ \texttt{attrib} & [\textit{large}, \\ & \textit{steel}] \\ \texttt{containing} & \textit{d1} \\ \texttt{near} & \textit{d2} \end{bmatrix}$$

Figure 5.4. AVM representation of two dogs and a bin

generates a referring expression for the object of the preposition (step 6(a)ii), recalculates *DQ* (step 6(a)iii) and either incorporates the relation in the referring expression or shifts the relation down the **preferred** list (step 6(a)iv). If, after incorporating all the attributes and relations, the contrast set is still non-empty, the algorithm returns the best expression it can find (step 7).

An Example

I now trace the algorithm above as it generates a referring expression for `d1` in figure 5.4.

call generate-ref-exp(d1,[d2],[])

- step 1: *ContrastSet* is not empty
- step 2: $DQ_{small} = -4$, $DQ_{grey} = -4$
  $DQ_{[in\ b1]} = 4/3$, $DQ_{[near\ d2]} = 4/4$
- step 3: **preferred** = [[in `b1`], [near `d2`], small, grey]
- Iteration 1 — *mod* = [in `b1`]
  - step 6(a)i: *ContrastSet2* = []
  - step 6(a)ii: **call generate-ref-exp(b1,[],[d1])**
    - ∗ step 1: *ContrastSet* = []
      return {bin}
  - step 6(a)iii: $DQ_{[in\ b1]} = 4/3$
  - step 6(a)ivα: *RefExp* = {[in, the, {bin}]}
  - step 6(a)ivβ: *ContrastSet* = []
- Iteration 2 — *mod* = [near `d2`]
  - step 4: *ContrastSet* = []
    return {[in the {bin}], dog}

The algorithm presented above tries to return the shortest referring expression that uniquely identifies an entity. If the scene in figure 5.4 were cluttered with bins, the algorithm would still refer to d1 as *the dog in the bin* as there is only one dog that is in a bin. The user gets no help in locating the bin. If helping the user locate entities is important to the discourse plan, I need to change step 6(a)i so that the contrast set includes all *bins* in context, not just *bins* that are objects of *in* relations of distractors of d1.

5.3.8 Handling Nominals

The analysis so far has assumed that attributes are adjectives. However, many nominals introduced through relations can also be introduced attributively, for example:

- the centre of the city ↔ the city centre
- the president of IBM ↔ the IBM president
- a company from East London ↔ an East London company

This means that I need to compare nominal attributes with the objects of relations and vice versa. Algorithm 5.3 calculates DQ for a nominal attribute a_{nom} of entity e_o.

Algorithm 5.3 Calculating DQ for nominals

```
Calculate-DQ-for-Nominals
1. FOR each distractor $e_i$ of $e_o$ DO
    (a) IF $a_{nom}$ is similar to any nominal attribute of $e_i$ THEN $SQ = SQ + 4$
    (b) IF $a_{nom}$ is similar to the head noun of the object of any relation of $e_i$ THEN
        i. $SQ = SQ + 4$
        ii. flatten that relation for $e_i$, i.e, add the attributes of the object of the
            relation to the attribute list for $e_i$
2. IF $SQ > 0$ THEN $DQ = -SQ$ ELSE $DQ = 4$
```

Step 1(b) compares a nominal attribute a_{nom} of e_o to the head noun of the object of a relation of e_i. If they are similar, it is likely that any attributes of that object might help distinguish e_o from e_i. I then add those attributes to the attribute list of e_i. If SQ is non-zero at the end of the loop, the nominal attribute a_{nom} has bad discriminating power and I set $DQ = -SQ$. If $SQ = 0$ at the end of the loop, then a_{nom} has good discriminating power and I set $DQ = 4$.

I also need to extend the algorithm for calculating DQ for a relation [$\texttt{prep}_\texttt{j}$ e_j] of e_o. The extension is presented in algorithm 5.4.

In short, I first compare the head noun of e_j in a relation [$\texttt{prep}_\texttt{j}$ e_j] of e_o to a nominal attribute a_{nom} of e_i. If they are similar, it is likely that any attributes of e_j might help distinguish e_o from e_i. I then add those attributes to the attribute list of e_o. I demonstrate how this kind of abduction works with a example. Consider simplifying the following sentence:

> Also contributing to the firmness in copper, the analyst noted, was [a report by Chicago purchasing agents]$_{e_0}$, *which precedes [the full purchasing agents*

Algorithm 5.4 Extension to calculate *DQ* for relations to handle nominal attributes

```
1. IF any distractor $e_i$ has a nominal attribute $a_{nom}$ THEN
    (a) IF $a_{nom}$ is similar to the head of $e_j$ THEN
        i. Add all attributes of $e_o$ to the attribute list and calculate their $DQ$s
2. calculate $DQ$ for the relation as in section 5.3.6
```

> *report]$_{e1}$* ***that is due out today*** *and gives an indication of what the full report might hold.*

There are two clauses that can be dis-embedded, shown above in italics and bold font respectively. To dis-embed the italicised *which* clause, I need to generate a referring expression for e_o when the distractor is e_1:

$$e_o = \begin{bmatrix} \texttt{head} & \textit{report} \\ \texttt{by} & \begin{bmatrix} \texttt{head} & \textit{agents} \\ \texttt{attrib} & [\textit{Chicago}, \\ & \textit{purchasing}] \end{bmatrix} \end{bmatrix}$$

$$e_1 = \begin{bmatrix} \texttt{head} & \textit{report} \\ \texttt{attributes} & [\textit{full, purchasing, agents}] \end{bmatrix}$$

The distractor *the full purchasing agents report* contains the nominal attribute *agents*. To compare *report by Chicago purchasing agents* with *full purchasing agents report*, my algorithm flattens the former to *Chicago purchasing agents report*. My algorithm now gives:

$DQ_{agents} = -4$
$DQ_{purchasing} = -4$
$DQ_{Chicago} = 4$
$DQ_{by\ Chicago\ purchasing\ agents} = 4/4 = 1$

and I end up with the referring expression *the Chicago report*. The simplified text is now:

> Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. **This Chicago report** precedes the full purchasing agents report *that is due out today* and gives an indication of what the full report might hold.

For the *that* clause, I need to find a referring expression for $e1$ (*full purchasing agents report*) when the distractor is e_o(*report by Chicago purchasing agents*). My algorithm again flattens e_o and gives:

$DQ_{agents} = -4$
$DQ_{purchasing} = -4$
$DQ_{full} = 4$

The simplified text is now:

> Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. **This Chicago report** precedes a full purchasing agents report and gives an indication of what the full report might hold. **This full report** is due out today.

5.3.9 Evaluation

Evaluating referring-expression generation algorithms is notoriously difficult. The problem is partly due to the difficulty in disentangling the role of the referring-expression generator from the rest of a generation system. For example, the content-selection modules and the discourse model used in a generation system could affect the output of the referring-expression generator. The larger problem is that, as existing algorithms are highly domain-specific, it is impossible to construct an evaluation corpus that is acceptable to everyone.

As there doesn't exist any consensus on what kind of evaluation is suitable for this task, I decided to use a harsh corpus-based evaluation. I should note that this corpus-based evaluation is possible only because my algorithm can generate referring expressions in open domains, like Wall Street Journal text.

For my evaluation, I identified instances of referring expressions in the Penn Wall Street Journal Treebank. I then identified the antecedent and all the distractors in a four sentence window. I used my program to generate a referring expression for the antecedent, giving it a contrast-set containing the distractors. My evaluation consisted of comparing the referring expression generated by my program with the one that was used in the WSJ.

I looked at 146 instances of definite descriptions (noun phrases with a definite determiner) in the WSJ that satisfied the three conditions below:

1. An antecedent was found for the referring expression.
2. There was at least one distractor in the discourse window.
3. The referring expression contained at least one attribute or relation.

In 81.5% of the cases, my program returned a referring expression that was identical to the one used in the WSJ. This is a surprisingly high accuracy, considering that there is a fair amount of subjectivity in the way human writers generate referring expressions. In fact, in many of the remaining 18.5% cases, my algorithm returned results that were acceptable, though different. For instance, the WSJ contained the referring expression *the p53 gene*, where the antecedent (as found by my algorithm) was *the p53 suppressor gene* and the contrast set (as found by my algorithm) was:

> {an obscure gene}

My program generated *suppressor gene*, where the WSJ writer preferred *p53 gene*.

It was in many cases difficult to decide whether what my program generated was acceptable or wrong. For example, the WSJ contained the referring expression *the one-day limit*, where the antecedent (as found by my algorithm) was *the maximum one-day limit for the S&P 500 stock-index futures contract* and the contrast set (as found by my algorithm) was:

> {the five-point opening limit for the contract, the 12-point limit, the 30-point limit, the intermediate limit of 20 points}

My program generated *the maximum limit* (where the WSJ writer preferred *the one-day limit*), and it is not obvious to me whether that is acceptable.

5.4 Preserving Anaphoric Structure

There are many linguistic devices available for referencing a previously evoked entity. The shortest such device is usually the use of a pronoun. Pronouns are more ambiguous than other forms of referencing (like the use of definite descriptions), and their correct resolution depends on the reader maintaining a correct focus of attention. As I cannot ensure that the attentional state (the model of the reader's focus of attention, refer to section 1.6.1) at every point in the discourse remains the same before and after simplification, I have to consider the possibility of broken pronominal links. In this section, I discuss the idea of an anaphoric post-processor for syntactically transformed text. The basic idea is that the rearrangement of textual units that results from syntactic simplification (or any other application with a rewriting component) can make the original pronominalisation unacceptable. It is therefore necessary to impose a new pronominal structure that is based on the discourse structure of the regenerated text, rather than that of the original. In particular, it is necessary to detect and fix pronominal links that have been broken by the rewriting operations.

5.4.1 *Pronominalisation, Cohesion and Coherence*

As stated in my objectives (section 1.1), my interest in pronominalisation stems from my desire to ensure that the simplified text retains anaphoric cohesion. This objective is different from that of Canning et al. (2000a) in the PSET project (section 1.4.2), whose objective was to replace any pronoun with its antecedent noun phrase. This was intended to help aphasics who, due to working memory limitations, might have difficulty in resolving pronouns. In this section, I only aim to fix broken pronominal links and do not approach pronoun-replacement as a form of text-simplification in itself.

Syntactic transformations can change the grammatical function of noun phrases and alter the order in which they are introduced into the discourse. This can result in an altered attentional state at various points in the discourse. If the text contains pronouns at these points, it is likely that pronominal use may no longer be acceptable under the altered attentional state. My theory of how detect and fix broken pronominal links is quite straightforward. A model of attentional state needs to be simultaneously maintained for both the original and the simplified text. At each pronoun in the simplified text, the attentional states are compared in both texts. If the attentional state has been altered by the simplification process, my theory deems pronominal cohesion to have been disrupted. Cohesion can then be restored by replacing the pronoun with a referring expression for its antecedent noun phrase.

I use a salience function to model attentional state. For the rest of this chapter, I use the term *salience list* (S) to refer to a list of discourse entities that have been sorted according to the salience function described in section 3.1.6. As an illustration, consider example 5.17 below:

(5.17) a. Mr Blunkett has said he is "deeply concerned" by the security breach which allowed a comedian to gatecrash Prince William's 21st birthday party at Windsor Castle.

b. **He** is to make a statement to the Commons on Tuesday after considering a six-page report on the incident by police.

After the transformation stage (including transform-specific regeneration tasks), the simplified text is:

(5.17) a'. Mr Blunkett has said he is "deeply concerned" by a security breach.

a". This breach allowed a comedian to gatecrash Prince William's 21st birthday party at Windsor Castle.

b'. **He** is to make a statement to the Commons on Tuesday after considering a six-page report on the incident by police.

At the highlighted pronoun *he*, the salience lists for the original and simplified texts are:

$$\begin{aligned} S_{orig} &= \{\text{Mr Blunkett, the security breach, a comedian, Prince William's 21st} \\ &\qquad \text{birthday party, Prince William, Windsor Castle, ... }\} \\ S_{simp} &= \{\text{this breach, a comedian, Prince William's 21st birthday party,} \\ &\qquad \text{Prince William, Windsor Castle, Mr Blunkett, ... }\} \end{aligned}$$

The altered attentional state suggests that the use of the pronoun *he* is no longer appropriate in the simplified text. The pronoun is therefore replaced with the noun phrase *Mr Blunkett*.

To replace a pronoun, its antecedent needs to be located using a pronoun resolution algorithm. As these algorithms have an accuracy of only 65-80%, pronoun-replacement can introduce new errors in the simplified text. I therefore want to replace as few pronouns as possible. I do this by relaxing my original objective of preserving pronominal cohesion to only preserving pronominal coherence. My procedure now is to run my pronoun-resolution algorithm on the simplified text. I deem pronominal coherence to be lost if my pronoun-resolution algorithm returns different antecedents for a pronoun in the original and simplified texts. For the highlighted *he* in example 5.17, my pronoun-resolution algorithm returns *Mr Blunkett* for the original text and *a comedian* for the simplified text. The pronoun is therefore replaced by *Mr Blunkett*. For this example, both procedures return the same result. However, consider example 5.18 below:

(5.18) a. Mr Barschak had climbed a wall to reach the terrace.

b. He then appears to have approached a member of staff of the contractors, who then took **him** quite properly to a police point.

After the transformation stage (including transform-specific regeneration tasks), the simplified text is:

(5.18) a'. Mr Barschak had climbed a wall to reach the terrace.

b'. He then appears to have approached a member of staff of the contractors.

b". This member then took **him** quite properly to a police point.

At the highlighted pronoun *him*, the salience lists for the original and simplified texts are:

S_{orig} = {Mr Barschak (he), a member, staff, contractors, wall, terrace, ... }
S_{simp} = {This member, Mr Barschak (he), a member, staff, contractors, wall, terrace, ... }

For this example, despite the change in attentional state, my pronoun resolution algorithm returns *Mr Barschak* as the antecedent of *him* in both texts (as the binding constraints described in section 3.1.5 rule out *this member* as a potential antecedent in the simplified text). The pronoun is therefore not replaced, as coherence is deemed to have been preserved, even if cohesion is disrupted.

In fact, I can relax my objective further, to only preserve *local* pronominal coherence. As described in section 3.1.9, my pronoun-resolution algorithm is significantly more accurate when finding the immediate antecedent than when finding the absolute antecedent. I therefore do not replace a pronoun if the immediate antecedent is the same in both texts. In example 5.18 above, the immediate antecedent of *him* is *he* in both texts. I assume that this is sufficient to preserve local coherence. My algorithm for detecting and fixing broken anaphoric links is:

Algorithm 5.5 Detecting and fixing pronominal links

Anaphoric-Postprocessor

```
1. FOR every pronoun $P$ in the simplified text DO
   (a) Find the antecedents of $P$ in the simplified text.
   (b) IF neither the immediate nor absolute antecedents are the same as in the
       original text THEN replace $P$ in the simplified text with a referring expression
       for the antecedent in the original text
```

My theory only aims to fix broken anaphoric links in a text and does not attempt to replace the existing anaphoric structure with a new one. In particular, algorithm 5.5 can only replace pronouns in a text and cannot, in any situation, introduce pronouns. Consider:

(5.19) a. Incredulity is an increasingly lost art.

b. It requires a certain self-confidence to go on holding the line that Elvis Presley isn't in an underground recording studio somewhere.

c. David Beckham is prone to provoking revisionist hints because the virtues he represents are rare not only in the general population but especially so in football.

The sentence 5.19(c) is transformed to 5.19(c') below:

(5.19) c'. The virtues **he** represents are rare not only in the general population but especially so in football. So, David Beckham is prone to provoking revisionist hints.

My pronoun-resolution algorithm resolves *he* to *David Beckham* in the original text, but incorrectly to *Elvis Presley* in the simplified text. My anaphoric post-processor therefore replaces *he* with *David Beckham* to give:

(5.19) c". The virtues **David Beckham** represents are rare not only in the general population but especially so in football. So, David Beckham is prone to provoking revisionist hints.

However, as the focus of the discourse is *David Beckham* at the start of the second sentence in 5.19(c"), it might be desirable to pronominalise the subject, to give:

(5.19) c"'. The virtues David Beckham represents are rare not only in the general population but especially so in football. So, **he** is prone to provoking revisionist hints.

I do not attempt this kind of anaphoric restructuring. This is because people who might benefit from text simplification might also have difficulty resolving pronouns and might therefore prefer (c") to (c"').

5.4.2 Attentional States and the Reader

As I have mentioned before, the correct resolution of pronouns by readers depends on their maintaining an accurate focus of attention. In my approach to fixing broken pronominal links, I have tried to ensure that if readers could correctly resolve pronouns in the original text, they would also be able to do so in the simplified text. I have done this by using a pronoun-resolution algorithm as a model of the reader and assuming that if the algorithm resolved a pronoun incorrectly in the simplified text, the reader would also have difficulty in resolving it. This raises the interesting question of whether I can adapt my anaphoric post-processor to different readers, simply by changing my pronoun-resolution algorithm.

In algorithm 5.5, I used the same pronoun resolution algorithm on both the original and the transformed texts. To tailor the text for particular readers who have trouble with resolving pronominal links, all I need to do is use a different pronoun resolution algorithm on the simplified text. I discuss two possibilities below. Note that I still need to use the best available pronoun resolution algorithm on the original text to locate the correct antecedent.

If I use my pronoun-resolution algorithm without the agreement and syntax filters, my approach reduces to one that aims to preserve cohesion. If the most salient entity when processing a pronoun is not the correct antecedent, the pronoun is replaced. This results in a model where pronouns can only be used to refer to the most salient entity and cannot be used to change the discourse focus.

Algorithm	No. Replaced	No. of Errors	Accuracy
Cohesion Preserving	68	19	.72
Coherence Preserving	17	5	.70
Local-Coherence Preserving	11	3	.73

Table 5.4. Results for pronoun replacement

If I do away with the pronoun-resolution algorithm completely, my approach reduces to one in which all pronouns being replaced. This is similar to the anaphoric simplification carried out by Canning et al. (2000a).

5.4.3 *Evaluation*

I now evaluate three different approaches to pronoun-replacement that I have described—cohesion preserving, coherence preserving and local-coherence preserving. These approaches are implemented using algorithm 5.5 with a pronoun resolution algorithm without any filters (for preserving cohesion), using filters and only comparing absolute antecedents (for preserving coherence) and using filters and comparing both immediate and absolute antecedents (for preserving local-coherence). Table 5.4 shows the results of these approaches on the corpus of Guardian news reports introduced in section 3.6. I do not attempt pronoun replacement for occurrences of the pronoun *it*. This is because 85% of *it*s in the Guardian news reports are not anaphoric (refer to section 3.1.10).

To summarise, there were 95 sentences that were simplified. These resulted in an altered attentional state at 68 pronouns. In most of these cases, agreement and binding constraints ensured that the pronoun was still correctly resolvable. There were only 17 pronouns for which my pronoun-resolution algorithm found different absolute antecedents in both texts. There were only 11 pronouns for which both the immediate and absolute antecedents differed between the texts. Hence, to preserve local coherence, only around one in ten simplifications required pronoun replacement. My approach resulted in the introduction of only three errors.

5.5 Discussion

In this chapter, I have motivated the need for a regeneration component in text simplification systems by showing how naive syntactic restructuring of text can significantly disturb its discourse structure. I have formalised the interactions between syntax and discourse during the text simplification process and shown that to preserve conjunctive cohesion and anaphoric coherence, it is necessary to model both intentional structure and attentional state. I have also described an algorithm for generating referring expressions that can be used in any domain. My algorithm selects attributes and relations that are distinctive in context. It does not rely on the availability of an adjective classification scheme and uses WordNet antonym and synonym lists instead. It is also, as far as I know, the first algorithm that allows for the incremental incorporations of relations.

My approach preserves conjunctive cohesion by using rhetorical structure theory and issues of connectedness to decide the regeneration issues of cue-word selection, sentence ordering and determiner choice. However this can lead to unavoidable conflict with my objective of preserving anaphoric coherence. Consider:

(5.20) a. Back then, scientists had no way of ferreting out specific genes, but under a microscope they could see the 23 pairs of chromosomes in the cells that contain the genes.

b. Occasionally, gross chromosome damage was visible.

c. Dr. Knudson found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent, who had necessarily had the disease.

At the end of sentence 5.20(c), the attentional state is:

$S = \{$Dr. Knudson, children, damaged copy, parent, eye cancer, ...$\}$

When I split the last sentence, I have the choice of ordering the simplified sentences as either of 5.20(c') or 5.20(c"):

(5.20) c'. A parent had necessarily had the disease. Dr. Knudson found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from this parent.

c". Dr. Knudson found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent. This parent had necessarily had the disease.

When sentence 5.20(c) is replaced by 5.20(c'), the attentional state is:

$S = \{$Dr. Knudson, children, damaged copy, parent, eye cancer, ...$\}$

When sentence 5.20(c) is replaced by 5.20(c"), the attentional state is:

$S = \{$parent, disease, Dr. Knudson, children, damaged copy, ...$\}$

There is now a conflict between preserving the discourse structure in terms of attentional state and preserving the discourse structure in terms of conjunctive cohesion. The non-restrictive relative clause has an *elaboration* relationship with the referent noun phrase. To maintain this *elaboration* relationship after simplification, the dis-embedded clause needs to be the second sentence, as in 5.20(c"). This ordering also leads to a more connected text, as described in section 5.2.1. However, this ordering significantly disrupts the attentional state that is more or less preserved by the ordering 5.20(c'). This conflict between picking the ordering that preserves attentional state and the ordering that preserves conjunctive cohesion is unavoidable as the simplification process places a noun phrase that was originally in a non-subject position in a subject position, hence boosting its salience.

My theory allows me to handle issues of conjunctive and anaphoric cohesion separately. It allows me to select the ordering that preserves conjunctive cohesion (5.20(c")) and postpone consideration of any issues of anaphoric cohesion that result from the altered attentional state.

In this example, the sentence that follows the simplified sentence 5.20(c) is:

(5.20) d. Under a microscope, **he** could actually see that a bit of chromosome 13 was missing.

The pronoun *he* refers to *Dr. Knudson* in the original text. However, under the altered attentional state in the simplified text, *he* can be misinterpreted to refer to *parent.* I have described how an anaphoric post-processor can be used to detect and fix such problems. For this example, it replaces *he* with *Dr. Knudson* to give:

(5.20) d'. Under a microscope, **Dr. Knudson** could actually see that a bit of chromosome 13 was missing.

The process of replacing pronouns with referring expressions provides the added benefit of restoring the attentional state in the rewritten text. For example, at the end of sentence 5.20(d) (sentence 5.20(d') in the simplified text), the attentional states are:

S_{orig} = {Dr. Knudson, microscope, bit, chromosome, children, ...}

S_{simp} = {Dr. Knudson, microscope, bit, chromosome, parent, ...}

My anaphoric post-processor is general enough to be reusable in applications other than simplification, such as summarisation and translation, as long as pronoun resolution algorithms for the languages involved exist and pronouns can be aligned in the original and rewritten texts.

6 *Evaluation*

In chapters 3 – 5, I described how text simplification could be achieved using shallow robust analysis, a small set of hand-crafted simplification rules and a detailed analysis of the discourse aspects of syntactic transforms. I presented evaluations for my approaches to various natural language processing problems (including clause and appositive identification and attachment, pronoun resolution and referring-expression generation) along the way. I now evaluate my text simplification system as a whole, discussing the correctness of the simplified text, the level of simplification achieved and the potential uses of the simplified text.

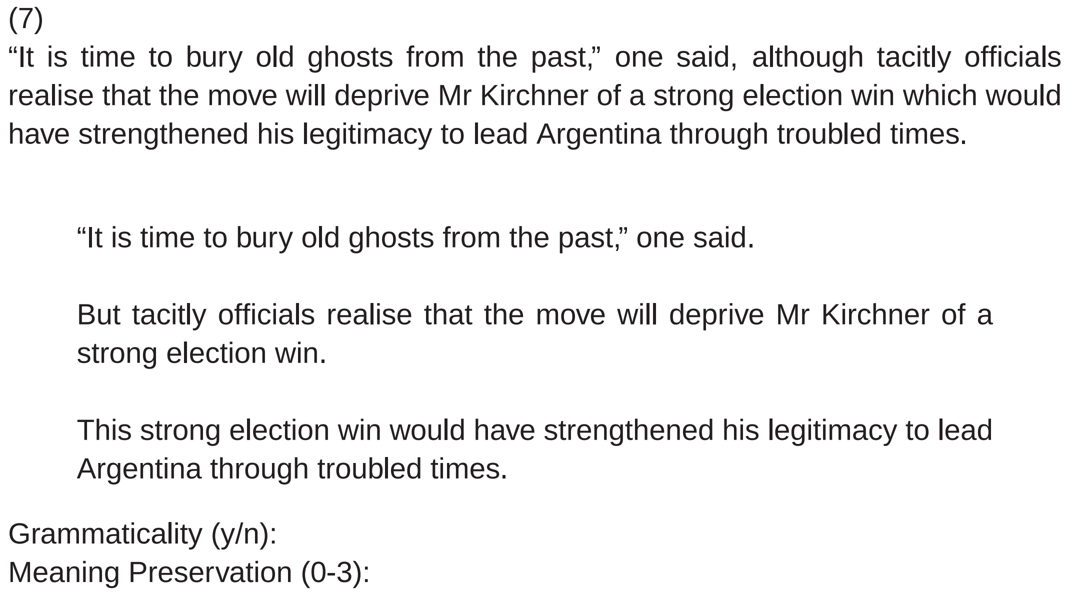
Evaluation criteria for NLP systems are broadly categorised as being either *intrinsic* or *extrinsic*. Sparck Jones and Galliers (1996) defines intrinsic criteria as "those relating to a system's objective" and extrinsic criteria as "those relating to its function".

My objectives in this thesis were to study the interaction between syntax and discourse during the simplification process and to demonstrate that text simplification was achievable in near-runtime using shallow and robust processing. My primary focus is therefore on evaluating the intrinsic aspects of text simplification; in particular, on evaluating the correctness of the simplified text (how well it preserves grammaticality, meaning and cohesion) and on measuring the level of simplification achieved. I evaluate these aspects in sections 6.1 – 6.3, which include a discussion on how to quantify readability in section 6.2.

In section 1.2 of the introduction, I described various potential uses of syntactic simplification. These included making newspaper text accessible to people with low reading ages or language disorders and assisting other NLP applications such as parsing or machine translation. While a detailed extrinsic evaluation is beyond the scope of this thesis, I present a few (preliminary) indicators of the usefulness of syntactic simplification in section 6.4.

6.1 Evaluating Correctness

There are three aspects to evaluating the correctness of text simplification— the grammaticality of the regenerated text, the preservation of meaning by the simplification process and the cohesiveness of regenerated text. In order to evaluate correctness, I conducted a human evaluation using three native-English speakers with a background in computational linguistics as subjects. I presented the three subjects with 95 examples. Each example consisted of a sentence from the Guardian news corpus described in section 3.6 that was simplified by my program, the corresponding simplified sentences that were generated and boxes for scoring grammaticality and semantic parity. An example from the



(7)
“It is time to bury old ghosts from the past,” one said, although tacitly officials realise that the move will deprive Mr Kirchner of a strong election win which would have strengthened his legitimacy to lead Argentina through troubled times.

“It is time to bury old ghosts from the past,” one said.

But tacitly officials realise that the move will deprive Mr Kirchner of a strong election win.

This strong election win would have strengthened his legitimacy to lead Argentina through troubled times.

Grammaticality (y/n):
Meaning Preservation (0-3):

Figure 6.1. An example from the data-set for the evaluation of correctness

evaluation is presented in figure 6.1, and the entire set, along with the subjects’ ratings is attached in appendix B.1.

The subjects were asked to answer *yes* or *no* to the grammaticality question. They were asked to score semantic parity between $0 - 3$ using the following guidelines (the guidelines are reproduced in full in appendix A.2):

0: The information content (predicative meaning) of the simplified sentences differs from that of the original.

1: The information content of the simplified sentences is the same as that of the original. However, the authors intensions for presenting that information has been drastically compromised, making the simplified text incoherent.

2: The information content of the simplified sentences is the same as that of the original. However, the author’s intensions for presenting that information have been subtly altered, making the simplified text slightly less coherent.

3: The simplified text preserves both meaning and coherence.

In short, they were asked to judge meaning preservation as either 0 (meaning altering) or non-0 (meaning preserving) and rate cohesion on a scale of $1 - 3$. The results of this evaluation are detailed below and summarised in table 6.1.

Judges	**Grammatical (G)**	**Meaning Preserving (MP)**	**G and MP**
Unanimous	80.0%	85.3%	67%
Majority vote	94.7%	94.7%	88.7%

Table 6.1. Percentage of examples that are judged to be grammatical and meaning-preserving

6.1.1 Grammaticality

Of the 95 examples, there were 76 where the simplified sentences were grammatical according to all three judges. There were a further 14 examples that were grammatical according to two judges and 2 that were grammatical according to one judge. Surprisingly, there were only 3 examples that were judged ungrammatical by all three judges.

Of the examples where there was disagreement between the judges, some involved cases where separating out subordination resulted in a possibly fragmented second sentence; for example (from #14, appendix B.1):

> But not before he had chased pursuing police officer onto the bonnet of their car.

Interestingly, many of the others involved cases where the ungrammaticality was present in the original sentence, usually in the form of bad punctuation. For example, the original sentence (#51, appendix B.1):

> An anaesthetist who murdered his girlfriend with a Kalashnikov souvenir of his days as an SAS trooper, was struck off the medical register yesterday, five years later.

resulted in one of the simplified sentences being deemed ungrammatical by one judge:

> An anaesthetist, was struck off the medical register yesterday, five years later.

The other two judges consistently marked sentences that inherited grammar errors from the original as grammatical.

6.1.2 Meaning

Out of the 95 cases, there were 81 where all three judges agreed that predicative meaning had been preserved (scores greater than 0). There were a further 9 cases where two judges considered the meaning to be preserved and 2 cases where one judge considered the meaning to be preserved. There were only three cases where all three judges considered the meaning to have been altered. Most of the cases where two or more judges deemed meaning to have been changed involved incorrect relative clause attachment; for example (#81, appendix B.1), the sentence:

> They paid cash for the vehicle, which was in “showroom” condition.

got simplified to:

> They paid cash for the vehicle. This cash was in “showroom” condition.

Interestingly, all three judges were comfortable judging meaning to be preserved even for examples that they had deemed ungrammatical. This suggests that marginal ungrammaticalities (like the examples under *grammaticality* above) might be acceptable from the comprehension point of view. The serious errors tended to be those that were judged to not preserve meaning (many of which were also judged ungrammatical); for example, the simplified sentences in #60, appendix B.1:

> In recent weeks the judiciary and security services have targeted some independent journalists were shut down, subjecting them to detention without trial and interrogation. These independent journalists turned to the internet after their newspapers.

These invariably arose from errors in the analysis module, in either clause identification or clause attachment.

As table 6.1 shows, around two-thirds of the examples were unanimously deemed to be grammatical and meaning-preserving while almost 90% of the examples were judged to preserve grammaticality and meaning by at least two out of three judges.

6.1.3 Cohesion

The judges were also asked to judge coherence (0 or 1 indicating major disruptions in coherence, 2 indicating a minor reduction in coherence and 3 indicating no loss of coherence). There were 39 examples (41%) for which all the judges scored 3. However, there was very little agreement between judges on this task. The judge were unanimous for only 45 examples. To get an indication of how well my system preserves coherence despite the lack of agreement between judges, I considered the average score for each example. There were 71 examples (75%) where the judges averaged above 2. An average score of above two can be assumed to indicate little or no loss of coherence. There were 16 examples (17%) where the judges averaged more than 1 and less than or equal to 2. These scores indicate that the judges were sure that there was a loss of cohesion, but were unsure about whether it was minor or major. There were 8 examples (8%) for which the judges averaged less than or equal to 1. These scores indicate incoherence and a possible change in meaning. The average of the scores of all the judges over all the examples was 2.43, while the averages of the individual judges were 2.55, 2.57 and 2.13. Figure 6.2 plots the scores of each judge for each example.

6.1.4 Interpreting these Results

In the last section, I reported that the judges averaged 2.43 over all the examples. I now try to interpret that result by discussing two issues. The first issue relates to whether there might be limitations in my experimental methodology that might have biased my results. The second issue relates to establishing upper and lower bounds for the cohesion of simplified text, between which my figure of 2.43 can be positioned.

As shown in figure 6.1, the judges were provided with the original sentence and the simplified sentences. The simplified sentences were presented on separate lines, rather than in one paragraph. This was done in order to aid the judgements on grammaticality, by using the typography to emphasise that each simplified sentence needed to be tested individually. The negative aspect of this typographic decision is that the judges might have employed lower benchmarks for cohesion than if the simplified sentences had been presented in paragraph form. This might have resulted in inflated cohesion scores. However, it is hoped that the judges would have compared the cohesion of the simplified sentences with the *upper bound* cohesion of the original sentence when assigning their scores, and not have been overly affected by the typography.

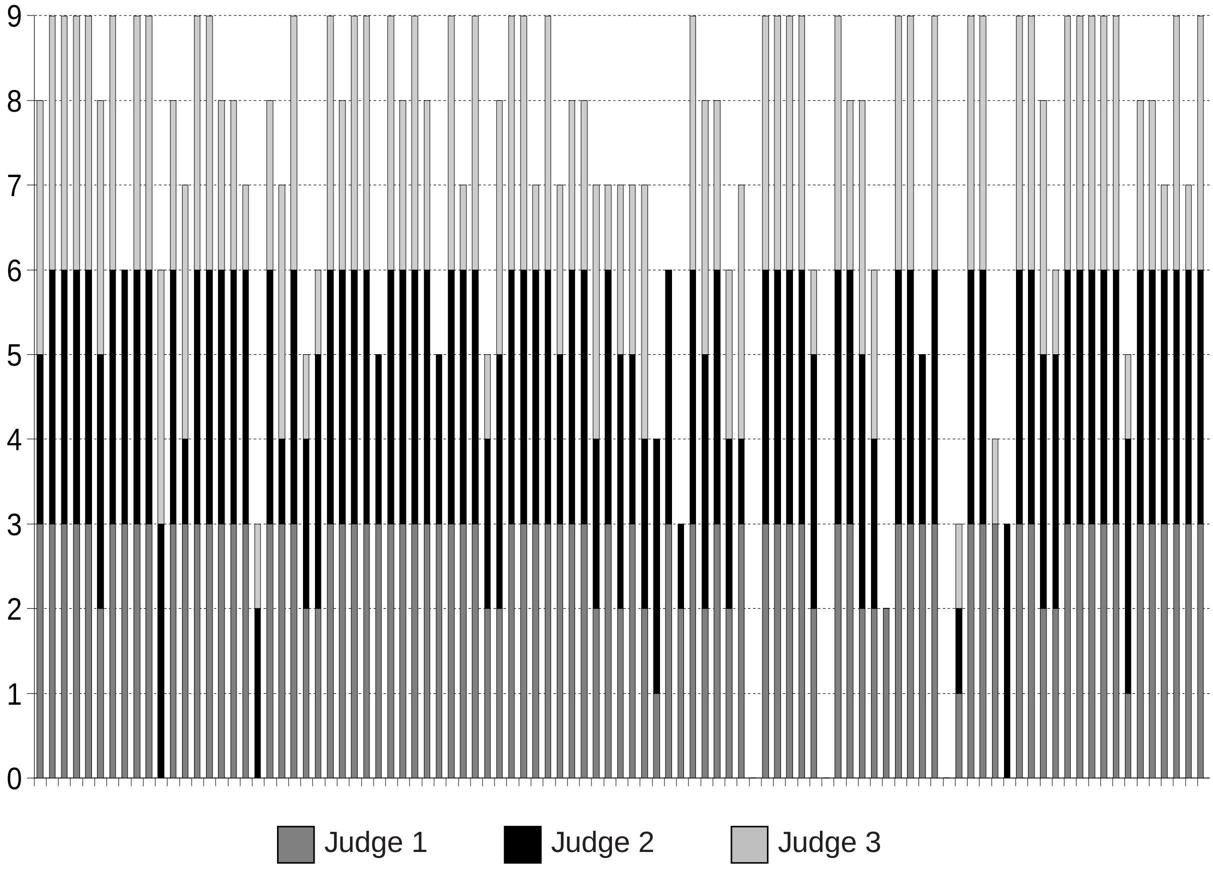


Figure 6.2. The coherence scores of the three judges for each example

I now consider the question of what an average cohesion score of 2.43 might mean. Using the guidelines provided to the judges, this figure can be interpreted to mean that on average, the loss of cohesion in the simplified text is minor. It would however be useful to compare this number with a suitable baseline and ceiling for cohesion in simplified text. Ideally, I would have presented the judges with three simplified versions; in addition to my program's output, they would also have been given a human simplified version and a version produced by a baseline program. If the judges were then asked to judge all three without being told which is which, their cohesion score for the human simplified text could have been used as an upper bound and their cohesion score for the baseline algorithm as a lower bound. This more comprehensive evaluation was not performed due to time constraints (A human simplifier would have had to be trained in the task because syntactic simplification is not as intuitive a task as, for example, summarisation). I can therefore only discuss what these bounds might have been.

The obvious upper bound is 3.00, which represents no loss in cohesion. However, this is unrealistically high. Relative clauses, appositives and conjunctions are all cohesive devices in language. It is quite plausible that these constructs cannot be removed from a text without some loss of cohesion. For example, the judge who gave #59 (appendix B.1) a low coherence score stated that he could not rewrite the simplified sentences in a manner that preserved the subtleties of the original. This suggests that any simplification would result in a loss of cohesion. Further, as reported in section 5.2.4, when the judges did offer revised versions of the simplified sentences, they were often quite dissimilar, and the revisions were often of a semantic nature. It is therefore quite hard to come up with a sensible upper bound for cohesion for a text simplification system that only addresses issues of syntax and discourse, and does not consider semantics; and while I can speculate that the upper bound might be less than 3.00, I cannot quantify what that bound might be.

To compensate for the lack of a lower bound, I tried to assess the utility of only my sentence ordering algorithm, by extrapolating from the results of the original evaluation. There were 17 examples (18%) where my sentence ordering algorithm returned a different order from that of a baseline algorithm which preserved the original clause order. This is a high enough percentage to justify the effort in designing the sentence ordering module. Also, my data set did not contain any instance of a *because* clause, which is the only instance of conjunction where my algorithm reverses clause order. On the 17 examples where my algorithm changed the original clause order, the average of the three judges scores was 2.53, which is higher than the average for all 95 examples.

6.2 Readability

In the previous section, I evaluated how well my system preserved grammaticality, meaning and coherence. I also need to quantify the amount of simplification achieved by my system. To do this, I need an objective measure of readability that I can use on both the original and the simplified texts. I discuss ways of measuring readability in this section and present my evaluation in the next.

The issue of assessing the difficulty of texts has received considerable attention in the last fifty years. Educationalists have been seeking time and labour saving means for deciding whether particular books are suitable for particular readers. It is widely acknowledged that there are three aspects to matching a text to a reader—comprehension (will the reader understand it?), fluency (can the reader read it at optimal speed) and interest (will the reader be sufficiently interested in reading it?). The term *readability* has come to denote the combination of all three aspects. These three aspects are to some extent interdependent, and sufficient interest can often result in good comprehension on difficult texts, with possibly reduced fluency. Gilliland (1972) discussed various methods that have been used to measure readability. I summarise his findings in section 6.2.1 below.

6.2.1 *Measuring Readability*

Gilliland (1972) compared three different approaches to measuring readability. The first involves the subjective judgements of teachers. A teacher skims through a text and ranks its readability. This method captures the expert knowledge that the teacher can be expected to have about students. It however suffers from the inconsistencies inherent in human judgements. Gilliland (1972) reported that using a panel of teachers resulted in more consistent readability rankings. However, using a panel is usually infeasible, and subjective judgements by individual experts (librarians or teachers) are still widely used to classify books in libraries and schools.

The second approach involves using question-answering techniques on a representative sample of end readers, in an attempt to directly measure comprehension. This method has proved unpopular due to many methodological limitations. Results have been shown to vary dramatically with the kinds of questions asked and the order in which they are asked, with whether the text is removed before the question-answering session or not, and even on who decides whether an answer is "sufficiently" correct. Objectivity can be improved by using a multiple choice questionnaire, but designing choices requires detailed knowledge of test construction, and is impractical for most situations. Badly designed questions can be more difficult than the text, or fail to test the reader's understanding at all.

The third approach, that has been widely adopted, is that of using *readability formulae* to decide the reading age that a text is suitable for. There are hundreds of formulae that have been proposed, of which around seven have found widespread acceptance. Most of the widely used formulae have been found to have correlations of over 0.7 with judgements by teachers (Lunzer and Gardner, 1979). Unlike the other two approaches described above, readability formulae provide an objective and easy to calculate means of quantifying readability. However, they only predict the *comprehension* and *fluency* aspects of readability and cannot predict *interest*. As there is very little difference in the predictions of the popular readability formulae, I consider only the most widely used among them, the Flesch readability formula (Flesch, 1951). I now describe the Flesch formula and why it is effective at predicting the readability of *normal* texts. I also discuss ways in which it has been abused, and its applicability for predicting the readability of rewritten text, such as the simplified text generated by my system.

6.2.2 The Flesch Formula

The reason that the Flesch formula has gained widespread acceptance is that it is valid (has been shown to correlate well with teacher judgements), reliable (is objective enough that different people using it on the same text come up with the same score) and easy to use (in terms of time and effort)[19]. The Flesch formula is:

$$\begin{aligned}\text{Reading ease score} = {} & 206.835\\ & -0.846\times\ \text{syllables\_per\_hundred\_words}\\ & -1.015\times\ \text{words\_per\_sentence}\end{aligned}$$

This formula is designed to score texts between 0 (hardest) and 100 (easiest), though exceptional texts like legal documents can result in negative scores. This reading ease score can then be converted into a grade level or reading age by the following mappings:

[19] The issue of ease of use was important in the era before computers became widely available. It is no longer relevant, but the Flesch formula remains firmly entrenched, to the extent that it is used as the readability metric in Microsoft Word©.

Reading ease score (RES)	Flesch grade level (FGL)	Reading age
Over 70	$-(RES - 150)/10$	$FGL + 5$
60 – 70	$-(RES - 110)/5$	$FGL + 5$
50 – 60	$-(RES - 93)/3.33$	$FGL + 5$
Below 50	$-(RES - 140)/6.66$	$FGL + 5$

Table 6.2 shows the Flesch scores for various genre. The minimum score for *plain English*[20] is 60 (corresponding to grade 10), which corresponds to roughly 20 words per sentence and 1.5 syllables per word.

The Flesch formula, like many other readability formulae, is based on very shallow features like sentence length (measured in words) and word length (measured in syllables). Its validity therefore relies on the assumption that sentence and word lengths have strong correlations with syntactic and lexical complexity. The formula makes no attempt at judging grammaticality or cohesion and is intended for use only on edited texts. Further, while it measures the *comprehension* and *fluency* aspects of readability, it does not measure the *interest* aspect. Due to these issues, care needs to be taken to ensure that the use of a formula for a particular purpose is appropriate. I discuss various ways in which these formulae have been abused in the next section.

Genre	Reading ease	Grade level	Reading age
Movie Screen	75	7.5	12.5
Reader's Digest	65	9	14
Time	52	12.4	17.4
Wall Street Journal	43	14.5	19.5
Harvard Law Review	32	16.2	21.2
Standard Auto Insurance Policy	10	19.5	24.5

Table 6.2. Flesch readability scores for some genre (taken from Flesch (1979))

6.2.3 The Abuse of Readability Formulae

The Flesch formula has been shown to be accurate (to within one grade level) at predicting reading levels when used judiciously[21]. However, it can be easily abused. Indeed the widespread misuse of readability measures led to a period of intense criticism, and even rejection, before a better appreciation of when and how they should be used led

[20] *Plain English* is a campaign to get companies and governments to write documents with the reader in mind, in a manner that gets information across clearly and concisely. The website http://www.plainenglish.co.uk/ defines plain English as "*language that the intended audience can understand and act upon from a single reading*".

[21] At least for grade levels 6 and above, it is less reliable at predicting reading levels for very elementary texts aimed at children below 10 years of age. For elementary school texts, the Dale-Chall Formula is more reliable. This formula counts the number of words in a text that do not feature in a stop-list of 3000 easy words. This is a more reliable indicator of lexical complexity than word length in elementary texts.

to a more guarded acceptance.

A common example of the misuse of readability formulae involved cases where teachers denied children access to books that were rated too difficult. This was unfortunate because children can often cope with books intended for higher reading ages, provided the content matter is of sufficient interest to them. Another example of misuse involved using the formulae on genre where linguistic complexity is known to be less correlated with comprehensibility; for example, philosophy and poetry.

However, the most blatant misuse of these formulae involved their use by authors in writing texts for particular reading ages. These formulae were designed to be used post-hoc, and can lose validity when authors revise their writing to achieve high readability scores. The 1950s–1970s saw a period where readability scores were made available to British educationalists when writing textbooks. The initial results were promising, with authors getting feedback on when particular portions needed rewriting. However, by the 1970s, a situation had been reached where the use of readability formulae at the authoring stage had resulted in dozens of unreadable textbooks. The problems arose because authors were subconsciously manipulating sentence and word lengths without decreasing the syntactic or lexical complexity; for example, by excessive use of pronouns and ellipses or by removing connecting phrases and cue-words, all of which can result in shorter sentences, while actually making a text harder to read. This caused a rethink on making these formulae available at the authorship stage. Flesch (1979) however argued that his formula was still useful at the authorship stage, provided certain guidelines were followed during revision. In short, though the formula only measures sentence length, in revising a text, the author needs to focus on reducing syntactic complexity, not sentence length. To quote from his book:

> "First, if you want to rewrite a passage to get a higher score, you'll have to cut the average sentence length. This means you'll have to break up long, complex sentences and change them to two, three or four shorter ones. In other words, sprinkle periods over your piece of writing. When you're turning subordinate clauses into independent sentences, you'll find that a lot of them will start with And, But or Or. Don't let that bother you. It's perfectly good English and has been good usage for many centuries. The Old Testament says, 'And God said, Let there be light; and there was light.' The New Testament says, 'But Jesus gave him no answer.' And Mark Twain wrote, 'Man is the only animal that blushes. Or needs to.' "
>
> Rudolf Flesch
> [From Chapter 2 of *How to write Plain English*]

Similarly, the author needs to focus on using simpler words, not shorter ones. Flesch (1979) claims that if these guidelines are followed, the use of the formula to judge the readability of the revised text remains valid.

In this thesis, I am interested in using the Flesch formula for judging the readability of the output of my syntactic simplification system. In the next section, I discuss the appropriateness of using the formula for that purpose and present my results on the readability of simplified text.

6.3 Evaluating the Level of Simplification achieved

I now evaluate the readability of the simplified text generated by my system using the Flesch formula described above. I present the results in section 6.3.2. But first, I discuss the appropriateness of using the Flesch formula on simplified text.

6.3.1 *Using the Flesch Formula on Simplified Text*

In some sense, my syntactic simplification system is an automatic text revision tool that aims to make text suitable for a lower reading age. As my system results in text containing shorter sentences than the original, the simplified text can be expected to achieve higher readability scores than the original. The question then arises— is it appropriate to use readability metrics on the simplified text?

My system does not increase pronominalisation or the use of ellipses and does not remove connecting phrases and cue-words. It reduces sentence length by breaking up complex and compound sentences into shorter sentences. It reduces syntactic complexity by following the guidelines set out by Flesch (1979). Further, the evaluation in section 6.1 suggests that though there is a slight loss in cohesion, by and large the simplified text is grammatical, coherent and meaning preserving. I feel that the use of readability formulae on my simplified text is therefore reasonable. I now discuss the readability of simplified text as predicted by the Flesch formula.

6.3.2 *The Readability of Simplified Text*

The Flesch reading ease score for my corpus of 15 Guardian news reports is 42.0 (suitable for a reading age of 19.7). After syntactic simplification by my program, the score increases to 50.1 (suitable for reading age 17.8). The increase in readability therefore appears to be only marginal. In particular, it stays significantly lower than 60, the alleged threshold for plain English. This is because the text has been simplified syntactically while retaining the vocabulary intended at a higher reading age. This pulls down the Flesch score, even though the reduction in sentence length is dramatic. While the average sentence length in the original text is 25.8 words, my syntactic simplification algorithm reduces it to 15.4 words. This is quite a significant decrease, and worth having a closer look at. Figure 6.3 shows the distribution of sentence lengths in the original and simplified texts. In the original text, over half the sentences are over 20 words long and around a third are longer than 25 words. In the simplified text, less than a quarter are over 20 words long, and only one in eight is longer than 25 words. Further, more than half the sentences are shorter than 15 words. However, despite this drastic reduction in syntactic complexity, the lexical complexity in Guardian news reports ensures that the simplified text is only suitable for a reading age of 18. It therefore appears that to make a mainstream newspaper accessible to a wider audience, it is important to also perform lexical simplification, in order to remove the mismatch between grammatical complexity and vocabulary.

In my next experiment, I ran my syntactic simplification program on news reports from other news sources (I used 15 reports per source). The results are summarised in table 6.3. My program appears to reduce average sentence lengths to around 15 words across

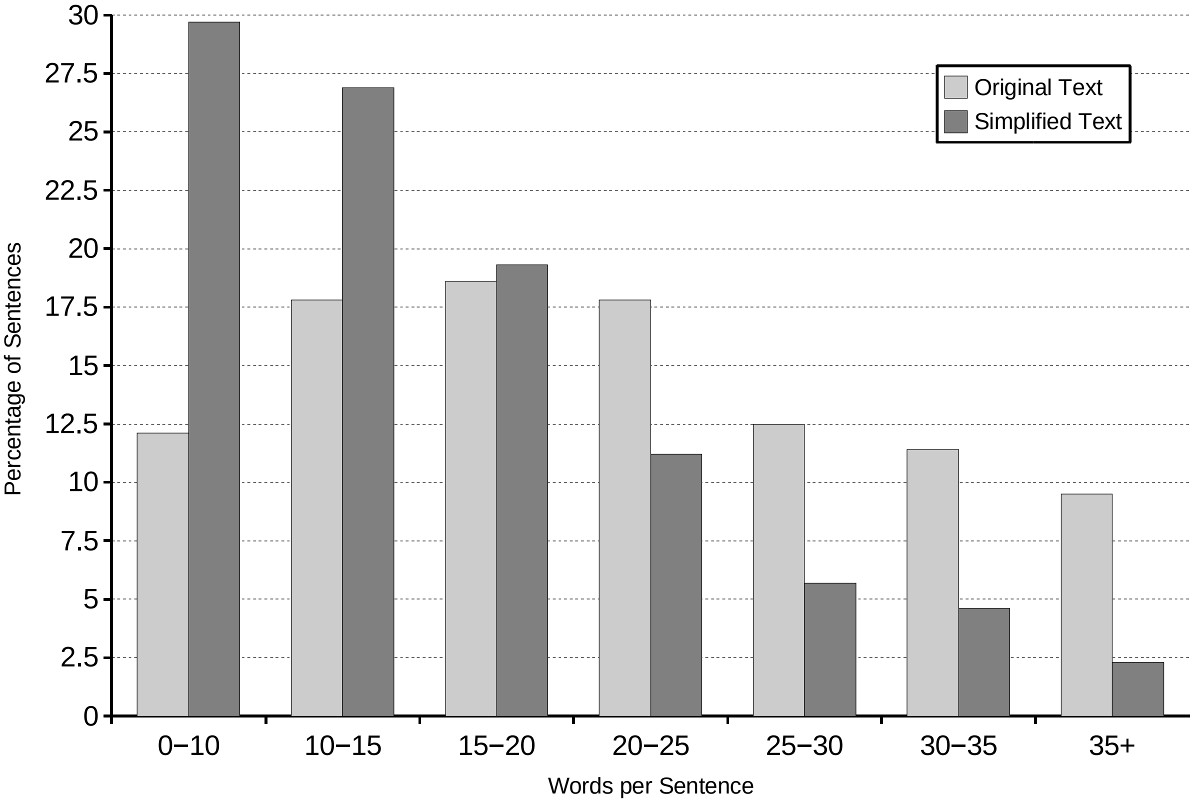


Figure 6.3. The distribution of sentence lengths in Guardian news reports before and after simplification

newspapers. However, there are big differences in the Flesch readability scores for the simplified news reports. Tabloids, regional newspapers and the BBC news online appear to use simpler vocabularies, and syntactic simplification alone is sufficient to raise their Flesch reading ease to over 60 (suitable for a reading age of 15).

6.3.3 *The Increase in Overall Text Length*

As described in the section above, syntactic simplification results in a decrease in average sentence length. However, it also results in an increase in overall text length. This is because it is often necessary to introduce new noun phrases and auxiliary verbs to make the simplified sentences grammatical. The increase in overall text length was 6% for my corpus of Guardian news reports. This equates to an average increase of 25 words per report. It is expected that this small increase in text length will not make the text too much harder to read as the increase is only one sentence's worth of words, and the average report is already 17 sentences long.

6.4 Evaluating Extrinsic Aspects

In section 1.2 of the introduction, I described various potential uses of syntactic simplification. These included making newspaper text accessible to people with low reading ages

Source of News Reports	Reading ease	Reading age	Av. Sentence Length
Wall Street Journal	40.1 ⟶ 44.2	20.0 ⟶ 19.3	20.8 ⟶ 16.7
Guardian	42.0 ⟶ 50.1	19.7 ⟶ 17.8	25.8 ⟶ 15.4
New York Times	43.8 ⟶ 52.4	19.4 ⟶ 17.2	19.2 ⟶ 14.4
Cambridge Evening News	51.3 ⟶ 60.8	17.5 ⟶ 14.8	21.7 ⟶ 14.6
Daily Mirror	54.7 ⟶ 63.2	16.5 ⟶ 14.3	18.9 ⟶ 14.7
BBC News	54.9 ⟶ 62.3	16.4 ⟶ 14.4	21.7 ⟶ 16.7

Table 6.3. Flesch readability scores and average sentence lengths before and after syntactic simplification (shown as *original* ⟶ *simplified*)

or language disorders and assisting other natural language applications such as parsing.

The results in the previous section suggest that news reports from regional newspapers, tabloids and online news sources can be simplified to a level suitable for people with a reading age of 14 – 15. It would have been interesting to verify that extrinsically, by performing comprehension tests on low reading-age subjects. Unfortunately, such experiments are difficult to conduct even for researchers who are qualified in experimental psychology. The process can easily take an year, from obtaining clearance from ethical committees, finding experimental subjects and fine-tuning the methodology through pilot experiments to performing the actual experiment. This means that performing a comprehensive end-user evaluation was always unfeasible given my time constraints. On the other hand, results obtained from any small-scale evaluation of comprehension on small numbers of low-reading-age subjects would be unreliable due to the problems associated with generalising from small data samples.

I therefore complete a full circle and turn my attention to evaluating syntactic simplification on another task, parsing, that provided the motivation for the first attempt at syntactic simplification (Chandrasekar et al., 1996). To see if syntactic simplification improves the throughput of a parser, I ran the RASP (Briscoe and Carroll, 1995) parser first on the 95 sentences from my Guardian corpus that were simplifiable, and then on the simplified sentences generated by my system. Table 6.4 shows the performance of the parser under different time-out settings[22]. It is evident from the table that the throughput of a parser can be doubled by applying syntactic simplification as a pre-process. However, the increase in throughput is only meaningful if there is no significant decrease in accuracy. It is therefore important to analyse how syntactic simplification alters the parser's output. I used the grammatical relations formalism (introduced in section 3.1.2) for comparing the parser's output on the original and simplified texts. There were some systematic differences in the GRs that RASP generated from the original and simplified texts. I enumerate these below:

1. **Relative Clauses:** Consider the sentence:

   > She called an ambulance *which took Mr Fitzgerald to Worcestershire Royal Hospital, Worcester*, but doctors decided he needed plastic surgery in Birmingham.

   RASP represents the relative clause attachment using the GRs:

[22] I used a 400MHz Pentium II Processor with 128MB of memory for all the experiments in this section.

Text	**Throughput & Failures**	**20 sec***	**5 sec***	**1 sec***
Original text	Total time taken(in seconds)	404	245	173
	Number of Parse failures	1	5	28
Simplified text	Total time taken(in seconds)	138	135	107
	Number of Parse failures	0	2	8

* Parser Setting for Timeout per sentence.

Time taken to simplify text: 41 seconds

Table 6.4. Throughput of the RASP parser on original and simplified sentences

```
(ncsubj take+ed which _)
(cmod which ambulance take+ed)
```

For the simplified text:

> She called an ambulance. An ambulance took Mr Fitzgerald to Worcestershire Royal Hospital, Worcester. But doctors decided he needed plastic surgery in Birmingham.

the corresponding GR is:

```
(ncsubj take+ed ambulance _)
```

RASP doesn't attach non-restrictive relative clauses, treating them as text adjuncts. In these cases, the `cmod` relation is absent. The GRs resulting from the simplified text might then be superior if it is necessary to attach relative clauses. However, this demonstrates the usefulness of a clause-attachment algorithm rather than a text simplification system.

2. **Apposition and Conjunction:** For apposition, the GR is `ncmod`. This gets changed to `xcomp` in the simplified text. In addition, an `ncsubj` relation is introduced. For conjunction, the `conj` (coordination) or `cmod` (subordination) relation is lost in the simplified text.

These systematic changes aren't particularly important, indeed they can easily be converted back to the original GRs from knowledge of the simplification rule used. More interestingly, RASP appears to analyse segments of text differently when simplified. For example, in:

> "It is time to bury old ghosts from the past," one said, although tacitly officials realise that **the move will deprive Mr Kirchner of a strong election win** which would have strengthened his legitimacy to lead Argentina through troubled times.,

the highlighted text in the middle is analysed as:

```
(ncsubj deprive move _)
(clausal deprive win)
(ncsubj win Kirchner _)
```

		Timeout: 20sec		Timeout: 1sec	
GR	**Metric**	**Original**	**Simplified***	**Original**	**Simplified***
ncsubj	precision(%)	84	77(84)	84	73(81[1])
	recall(%)	83	78(88)	78	76(87)
	F-measure	0.83	0.77(0.86)	0.81	0.74(0.83)
dobj	precision(%)	84	79	84	78
	recall(%)	79	78	76	77
	F-measure	0.81	0.78	0.80	0.77
iobj	precision(%)	22	22	19	22
	recall(%)	57	64	50	64
	F-measure	0.32	0.33	0.28	.33

* The numbers in brackets are obtained by correcting for GRs in the gold standard that have

relative pronouns as subjects. The correction involves matching only the verb in relations

with relative pronouns like `(ncsubj attend who _)`.

[1] The loss in precision arises due to the `ncsubj` relations introduced when simplifying apposition.

Table 6.5. GR-based evaluation of parser on original and simplified sentences

But the same text in the simplified sentence:

> But tacitly officials realise that **the move will deprive Mr Kirchner of a strong election win**.

gets analysed as:

```
(ncsubj deprive move _)
(dobj deprive Kirchner _)
(iobj of deprive win)
```

In this example, the GRs for the simplified text are correct. However, the fact that non-systematic changes in GRs can occur means that I need to evaluate the accuracy of the GRs generated from the parses of the original and simplified texts. In order to do this in an objective manner, I used the evaluation corpus for GRs (Carroll et al., 1999a; Briscoe et al., 2002)[23]. There were 113 sentences in the corpus that were simplifiable by my program. I ran the RASP parser on these sentences and their simplified forms. Table 6.5 compares the performance of the RASP parser on the three main GRs under two timeout settings.

The *difference in proportions test*[24] shows that there is no significant degradation in performance when the simplified text is used with a 1 second timeout, as compared to

[23] The evaluation corpus for GRs is available at http://www.cogs.susx.ac.uk/lab/nlp/carroll/greval.html.

[24] The *difference in proportions test* (see Snedecor and Cochran (1989) for description) is based on measuring the difference between the error (or success) rates of two algorithms. Suppose the error rates

using the original text with any timeout. The differences in the F-measures (ncsubj:.83 and .83, dobj: .81 and .77, iobj .32 and .33) on a data set of this size are not statistically significant at a confidence level of 85% or above.

The increase in parser throughput then appears to come unhindered by any significant decrease in accuracy, at least on the three GRs that I have evaluated parser performance on. This increased throughput could be useful when using parsers for tasks relating to information retrieval, such as information extraction or question answering, where there might be a reasonable tolerance for the changes in the parses that the simplification process introduces.

for the two algorithms are p_a and p_b. Then the number of errors made by algorithm a on n test examples is a binomial random variable with mean np_a and variance $p_a(1-p_a)/n$. This can be approximated as a normal distribution when the conditions $np_a > 5$ and $n(1-p_a) > 5$ hold (that is, for large enough data sets). As the difference of normal distributions is also a normal distribution, $p_a - p_b$ can be considered a normal distribution. Under the null hypothesis (that the two algorithms perform equally well), $p_a - p_b$ has a mean of 0 and standard error of $\sqrt{2p(1-p)/n}$, where p is the average error probability $(p_a+p_b)/2$. Then $z = (p_a - p_b)/\sqrt{2p(1-p)/n}$ has a standard normal distribution. The null hypothesis can be rejected if $|z| > 1.44$ (for a two-sided test with the probability of wrongly rejecting the null hypothesis of 0.15).

7 *Conclusions*

I now conclude by summarising the contributions of this thesis in section 7.1, discussing the scope for improvement in section 7.2 and finally, suggesting avenues for future work in section 7.3.

7.1 Summary of Results

In this thesis, I have presented a theory of text simplification that offers a treatment of the discourse-level aspects of syntactic rewriting. I have also proposed a modular architecture for a text simplification system and described a shallow implementation of each module. I now describe the contributions of this thesis by summarising the results obtained in the chapters on analysis, transformation and regeneration.

Analysis

I have demonstrated that syntactic simplification is feasible using shallow and robust analysis, and without using a parser.

I have compared different approaches to relative clause attachment and demonstrated that it is not a purely syntactic phenomenon. I have shown how attachment decisions can be made reliably using information about animacy and prepositional preferences in a machine learning framework. I have also shown how comparable results can be achieved by treating clause attachment as a relative-pronoun resolution problem, and provided a solution based on salience, agreement filters and syntactic filters. I have extended this approach to handle appositive attachment, treating the head noun in the appositive as an anaphor that needs to be resolved.

I have shown that shallow inference procedures used with a shallow discourse model can give good results on third-person pronoun resolution, even without using a parser. I have shown that it is worthwhile to try and acquire animacy and gender information about nouns and that the information acquired significantly boosts the accuracy of salience-based pronoun resolution.

My results also suggest that shallow solutions tailored to specific syntactic problems can achieve performance on those problems that equal, or even exceed, that of more sophisticated general purpose models, like parsers. This is quite understandable; as statistical models for parsing are trained using an evaluation criteria that involves many syntactic constructs, it is quite plausible that they are not optimised for my specific tasks.

Transformation

I have expanded on the number of syntactic constructs previously handled by *transformation* stages in text simplification systems. The PSET project considered only coordination and voice change. I have offered a treatment of relative clauses, coordination, subordination and apposition. By extending the number of constructs simplified, my system generates simplified news reports with an average sentence length of around 15 words, down from 25 words for Guardian news reports. I have described how sentences can be simplified recursively and how transform-order can be guided by constraints on sentence-order.

Regeneration

I have presented a detailed analysis of the discourse-level issues that arise from sentence-level syntactic transformations. I have demonstrated that it is necessary to offer a treatment of generation issues like cue-word selection, sentence order, referring-expression generation, determiner choice and pronominal usage in order to preserve cohesion in the simplified text. I have shown that to preserve conjunctive cohesion and anaphoric coherence, it is necessary to model both intentional structure and attentional state.

I have also described an algorithm for generating referring expressions that can be used in any domain. My algorithm selects attributes and relations that are distinctive in context. It does not rely on the availability of an adjective classification scheme and uses WordNet antonym and synonym lists instead. It is also, as far as I know, the first algorithm that allows for the incremental incorporation of relations in the referring expression.

I have also discussed the idea of an anaphoric post-processor for rewritten text. The post-processor models attentional state in the rewritten text and determines where pronominal use is inappropriate. I believe that this post processor is general enough to be used in other applications that involve rewriting text, such as translation and summarisation.

7.2 Scope for Improvement

In this thesis, I have described how text can be syntactically simplified by making new sentences out of relative clauses, appositives and conjoined clauses. The results presented in chapter 6 suggest that there is scope for improvement when simplifying all three of these constructs. I now discuss the problems with simplifying each of these constructs.

7.2.1 Relative Clauses

Incorrect relative clause attachment remains the most important source of errors in the simplified text. Unlike other analysis errors, which tend to result in ungrammatical or incoherent text, incorrect attachment results in grammatical text with the wrong meaning. For example, consider #66 in appendix B.1. Due to two incorrect attachment decisions, the sentence:

> Sharif, 27, is thought to have been the accomplice of fellow Briton Asif Hanif, 21, who died after setting off explosives during the attack in Tel Aviv, which killed three people.

gets simplified to:

> Sharif, 27, is thought to have been the accomplice of fellow Briton Asif Hanif, 21. This accomplice died after setting off explosives during the attack in Tel Aviv. Tel Aviv killed three people.

It is clear that relative clause attachment can never by decided with 100% accuracy. And if a decision is made to only simplify relative clauses that have unambiguous attachment, the coverage is halved. It is possible that some further improvement in performance is possible by lexicalisation over the verb in the relative clause, following Clark and Weir (2000) (cf. section 3.2). Then, for example, it would be possible to deduce that the relative clause attaches to *dog* rather than *nose* in the example below by observing that *dog*s run more often than *nose*s in a corpus of news reports.

> Dogs with long noses that run fast tend to be expensive.

Another option might be to generate confidence levels for attachment decisions and use those to decide whether to carry out the simplification or not. More research is required to investigate the feasibility of that option.

7.2.2 *Appositives*

An examination of the examples in appendix B.1 suggests that my approach to simplifying apposition often results in awkward text. The problem arises because my treatment of appositives as parenthetical units is too simplistic. A better rhetorical treatment of apposition could result in more fluent output. For example, consider #1 in appendix B.1. In the sentence:

> Argentina's former president, Carlos Menem, was last night on the brink of throwing in the towel on his re-election bid, as aides admitted that he was ready to withdraw from this Sunday's run-off vote.

the appositive *Carlos Menem* serves to *identify* rather than to *elaborate* the noun phrase *Argentina's former president*. If this had been recognised, the sentence could have been simplified to:

> Carlos Menem was Argentina's former president. Carlos Menem was last night on the brink of throwing in the towel on his re-election bid...

rather than the awkward:

> Argentina's former president was Carlos Menem. Argentina's former president was last night on the brink of throwing in the towel on his re-election bid...

that is generated by my system. It is clear that I need to treat the apposition in the example above differently from the apposition in, for example, #20:

> A Danish newspaper quoted Niels Joergen Secher, a Danish doctor at Riyadh's King Faisal hospital, as saying between 40 to 50 bodies were brought to his hospital.

where my system correctly generates:

> A Danish newspaper quoted Niels Joergen Secher as saying between 40 to 50 bodies were brought to his hospital. Niels Joergen Secher was a Danish doctor at Riyadh's King Faisal hospital.

My treatment of apposition would also benefit from a more intelligent algorithm for deciding the auxiliary verb. My system uses a baseline approach that ensures that the tense of the copula construction is the same as the tense of the other simplified sentence. The rationale behind this was to avoid having frequent tense changes in the simplified text. However, it is clear that this baseline is too naive. The choice of auxiliary verb can be guided by the appositive, as well as the tense of the sentence. For example, consider the apposition in the sentence below:

> Pierre Vinken, 61 last month, has decided to...

The only acceptable auxiliary verb for the copula construction is the singular past tense:

> Pierre Vinken was 61 last month.

This is determined by the appositive, rather than the main clause. In general, the choice of the auxiliary verb can become quite involved when the information content of the copula construction is not valid for all times. Consider:

> The Labour Party, an important constituent of Mr Sharon's broad-based coalition, has pulled out of the government.

The past tense in the copula construct appears preferable on the basis that the Labour Party is no longer a member of the coalition; compare (a) with (b) below:

(a) The Labour Party has pulled out of the government. The Labour Party was an important constituent of Mr Sharon's broad-based coalition.

(b) The Labour Party has pulled out of the government. The Labour Party is an important constituent of Mr Sharon's broad-based coalition.

This example suggests that an intelligent choice of auxiliary verb would require a semantic treatment, rather than the discourse level treatment offered in this thesis.

7.2.3 Conjoined Clauses

The separation of conjoined clauses works well when the rhetorical relation is *concession* or *justify*, as there are convenient cue-words like *but* and *so* that can be used to signal the relation in the simplified text. However, all three judges in the evaluation commented that my use of the cue-phrase *This AUX X* for other relations was generally awkward. There are instances where the simplified text is acceptable without a cue-phrase. Consider #31, where the sentence:

> Two weeks ago the United States said it was removing virtually all forces from the kingdom as they were no longer needed after the war in Iraq toppled Saddam Hussein.

was simplified by my system to:

> Two weeks ago the United States said it was removing virtually all forces from the kingdom. This was as they were no longer needed after the war in Iraq toppled Saddam Hussein.

In this example, the simplified text:

> Two weeks ago the United States said it was removing virtually all forces from the kingdom. They were no longer needed after the war in Iraq toppled Saddam Hussein.

is more fluent, without disturbing conjunctive cohesion too much. If such cases can be detected, the generated text might be more fluent. It appears that the temporal relations signalled by the conjunctions *after* and *before* can be made implicit by using a particular sentence order. For example, #17:

> Between 40 and 50 people were feared dead today after a series of suicide bomb explosions rocked the Saudi capital, Riyadh, overnight.

can be simplified to:

> A series of suicide bomb explosions rocked the Saudi capital, Riyadh, overnight. Between 40 and 50 people were feared dead today.

More analysis of data is required in order to find out whether this is always a feasible option.

7.3 Future Work

As discussed in section 7.2, there is scope for improvement in my treatment of relative clauses, apposition and conjunction. Other future work on my text simplification system would include implementing a lexical simplification module and performing a comprehension-based evaluation on end users with low reading ages.

In addition to improving, extending and evaluating my text simplification system, I am also interested in addressing a number of interesting questions that this thesis has raised. The first is the use of text simplification as a preprocessor for other NLP tasks; in particular, parsing, translation and summarisation.

Preliminary results in this thesis suggest that text simplification has the potential to increase the throughput of a parser without degrading its performance. Like parsers, the performance of machine translation systems also decreases with increased sentence length (Gerber and Hovy, 1998). It would be interesting to investigate whether simplified sentences would be easier to translate correctly. Preliminary experiments using SYSTRAN[25] (to translate from English to German) suggests two possible reasons why syntactically simplified text might translate better. The first is that SYSTRAN can get the gender of relative pronouns wrong in the translated text (German, unlike English, uses a gender system for relative pronouns). If the relative pronoun is removed using syntactic simplification, that error can be avoided. The second is that there are many sources of stiltedness in translation. Individual words and local phrase structures can get translated poorly by a program. For long sentences, these sources of stiltedness can combine to result in translated sentences that are very difficult to read. If a long sentence is split into two or more before translation, it is possible that at least one of the translated sentences might be easily understandable. As an example, SYSTRAN translated the sentence:

> A Danish newspaper quoted Niels Joergen Secher, a Danish doctor at Riyadh's King Faisal hospital, as saying between 40 to 50 bodies were brought to his hospital.

to:

> Ein dänische Zeitung veranschlagenes Niels Joergen Secher, ein dänischer Doktor am Krankenhaus des Königs Faisal Riyadhs, wie, sagend zwischen 40 bis 50 Körpern, wurden zu seinem Krankenhaus geholt.

This is a messy translation that is difficult to understand. When the English sentence is simplified prior to translation, SYSTRAN produces:

> Eine dänische Zeitung veranschlug Niels Joergen Secher, wie, sagend zwischen 40 bis 50 Körpern, zu seinem Krankenhaus geholt wurden. Niels Joergen Secher war ein dänischer Doktor am Krankenhaus des Königs Faisal Riyadhs.

In this translation, the second sentence comes out reasonably clean, and the messiness is confined to the first. This example is presented only to illustrate how simplification might aid translation and further empirical work is required before it can be claimed that it indeed does.

There are also potential uses of simplification in summarisation that could be explored. Syntactic simplification results in less information content per sentence. This is likely to improve the performance of summarisation systems that are based on sentence extraction, because smaller units of information are being extracted. It is possible the simplified

[25] Online translations are available at http://www.systran.org.

sentences in the summary can then be recombined by using my simplification rules backwards. For example, suppose the following sentences were selected for a summary:

(a) Colin Powell said at the weekend that resuming diplomatic relations was not on the table.

(b) Colin Powell is the US secretary of State.

Then, the simplification rule for apposition could be used backwards:

$$\texttt{U V, W, X.} \longleftarrow \begin{array}{ll} (a) & \texttt{U V X.} \\ (b) & \texttt{V Aux W.} \end{array}$$

to perform sentence aggregation and generate the following sentence in the summary:

> Colin Powell, the US secretary of State, said at the weekend that resuming diplomatic relations was not on the table.

Inverse syntactic-simplification, or sentence aggregation, has already been shown to be useful in summarisation (McKeown et al., 1995; Shaw, 1998). A possible algorithm for summarisation might now be:

1. Perform syntactic simplification on the original text (or texts for multiple sources).
2. Use information theoretic measures like $tf{*}idf$[26] to select sentences for the summary.
3. Prune sentences by removing unnecessary modifiers (adjectives, adverbs, prepositional phrases). This can be achieved using sentence shortening (cf. section 1.3.2).
4. Use inverse syntactic-simplification to combine sentences in the summary, as illustrated by the example above.
5. Resolve cohesion issues like sentence ordering and anaphoric usage.

It needs to be emphasised that this algorithm is presented here only as a possible direction for future work. In particular, I have no empirical evidence that this algorithm would be useful.

Simplification, summarisation and translation are all tasks that involve transforming text at the sub-sentence or sentence levels. It is therefore likely that the discourse level issues of cohesion and coherence associated with each of them will be similar. It would be

[26] Measuring information content is a key problem in information retrieval and text summarisation. The $tf * idf$ metric measures the information value of a word by dividing the term frequency (the frequency of the word in the document) by the inverted document frequency (the number of documents in the collection that word features in). Then, a word that features frequently in one particular document, but rarely in others, is highly informative about that document. The information content of a sentence is measured by aggregating the information content of its words by some means, addition being the obvious way (Luhn, 1958; Edmundson, 1964; Rath et al., 1961).

interesting to compare the issues of text cohesion that arise in these three domains and develop on the idea of a generic discourse-level post-processor.

I am also interested in exploring the utility of my referring-expression generator in other applications and other genre. As described in section 5.3, my algorithm is suitable for open domains and can be easily ported across applications. It would be interesting to compare it with a domain specific referring-expression generator on that domain.

I am aware of studies of how humans resolve relative clause attachment that address issues of structure (tendency to attach locally for English)(Cuetos and Mitchell, 1988; Gilboy et al., 1995; Fernandez, 2000) and lexicalisation (tendency to attach wide for select prepositions) (Felser et al., To appear; Felser et al., 2003). My corpus analysis confirms that both these strategies are effective on written text; a strategy of attaching ambiguous cases locally gives an accuracy of 65-70%, as does a strategy of always attaching according to the preposition. An interesting result from my research is that a strategy of attaching ambiguous relative clauses according to salience gives better results still. It would be interesting to conduct experiments to see if humans use discourse structure to resolve attachment ambiguities; ie. whether they make attachment decisions differently when presented with a sentence in context rather than a sentence in isolation.

A Guidelines for Annotators

A.1 Guidelines for Evaluating the Analysis Stage

General Instructions

Every instance of a # or a [needs to be marked with a tick or a cross. The #s correspond to attachment decisions and the [s correspond to identification decisions.

Appositives

Appositives are marked-up as $[_{appos} \cdots {}_{appos}]$. As an illustration, consider:

> Mr. Vinken^1 is the chairman^2 of Elsevier N.V. ^{3}, $[_{appos}$ the Dutch publishing group $^{4\#3}$ $_{appos}]$.

- For the appositive identification task, ask yourself the following questions:
    1. Is "`X` is/are `Appositive`" a grammatical sentence? For the example above, is *Elsevier N.V. is the Dutch publishing group.* a grammatical sentence with the correct meaning?
    2. Remove the appositive. Is what remains a grammatical sentence? For the example above, is *Mr. Vinken is chairman of Elsevier N.V.* a grammatical sentence with the correct meaning?

  If the answers to both are *yes*, the appositive is correctly identified, otherwise it is wrongly identified.

- For the appositive attachment task, find the noun phrase with the index n given by the $^{\#n}$ within the appositive. Is this correct? In the example above, does *the Dutch publishing group* refer to *the chairman of Elsevier N.V.* or does it refer to *Elsevier N.V.*?

- Note: both attachment and identification are correct for the example above.

Relative Clauses

Relative Clauses are marked-up as $[_{clause} \cdots {}_{clause}]$. The first word in the relative clause is *who*, *which* or *that*. As an illustration, consider:

> The business side[1] is run by Robert Gottlieb[2], $[_{clause}$who$^{3\#2}$ left Random House's Alfred A. Knopf to run the New Yorker$_{clause}]$, also owned by the Newhouse family.

- For the relative-clause identification task, ask yourself the following questions:
  1. Is the marked-up clause with the relative pronoun replaced by the noun phrase it attaches to a grammatical sentence? For the example above, is *Robert Gottlieb left Random House 's Alfred A. Knopf to run the New Yorker* a grammatical sentence with the correct meaning?
  2. Remove the clause and the relevant commas. Is what remains a grammatical sentence? For the example above, is *The business side*[1] *is run by Robert Gottlieb, also owned by the Newhouse family* a grammatical sentence with the correct meaning?

  If the answers to both are *yes*, the relative clause is correctly identified, otherwise it is wrongly identified.
- For the relative clause attachment task, follow the instructions for appositive attachment above
- Note: the attachment is correct in the example above, but the identification is *wrong* as the answer to question 2 is *No*.

Conjoined Clauses

Conjoined clauses are marked-up as $[_{clause}\cdots{}_{clause}]$. The first word in the conjoined clause is either a coordinating conjunction like *and* or a subordinating conjunction like *but*, *when*, *though*, *before*... There are no attachment decisions that need to be made here. Just decide if the what is marked-up is a clause or not. When you remove the conjunction, is what is left in the clause a grammatical sentence? Does it have a verb and a subject? If the answers to these questions are *yes* then the clause is identified correctly. Examples of incorrect identification follow:

> Last March, after attending a teaching seminar in Washington, Mrs. Yeargin says she returned to Greenville two days $[_{clause}$ before annual testing feeling that she hadn't prepared her low-ability geography students adequately$_{clause}]$.

> The average maturity for funds open only to institutions, considered by some to be a stronger indicator $[_{clause}$ because those managers watch the market closely, reached a high point for the year – 33 days $_{clause}]$.

A.2 Guidelines for Evaluating Grammaticality, Meaning and Cohesion

- The evaluation set contains 95 examples of sentences that have been simplified by my program. Each example consists of the original sentence, followed by the simplified sentences (that are indented).

- Each example contains the following fields that you are required to fill:
  Grammaticality (y/n):
  Meaning Preservation (0-3):

  - Please fill the Grammaticality field with a "n" if ANY of the simplified sentences are ungrammatical in your judgement.
  - Please score the Meaning Preservation field between 0 and 3. These values correspond to:

    0: The information content (predicative meaning) of the simplified sentences differs from that of the original.

    1: The information content of the simplified sentences is the same as that of the original. However, the authors intensions for presenting that information has been drastically compromised, making the simplified text incoherent.

    2: The information content of the simplified sentences is the same as that of the original. However, the author's intensions for presenting that information have been subtly altered, making the simplified text slightly less coherent.

    3: The simplified text preserves both meaning and coherence.

- If you answered "n" for grammaticality, use your discretion for whether to answer the meaning preservation part. If the sentence is completely garbled, ignore the meaning part. But if theres a minor grammaticality problem, please answer the meaning part.

B Data Set for Evaluation

B.1 Data Set annotated with Results

1) Argentina's former president, Carlos Menem, was last night on the brink of throwing in the towel on his re- election bid, as aides admitted that he was ready to withdraw from this Sunday's run-off vote.

> Argentina's former president was Carlos Menem.
>
> Argentina's former president was last night on the brink of throwing in the towel on his re-election bid.
>
> This was as aides admitted that he was ready to withdraw from this Sunday's run-off vote.

Grammaticality (y/n): nyy[27]
Meaning Preservation (0-3):323[28]

---

2) Mr Menem, 72, was expected to announce his decision this morning, after opinion polls showed he would suffer a humiliating defeat in the second round vote against Nestor Kirchner, a fellow Peronist and a regional governor.

> Mr Menem, 72, was expected to announce his decision this morning.
>
> This was after opinion polls showed he would suffer a humiliating defeat in the second round vote against Nestor Kirchner, a fellow Peronist and a regional governor.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

3) But the extrovert veteran with a taste for young television starlets and expensive Italian sports cars, quickly slumped in polls for the run-off, which predicted a 63 % vote for Mr Kirchner.

> But the extrovert veteran with a taste for young television starlets and expensive Italian sports cars, quickly slumped in polls for the run-off.
>
> These polls predicted a 63 % vote for Mr Kirchner.

[27] nyy represents the result where judge$_1$ decided *no*, judge$_2$ decided *yes* and judge$_3$ decided *yes*.

[28] 323 represents the result where judge$_1$ decided 3, judge$_2$ decided 2 and judge$_3$ decided 3.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):333

4) Some aides had suggested it would be better to avoid such a humiliation, but not all agreed.

> Some aides had suggested it would be better to avoid such a humiliation.
>
> But not all agreed.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

5) “I’m highly critical of the decision,” said a top aide, Diego Guelar, a former ambassador to the United States.

> “I’m highly critical of the decision,” said a top aide.
>
> This top aide is Diego Guelar, a former ambassador to the United States.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

6) “It is time to bury old ghosts from the past,” one said, although tacitly officials realise that the move will deprive Mr Kirchner of a strong election win which would have strengthened his legitimacy to lead Argentina through troubled times.

> “It is time to bury old ghosts from the past,” one said.
>
> But tacitly officials realise that the move will deprive Mr Kirchner of a strong election win.
>
> This strong election win would have strengthened his legitimacy to lead Argentina through troubled times.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):233

---

7) He first rose to national prominence as a proteg of his party’s founder, General Juan Peron, back in the early 1970s, when sporting shoulder-length hair and bushy side-whiskers, he became the young governor of his northern province of La Rioja.

> He first rose to national prominence as a proteg of his party’s founder, General Juan Peron, back in the early 1970s.
>
> This was when sporting shoulder-length hair and bushy side-whiskers, he became the young governor of his northern province of La Rioja.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):333

---

8) Arrested in the bloody military coup of 1976, he went back into politics after the return of democracy in 1983, and won the presidential elections of 1989, causing panic among Argentina's bankers and businessmen who feared his populist leanings.

> Arrested in the bloody military coup of 1976, he went back into politics after the return of democracy in 1983, and won the presidential elections of 1989, causing panic among Argentina's bankers and some businessmen.
>
> These Businessmen feared his populist leanings.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):330

---

9) The spectacular first-term economic growth made him wildly popular, but the high unemployment and deep recession that followed his re-election seemed to spell the end of his political career when he left office in 1999.

> The spectacular first-term economic growth made him wildly popular, but the high unemployment and deep recession that followed his re-election seemed to spell the end of his political career.
>
> This was when he left office in 1999.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):333

---

10) His resignation also ended Mr Menem's long-cherished dream of matching the record of his party's founder General Peron, who won the Argentinian presidency three times.

> His resignation also ended Mr Menem's long-cherished dream of matching the record of his party's founder General Peron.
>
> General Peron won the Argentinian presidency three times.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

11) Mrs Fitzgerald, 60, said she and her husband had gone to bed at around 11pm last Friday when they heard a loud bang in their garage.

> Mrs Fitzgerald, 60, said she and her husband had gone to bed at around 11pm last Friday.
>
> This was when they heard a loud bang in their garage.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):033

---

12) Mrs Fitzgerald, who had come downstairs and was standing behind her husband at the time, said : "It was like something out of a horror movie, he was bleeding so badly."

> Mrs Fitzgerald said : “It was like something out of a horror movie, he was bleeding so badly.”
>
> Mrs Fitzgerald had come downstairs and was standing behind her husband at the time.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

13) She called an ambulance which took Mr Fitzgerald to Worcestershire Royal Hospital, Worcester, but doctors decided he needed plastic surgery in Birmingham.

> An ambulance took Mr Fitzgerald to Worcestershire Royal Hospital, Worcester.
>
> She called this ambulance.
>
> But doctors decided he needed plastic surgery in Birmingham.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):313

---

14) Worcestershire Badger Society put down Boris after catching him in a trap laid on the Fitzgeralds’ front lawn, but not before he had chased pursuing police officer onto the bonnet of their car.

> Worcestershire Badger Society put down Boris after catching him in a trap laid on the Fitzgeralds’ front lawn.
>
> But not before he had chased pursuing police officer onto the bonnet of their car.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):333

---

15)
“I have been involved with badgers for 24 years and I have never heard of anything like this.”

> “I have been involved with badgers for 24 years.
>
> I have never heard of anything like this.”

Grammaticality (y/n):yyn
Meaning Preservation (0-3):333

---

16) Weaver said badgers were notoriously powerful animals and the incident showed the folly of trying to turn wild animals into pets.

> Weaver said badgers were notoriously powerful animals.
>
> The incident showed the folly of trying to turn wild animals into pets.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

17) Between 40 and 50 people were feared dead today after a series of suicide bomb explosions rocked the Saudi capital, Riyadh, overnight.

> Between 40 and 50 people were feared dead today.
>
> This was after a series of suicide bomb explosions rocked the Saudi capital, Riyadh, overnight.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

18) "It seems we have lost 10 Americans killed, many other nationalities were also killed," the US secretary of state, Colin Powell, told reporters as he arrived at Riyadh airport earlier today, within hours of the devastating attacks.

> The US secretary of State is Colin Powell.
>
> "It seems we have lost 10 Americans killed, many other nationalities were also killed," the US secretary of state told reporters.
>
> This is as he arrived at Riyadh airport earlier today, within hours of the devastating attacks.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):331

---

19) One Australian man was also killed and another injured in the four bomb blasts that ripped through foreign housing compounds, according to the Australian government.

> One Australian man was also killed.
>
> Another injured in some four bomb blasts.
>
> These four bomb blasts ripped through foreign housing compounds, according to the Australian government.

Grammaticality (y/n):nnn
Meaning Preservation (0-3):-21

---

20) A Danish newspaper quoted Niels Joergen Secher, a Danish doctor at Riyadh's King Faisal hospital, as saying between 40 to 50 bodies were brought to his hospital.

> A Danish newspaper quoted Niels Joergen Secher as saying between 40 to 50 bodies were brought to his hospital.
>
> Niels Joergen Secher was a Danish doctor at Riyadh's King Faisal hospital.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

21)
"We believe there are a small number of British nationals who have been injured, not seriously."

> “We believe there are a small number of some British nationals.
>
> These British nationals have been injured, not seriously.”

Grammaticality (y/n):yyn
Meaning Preservation (0-3):313

---

22) Mr Powell was greeted on his arrival by Prince Saud, the Saudi foreign minister, who expressed his sorrow and vowed to cooperate with the United States in fighting terrorism.

> Mr Powell was greeted on his arrival by Prince Saud, the Saudi foreign minister.
>
> Prince Saud expressed his sorrow and vowed to cooperate with the United States in fighting terrorism.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

23) No group has yet claimed responsibility for the attacks, but Mr Powell said it bore “all the hallmarks” of al-Qaida and its Saudi-born leader, Osama bin Laden.

> No group has yet claimed responsibility for the attacks.
>
> But Mr Powell said it bore “all the hallmarks” of al-Qaida and its Saudi-born leader.
>
> Its Saudi-born leader was Osama bin Laden.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):221

---

24) According to reports, security guards fought a furious gun battle with the terrorists as they tried to prevent one of the attacks.

> According to reports, security guards fought a furious gun battle with the terrorists.
>
> This is as they tried to prevent one of the attacks.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):231

---

25) Witnesses said they had heard three blasts, which sent fireballs into the night sky above the Gharnata, Ishbiliya and Cordoba compounds.

> Witnesses said they had heard three blasts.
>
> These three blasts sent fireballs into the night sky above the Gharnata, Ishbiliya and Cordoba compounds.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

26) Television pictures showed scenes of devastation as emergency vehicles raced through Riyadh's streets.

> Television pictures showed scenes of devastation.
>
> This was as emergency vehicles raced through Riyadh's streets.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

27) Cars and pick-up trucks with badly twisted and still smouldering frames littered the three compounds, which housed villas and four-storey blocks.

> Cars and pick-up trucks with badly twisted and still smouldering frames littered the three compounds.
>
> These three compounds housed villas and four-storey blocks.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

28) "We were sleeping when we were woken up by the sound of gunfire," he told the Arab News newspaper.

> "We were sleeping.
>
> This was when we were woken up by the sound of gunfire," he told the Arab News newspaper.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

29) "Moments later, a loud explosion was heard followed by another, bigger explosion."

> "Moments later, a loud explosion was heard followed by another.
>
> This loud explosion was bigger explosion."

Grammaticality (y/n):nnn
Meaning Preservation (0-3):320

---

30) The Saudi interior minister, Prince Nayef, told local newspapers the attackers could be linked to the discovery of a large weapons cache on May 6.

> The Saudi interior minister told local newspapers the attackers could be linked to the discovery of a large weapons cache on May 6.
>
> This Saudi interior minister was Prince Nayef.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

31) Two weeks ago the United States said it was removing virtually all forces from the kingdom as they were no longer needed after the war in Iraq toppled Saddam Hussein.

> Two weeks ago the United States said it was removing virtually all forces from the kingdom.
>
> This was as they were no longer needed after the war in Iraq toppled Saddam Hussein.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

32) Last night's attacks throw fresh doubt on the safety of westerners in Saudi Arabia, but they also strengthen the case of six Britons held in Saudi over earlier bomb attacks, a leading legal campaigner said today.

> Last night's attacks throw fresh doubt on the safety of westerners in Saudi Arabia.
>
> But they also strengthen the case of six Britons held in Saudi over earlier bomb attacks, a leading legal campaigner said today.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

33) In one of the attacks Briton Christopher Rodway, 47, was killed when his car was blown up.

> In one of the attacks Briton Christopher Rodway, 47, was killed.
>
> This was when his car was blown up.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

34) At least 60 people were killed after a gas explosion ripped through a coal mine in eastern China, state television reported today.

> At least 60 people were killed.
>
> This was after a gas explosion ripped through a coal mine in eastern China, state television reported today.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):320

---

35) Authorities said that another 23 miners were trapped 500 metres below ground following the blast - but there was little hope for their survival.

> Authorities said that another 23 miners were trapped 500 metres below ground following the blast.
>
> But there was little hope for their survival.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

36) An official at the local bureau in charge of mines said rescuers had recovered 63 bodies this morning and there was little hope those missing would be found alive after the explosion, the latest major accident in China's mining industry.

> An official at the local bureau in charge of mines said rescuers had recovered 63 bodies this morning.
>
> There was little hope those missing would be found alive after the explosion.
>
> This explosion was the latest major accident in China's mining industry.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):331

37) "I think their chance for survival is very small," the official, who declined to give his name, said.

> "I think their chance for survival is very small," the official said.
>
> This official declined to give his name.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

38) State Administration of Work Safety said on its website that more than 100 workers were in the Luling coal mine in the city of Huaibei, 420 miles south of Beijing, when the blast occurred at 4 : 13 p.m. (0813 GMT) yesterday.

> Huaibei is 420 miles south of Beijing.
>
> State Administration of Work Safety said on its website that more than 100 workers were in the Luling coal mine in the city of Huaibei.
>
> This is when the blast occurred at 4 : 13 p.m. (0813 GMT) yesterday.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):221

---

39) The local official said the miners were working 500 metres to 600 metres below the surface when the explosion occurred.

> The local official said the miners were working 500 metres to 600 metres below the surface.
>
> This was when the explosion occurred.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):233

---

40) Explosions are common and often are blamed on a lack of ventilation to clear natural gas that seeps out of the coal bed.

> Explosions are common and often are blamed on a lack of ventilation to clear natural gas.
>
> This natural gas seeps out of the coal bed.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

41) The inspectors' concerns are shared internationally and the British government has report edly offered to raise the matter with Washington to try to get agreement on a return of the UN nuclear inspectors to Iraq.

> The inspectors' concerns are shared internationally.
>
> The British government has reportedly offered to raise the matter with Washington to try to get agreement on a return of the UN nuclear inspectors to Iraq.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

42) The main worry revolves around the fate of at least 200 radioactive isotopes which were stored at the sprawling al- Tuwaitha nuclear complex, 15 miles south of Baghdad.

> The main worry revolves around the fate of at least 200 radioactive isotopes.
>
> These radioactive isotopes were stored at the sprawling al-Tuwaitha nuclear complex.
>
> This Tuwaitha nuclear complex was 15 miles south of Baghdad.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):331

---

43) It has seen widespread looting, and reports from Baghdad speak of locals making off with barrels of raw uranium and the isotopes which are meant for medical or industrial use.

> It has seen widespread looting, and reports from Baghdad speak of locals making off with barrels of raw uranium and some isotopes.
>
> These isotopes are meant for medical or industrial use.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

44) “If this happened anywhere else there would be national outrage and it would be the highest priority,” said a senior source at the UN nuclear watchdog, the Vienna-based International Atomic Energy Agency.

> “If this happened anywhere else there would be national outrage and it would be the highest priority,” said a senior source at the UN nuclear watchdog.
>
> This Nuclear watchdog was the Vienna-based International Atomic Energy Agency.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):322

---

45) “The radioactive sources, some very potent ones, could get on to the black market and into the hands of terrorists planning dirty-bomb attacks,” said Melissa Fleming, an IAEA spokeswoman.

> “The radioactive sources, some very potent ones, could get on to the black market and into the hands of terrorists planning dirty-bomb attacks,” said Melissa Fleming.
>
> Melissa Fleming is an IAEA spokeswoman.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

46) The IAEA chief, Mohammed El Baradei, has appealed twice to the US in the past month to be allowed to resume inspections of the Iraqi nuclear sites.

> The IAEA chief has appealed twice to the US in the past month to be allowed to resume inspections of the Iraqi nuclear sites.
>
> This IAEA chief is Mohammed El Baradei.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

47) The requests have gone unanswered, although the IAEA has forwarded details of suspect nuclear sites to the US.

> The requests have gone unanswered.
>
> But the IAEA has forwarded details of suspect nuclear sites to the US.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):223

---

48) On Monday, Dr El Baradei raised the problem in London with the foreign secretary, Jack Straw, who is said to have been “supportive and sympathetic”.

> On Monday, Dr El Baradei raised the problem in London with the foreign secretary.

> This foreign secretary was Jack Straw.
>
> This foreign secretary is said to have been “supportive and sympathetic”.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):331

---

49) Mark Gvozdecky, the chief IAEA spokesman, said : If this was happening anywhere else in the world, we would insist on an immediate inspection.

> Mark Gvozdecky said: If this was happening anywhere else in the world, we would insist on an immediate inspection.
>
> Mark Gvozdecky was the chief IAEA spokesman.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):232

---

50) It has been more than a month since the initial reports of looting, more than a month since US forces took control.

> It has been more than a month since the initial reports of looting, more than a month.
>
> This is since US forces took control.

Grammaticality (y/n):nny
Meaning Preservation (0-3):322

---

51) An anaesthetist who murdered his girlfriend with a Kalashnikov souvenir of his days as an SAS trooper, was struck off the medical register yesterday, five years later.

> An anaesthetist, was struck off the medical register yesterday, five years later.
>
> This anaesthetist murdered his girlfriend with a Kalashnikov souvenir of his days as an SAS trooper.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):223

---

52) A hearing of the General Medical Council, convened after the appeal court in March had upheld his murder conviction and life sentence, dealt rapidly with the case.

> A hearing of the General Medical Council, convened.
>
> This is after the appeal court in March had upheld his murder conviction and life sentence, dealt rapidly with the case.

Grammaticality (y/n):nyn
Meaning Preservation (0-3):130

---

53) Shanks used the automatic rifle to kill nurse Vicky Fletcher, 21, when she ended their relationship at Pontefract general hospital, West Yorkshire, in May 1998.

> Shanks used the automatic rifle to kill nurse Vicky Fletcher, 21.
>
> This was when she ended their relationship at Pontefract general hospital, West Yorkshire, in May 1998.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):330

---

54) A crown court jury heard how he ambushed her and shot her as she tried to escape - then drove off, apparently without emotion.

> A crown court jury heard how he ambushed her and shot her.
>
> This was as she tried to escape - then drove off, apparently without emotion.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):210

---

55) Although the two countries have not had diplomatic relations since the US-backed shah was overthrown in 1979, officials on both sides have acknowledged that ongoing low-key talks on regional issues and Iran's nuclear programme will resume this month.

> The two countries have not had diplomatic relations since the US-backed shah was overthrown in 1979.
>
> But officials on both sides have acknowledged that ongoing low-key talks on regional issues and Iran's nuclear programme will resume this month.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

56) The influential former president, Hashemi Rafsanjani, recently even suggested a referendum on restoring diplomatic relations, creating a stir in a country where state television still refers to America as the "Great Satan".

> The influential former president recently even suggested a referendum on restoring diplomatic relations, creating a stir in a country where state television still refers to America as the "Great Satan".
>
> This influential former president was Hashemi Rafsanjani.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):233

---

57) The US secretary of state, Colin Powell, said at the weekend that resuming diplomatic relations was not on the table, but that the governments were speaking "in light of the changed strategic situation".

> The US secretary of state is Colin Powell.

> This US secretary of state said at the weekend that resuming diplomatic relations was not on the table, but that the governments were speaking “in light of the changed strategic situation”.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

58) One former member of the establishment, Ayatollah Hosein Ali Montazeri, who has become its most prominent critic, dared to say in public that the clerical leadership could face the same fate as Saddam Hussein if it continued its autocratic ways.

> One former member of the establishment dared to say in public that the clerical leadership could face the same fate as Saddam Hussein if it continued its autocratic ways.
>
> This former member was Ayatollah Hosein Ali Montazeri.
>
> This former member has become its most prominent critic.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):222

---

59) The hardliners, who have blocked attempts at reform by President Mohammad Khatami and his allies, have drawn a different lesson from the Iraq conflict.

> The hardliners have drawn a different lesson from the Iraq conflict.
>
> These hardliners have blocked attempts at reform by President Mohammad Khatami and his allies.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):313

---

60) In recent weeks the judiciary and security services have targeted independent journalists who turned to the internet after their newspapers were shut down, subjecting them to detention without trial and interrogation.

> In recent weeks the judiciary and security services have targeted some independent journalists were shut down, subjecting them to detention without trial and interrogation.
>
> These independent journalists turned to the internet after their newspapers.

Grammaticality (y/n):nnn
Meaning Preservation (0-3):-00

---

61) Earlier this month, MPs were told their comments would be “monitored” to safeguard national security, a clear message aimed at intimidating reformists, who form a majority in parliament.

> Earlier this month, MPs were told their comments would be "monitored" to safeguard national security, a clear message aimed at intimidating reformists.
>
> These Reformists form a majority in parliament.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

62) Ahmad Jannati, a leading cleric, told worshippers this month : "Iraqis will eventually reach the conclusion that the only way to oust Americans is an intifada."

> Ahmad Jannati is a leading cleric.
>
> Ahmad Jannati told worshippers this month :
> "Iraqis will eventually reach the conclusion that the only way to oust Americans is an intifada."

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

63) However, the most serious threat in a post-Saddam world may come from Iraq's dormant oilfields, which are already attracting the interest of foreign oil companies.

> However, the most serious threat in a post-Saddam world may come from Iraq's dormant oilfields.
>
> Iraq's dormant oilfields are already attracting the interest of foreign oil companies.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

64) The revival of Iraq's oil industry could eventually drive down oil prices, possibly triggering a social crisis in Iran, which relies on its oil income to keep the economy afloat.

> The revival of Iraq's oil industry could eventually drive down oil prices, possibly triggering a social crisis in Iran.
>
> Iran relies on its oil income to keep the economy afloat.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

65) British embassy officials were in Israel today conducting inquiries after the body was discovered yesterday, the Foreign Office said.

> British embassy officials were in Israel today conducting inquiries.
>
> This was after the body was discovered yesterday, the Foreign Office said.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):231

---

66) Sharif, 27, is thought to have been the accomplice of fellow Briton Asif Hanif, 21, who died after setting off explosives during the attack in Tel Aviv, which killed three people.

> Sharif, 27, is thought to have been the accomplice of fellow Briton Asif Hanif, 21.
>
> This accomplice died after setting off explosives during the attack in Tel Aviv.
>
> Tel Aviv killed three people.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):000

---

67) Israeli authorities have been hunting Sharif since he vanished from the scene of the bombing on April 30, outside Mike's Place, a busy sea-front bar.

> Israeli authorities have been hunting Sharif.
>
> This is since he vanished from the scene of the bombing on April 30, outside Mike's Place.
>
> Mike's Place was a busy sea-front bar.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

68) It is thought he was carrying an explosive belt that failed to detonate.

> It is thought he was carrying an explosive belt.
>
> This explosive belt failed to detonate.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

---

69) British intelligence is helping Israel with its investigation into the attack, which was carried out hours after the Palestinian Authority installed Mahmoud Abbas as its first prime minister.

> British intelligence is helping Israel with its investigation into the attack.
>
> This attack was carried out hours after the Palestinian Authority installed Mahmoud Abbas as its first prime minister.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):233

---

70) Sharif, described by friends and neighbours as polite and caring as a child, was a public schoolboy who went to university in London, but is believed to have been radicalised by the teachings of an extreme cleric at Finsbury Park mosque and the al-Muhajiroun group.

> Sharif, described by friends and neighbours as polite and caring as a child, was a public schoolboy, but is believed to have been radicalised by the teachings of an

> extreme cleric at Finsbury Park mosque and the al-Muhajiroun group.
>
> This public schoolboy went to university in London.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):222

---

71) Sharif's brother, Zahid Hussain Sharif, 36, his wife, Tahira Shad Tabassum, 27, and Parveef, who are all from Derbyshire, are also accused of failing to pass on information to the authorities which may have prevented a terrorist act.

> Sharif's brother, Zahid Hussain Sharif, 36, his wife, Tahira Shad Tabassum, 27, and Parveef, who are all from Derbyshire, are also accused of failing to pass on information to some authorities.
>
> These authorities may have prevented a terrorist act.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):200

---

72) More than 100 have signed a petition protesting at the place where the Soviet leaders Lenin, Stalin, and Brezhnev, and the pioneer cosmonaut Yuri Gagarin, lie buried in Moscow being used for a rock concert which carries, they say, "a covert political meaning".

> More than 100 have signed a petition protesting at the place where the Soviet leaders Lenin, Stalin, and Brezhnev, and the pioneer cosmonaut Yuri Gagarin, lie buried in Moscow being used for a rock concert.
>
> This rock concert carries, they say, "a covert political meaning".

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

73) We await his answer and hope he will change his mind.

> We await his answer.
>
> And hope he will change his mind.

Grammaticality (y/n):nyy
Meaning Preservation (0-3):333

---

74) The Beatles were censored by Brezhnev's government, which would only let the Soviet press criticise rock music, derided as the "belch of western culture".

> The Beatles were censored by Brezhnev's government derided as the "belch of western culture".
>
> Brezhnev's government would only let the Soviet press criticise rock music.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):320

---

75) Sir Paul's second group, Wings, was also criticised, but perhaps because Soviet musical tastes had improved.

> Sir Paul's second group, Wings, was also criticised.
>
> But perhaps because Soviet musical tastes had improved.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

76) It is 35 years since The Beatles recorded Back in the USSR, which contained the premature lines : "Well, the Ukraine girls really knock me out. They leave the West behind."

> It is 35 years.
>
> This is since The Beatles recorded Back in the USSR.
>
> The USSR contained the premature lines :.
>
> "Well, the Ukraine girls really knock me out.
>
> They leave the West behind"

Grammaticality (y/n):yyy
Meaning Preservation (0-3):000

---

77) Police have recovered the car used by gunmen who murdered two teenagers as they celebrated new year in Birmingham, the detective leading the investigation said today.

> Police have recovered the car used by gunmen who murdered two teenagers.
>
> This is as they celebrated new year in Birmingham, the detective leading the investigation said today.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):111

---

78) Two shell casings used in the weapon which shot dead Charlene Ellis, 18, and 17-year -old Letisha Shakespeare, were found in the burnt-out red Ford Mondeo, detective superintendent Dave Mirfield, of West Midlands police, told reporters.

> Two shell casings used in a weapon, were found in the burnt-out red Ford Mondeo, detective superintendent Dave Mirfield, of West Midlands police, told reporters.
>
> This weapon shot dead Charlene Ellis, 18, and 17-year -old Letisha Shakespeare.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):333

---

79) The car, registration number P 941 UTG, was bought from a Northamptonshire motor trader on December 31 last year, two days before the shooting, which happened at the rear of a hairdresser's salon in Birchfield Road, Aston.

> The car was registration number P 941 UTG.
>
> This car was bought from a Northamptonshire motor trader on December 31 last year.
>
> December 31 last year was two days before the shooting.
>
> This shooting happened at the rear of a hairdresser's salon in Birchfield Road, Aston.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):323

---

80) It was an extremely detailed search of that vehicle which recovered those bullets.

> It was an extremely detailed search of that vehicle.
>
> This extremely detailed search recovered those bullets.

Grammaticality (y/n):yyn
Meaning Preservation (0-3):301

---

81) They paid cash for the vehicle, which was in "showroom" condition.

> They paid cash for the vehicle.
>
> This cash was in "showroom" condition.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):030

---

82) Airport operator BAA today lost its immunity from hostile takeover when an EU court ruling struck down the government's "golden share" in the company.

> Airport operator BAA today lost its immunity from hostile takeover.
>
> This was when an EU court ruling struck down the government's "golden share" in the company.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

83) BAA, which operates Heathrow and six other major airports, said that it would not oppose the government getting rid of its "golden share".

> BAA said that it would not oppose the government getting rid of its "golden share".
>
> BAA operates Heathrow and six other major airports.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

84) BAA has benefited from special government protection since it was privatised in 1987.

> BAA has benefited from special government protection.
>
> This is since it was privatised in 1987.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):233

85) According to BAA officials, it has been exercised once, when one shareholder attempted to obtain more than 15 % of its shares.

> According to BAA officials, it has been exercised once.
>
> This is when one shareholder attempted to obtain more than 15 % of its shares.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):231

86) Today's rulings could lead to an increase in mergers and acquisitions in the EU as it seeks to dismantle restriction on the free movement of capital within the single European market.

> Today's rulings could lead to an increase in mergers and acquisitions in the EU.
>
> This is as it seeks to dismantle restriction on the free movement of capital within the single European market.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

87) The latest cases followed landmark judgments last June, when the European court of justice struck down a golden share held by the French government.

> The latest cases followed landmark judgments last June.
>
> This was when the European court of justice struck down a golden share held by the French government.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

88) It had signalled its willingness to surrender its golden shares in privatised firms, but the failure to specify how or when persuaded the commission to force its hand.

> It had signalled its willingness to surrender its golden shares in privatised firms.
>
> But the failure to specify how or when persuaded the commission to force its hand.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

89) State troopers in Texas were hunting yesterday for some of the 59 Democratic state legislators who went into hiding to avoid voting on measures they say would aid their Republican opponents.

> State troopers in Texas were hunting yesterday for some of some 59 Democratic state legislators.
>
> These 59 Democratic state legislators went into hiding to avoid voting on measures they say would aid their Republican opponents.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

90) The Democrats, members of the Texas House of Representatives in Austin, executed their secretly coordinated plan late on Sunday, vanishing in order to prevent the legislative body from reaching a quorum.

> The Democrats executed their secretly coordinated plan late on Sunday, vanishing in order to prevent the legislative body from reaching a quorum.
>
> The Democrats were members of the Texas House of Representatives in Austin.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

---

91) They said their aim was to frustrate a bill which would help Republicans by redrawing constituency boundaries, along with other proposals for spending cuts which they argued would harm the poor.

> They said their aim was to frustrate a bill.
>
> This bill would help Republicans by redrawing constituency boundaries, along with other proposals for spending cuts which they argued would harm the poor.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):131

---

92) Tom Craddick, the Republican speaker of the house, ordered troopers to find the Democrats and bring them back, utilising a law which allows members deliberately breaking quorums to be arrested.

> Tom Craddick, the Republican speaker of the house, ordered troopers to find the Democrats and bring them back, utilising a law.
>
> This law allows members deliberately breaking quorums to be arrested.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

93) “We refuse to participate in an inherently unfair process that slams the door of opportunity in the face of Texas voters.”

> “We refuse to participate in an inherently unfair process.
>
> This inherently unfair process slams the door of opportunity in the face of Texas voters.”

Grammaticality (y/n):yyy
Meaning Preservation (0-3):332

94) There are some issues that are important to us, important to all Texans.

> There are some issues.
>
> These issues are important to us, important to all Texans.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):331

95) New Mexico’s attorney-general, Patricia Madrid, wrote in her reply that she did not think the warrants could be executed.

> New Mexico’s attorney-general wrote in her reply that she did not think the warrants could be executed.
>
> Mexico’s attorney-general was Patricia Madrid.

Grammaticality (y/n):yyy
Meaning Preservation (0-3):333

Author Index

Index

References

Richard Anderson and Peter Freebody. 1981. 'Vocabulary Knowledge'. In John Guthrie, editor, *Comprehension and Teaching: Research Reviews*, pages 77–117. International Reading Association, Newark, DE.

Richard Anderson. 1981. 'A proposal to continue a center for the study of reading'. Technical Report 487, University of Illinois, Center for the Study of Reading, Urbana-Champaign.

Breck Baldwin. 1997. 'CogNIAC: High precision coreference with limited knowledge and linguistic resources'. In *Proceedings of the ACL Workshop on Operational Factors in Practical, Robust Anaphora resolution for Unrestricted Texts*, Madrid, Spain. pages 38–45.

Catalina Barbu and Ruslan Mitkov. 2001. 'Evaluation Tool for Rule-based Anaphora Resolution Methods'. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*, Toulouse, France. pages 34–41.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. 'Information Fusion in the Context of Multi-Document Summarization'. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, Maryland, USA. pages 550–557.

Regina Barzilay. 2003. *Information Fusion for Multidocument Summarization: Paraphrasing and Generation.* Ph.D. thesis, Columbia University, New York.

Arendse Bernth. 1998. 'EasyEnglish: Preprocessing for MT'. In *Proceedings of the Second International Workshop on Controlled Language Applications*, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania. pages 30–39.

Susan E. Brennan, Marilyn W. Friedman, and Carl Pollard. 1987. 'A centering approach to pronouns'. In *Proc. of the 25th Annual Meeting of the Association for Computational Linguistics (ACL'87)*, Stanford, California. pages 155–162.

Ted Briscoe and John Carroll. 1993. 'Generalised probabilistic LR parsing for unification-based grammars'. *Computational Linguistics*, 19(1):25–60.

Ted Briscoe and John Carroll. 1995. 'Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels'. In *Proceedings of the ACL/SIGPARSE*

4th International Workshop on Parsing Technologies, Prague / Karlovy Vary, Czech Republic. pages 48–58.

Ted Briscoe, John Carroll, Jonathan Graham, and Ann Copestake. 2002. ‘Relational evaluation schemes’. In *Proceedings of the Beyond PARSEVAL Workshop at the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Gran Canaria. pages 4–8.

Yvonne Canning, John Tait, Jackie Archibald, and Ros Crawley. 2000a. ‘Cohesive Generation of Syntactically Simplified Newspaper Text’. In Petr Sojka, Ivan Kipecek, and Karel Pala, editors, *Text, Speech and Dialogue: Third International Workshop (TSD’00)*, Lecture Notes in Artificial Intelligence 1902, Brno, Czech Republic. Springer-Verlag, pages 145–150.

Yvonne Canning, John Tait, Jackie Archibald, and Ros Crawley. 2000b. ‘Replacing Anaphora for Readers with Acquired Dyslexia’. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC’00)*, Lancaster, U.K.

Yvonne Canning. 2002. *Syntactic simplification of Text*. Ph.D. thesis, University of Sunderland, UK.

David Caplan. 1992. *Language: Structure, Processing, and Disorders*. MIT Press, Cambridge, Massachusetts.

Jean Carletta. 1996. ‘Assessing Agreement on Classification tasks: The Kappa Statistic’. *Computational Linguistics*, 22(2):249–254.

Andrew J. Carlson, Chad M. Cumby, Jeff L. Rosen, and Dan Roth. 1999. ‘The SNoW learning architecture’. Technical report, Tech. Report UIUCDCS-R-99-2101, UIUC Computer Science Department.

Patricia Carpenter, Akira Miyake, and Marcel Adam Just. 1994. ‘Working memory constraints in comprehension: Evidence from individual differences, aphasia, and aging’. In Morton Ann Gernsbacher, editor, *Handbook of psycholinguistics*, pages 1075–1122. Academic Press, New York.

Xavier Carreras and Luís Màrquez. 2001. ‘Boosting Trees for Clause Splitting’. In Walter Daelemans and Rémi Zajac, editors, *Proceedings of the fifth Computational Natural Language Learning Workshop*, Toulouse, France. pages 73–75.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. ‘Practical simplification of English newspaper text to assist aphasic readers’. In *Proceedings of AAAI98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, Madison, Wisconsin. pages 7–10.

John Carroll, Guido Minnen, and Ted Briscoe. 1999a. ‘Corpus Annotation for Parser Evaluation’. In *Proceedings of the EACL’99 workshop on Linguistically Interpreted Corpora (LINC)*, Bergen, Norway. pages 35–41.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999b. ‘Simplifying English text for language impaired readers’. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, Bergen, Norway. pages 269–270.

Raman Chandrasekar and Bangalore Srinivas. 1997. ‘Automatic Induction of Rules for Text Simplification’. *Knowledge-Based Systems*, 10:183–190.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. ‘Motivations and Methods for Text Simplification’. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, Copenhagen, Denmark. pages 1041–1044.

Eugene Charniak. 2000. ‘A Maximum-Entropy-Inspired Parser’. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, Seattle, Washington. pages 132–139.

Stephen Clark and David Weir. 2000. ‘A Class-Based Probabilistic Approach to Structural Disambiguation’. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany. pages 194–200.

Michael Collins and James Brooks. 1995. ‘Prepositional Attachment through a Backed-off Model’. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, Somerset, New Jersey. Association for Computational Linguistics, pages 27–38.

Michael Collins. 1997. ‘Three Generative, Lexicalized Models for Statistical Parsing’. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, Madrid, Spain. pages 16–23.

Fernando Cuetos and Don Mitchell. 1988. ‘Cross-linguistic differences in parsing: Restrictions on the use of the Late Closure strategy in Spanish’. *Cognition*, 30:72–105.

Walter Daelemans and Rémi Zajac, editors. 2001. *Proceedings of the fifth Computational Natural Language Learning Workshop*, Toulouse, France.

Robert Dale and Nicholas Haddock. 1991. ‘Generating Referring Expressions Involving Relations’. In *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics (EACL'91)*, Berlin, Germany. pages 161–166.

Robert Dale. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes.* MIT Press, Cambridge, Massachusetts.

Meredyth Daneman and Patricia Carpenter. 1980. ‘Individual differences in working memory and reading’. *Journal of Verbal Learning and Verbal Behavior*, 19:450–466.

Siobhan Devlin and John Tait. 1998. ‘The use of a psycholinguistic database in the simplification of text for aphasic readers’. In J. Nerbonne, editor, *Linguistic Databases*, pages 161–173. CSLI Publications, Stanford, California.

Siobhan Devlin. 1999. *Simplifying natural language for aphasic readers.* Ph.D. thesis, University of Sunderland, UK.

H. P. Edmundson. 1964. 'Problems in Automatic Abstracting'. *Communications of the ACM, 7(4)*, pages 259–263.

Michael Elhadad and Jacques Robin. 1992. 'Controlling Content Realization with Functional Unification Grammar'. In Robert Dale, Eduard Hovy, Dieter Roesner, and Oliviero Stock, editors, *Proceedings of the Sixth International Workshop on Natural Language Generation*, Lecture Notes in Artificial Intelligence, pages 89–104. Springer Verlag, Trento, Italy.

Claudia Felser, Leah Roberts, Rebecca Gross, and Theodore Marinis. 2003. 'The processing of ambiguous sentences by first and second language learners of English'. *Applied Psycholinguistics*, 24:453–489.

Claudia Felser, Theodore Marinis, and Harald Clahsen. To appear. 'Children's processing of ambiguous sentences: A study of relative clause attachment'. *Language Acquisition*, 11.

Eva Fernandez. 2000. *Bilingual Sentence Processing: Relative Clause Attachment in English and Spanish.* Ph.D. thesis, City University of New York.

Rudolf Flesch. 1951. *How to test readability.* Harper and Brothers, New York.

Rudolf Flesch. 1979. *How to write plain English.* Harper and Brothers, New York.

Niyu Ge, John Hale, and Eugene Charniak. 1998. 'A statistical approach to anaphora resolution'. In *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Canada. pages 161–171.

Laurie Gerber and Eduard Hovy. 1998. 'Improving Machine Translation Quality by Manipulating Sentence Length'. In D. Farwell, Laurie Gerber, and Eduard Hovy, editors, *Machine Translation and the Information Soup: Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA'98)*, Heidelberg, Germany. pages 448–460.

Elizabeth Gilboy, Josep Sopena, Charles Clifton, and Lyn Frazier. 1995. 'Argument structure and preferences in the processing of Spanish and English complex sentences'. *Cognition*, 54:131–167.

John Gilliland. 1972. *Readability.* University of London Press for the United Kingdom Reading Association, London.

Gregory Grefenstette. 1998. 'Producing Intelligent Telegraphic Text Reduction to Provide an Audio Scanning Service for the Blind'. In *Intelligent Text Summarization, AAAI Spring Symposium Series*, Stanford, California. pages 111–117.

Barbara Grosz and Candace Sidner. 1986. ‘Attention, intentions, and the structure of discourse’. *Computational Linguistics*, 12(3):175–204.

Barbara Grosz, Aravind Joshi, and Scott Weinstein. 1995. ‘Centering: A framework for modelling the local coherence of discourse’. *Computational Linguistics*, 21(2):203–226.

Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. 2000. ‘LT TTT - A Flexible Tokenisation Tool’. In *Proceedings of Second International Conference on Language Resources and Evaluation*, Athens,Greece. pages 1147–1154.

Eva Hajicova. 1993. *Issues of sentence structure and discourse patterns*, volume 2 of *Theoretical and computational linguistics*. Charles University, Prague, Czech Republic.

Michael A.K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman Group Ltd, London, U.K.

P Van Hentenryck. 1989. *Constraint satisfaction in logic programming*. MIT Press, Cambridge, Mass.

Jerry R. Hobbs. 1986. ‘Resolving Pronoun References’. In Barbara J. Grosz, Karen Sparck-Jones, and Bonnie L. Webber, editors, *Readings in Natural Language Processing*, pages 339–352. Morgan Kaufmann, Los Altos, California.

Eduard Hovy and Chin-Yew Lin. 1999. ‘Automated text summarization in SUMMARIST’. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*, pages 18–24. MIT Press, Cambridge, Massachusetts.

Aravind Joshi and Bangalore Srinivas. 1994. ‘Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing’. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING’94)*, Kyoto University, Japan. pages 154–160.

Nobuhiro Kaji, Daisuke Kawahara, Sadao Kurohash, and Satoshi Sato. 2002. ‘Verb Paraphrase based on Case Frame Alignment’. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL’02)*, Philadelphia, USA. pages 215–222.

Christopher Kennedy and Branimir Boguraev. 1996. ‘Anaphora in a Wider Context: Tracking Discourse Referents’. In *Proceedings of the European Conference on Artificial Intelligence (ECAI’96)*. John Wiley and Sons, Ltd, London/New York, pages 582–586.

Kevin Knight and Daniel Marcu. 2000. ‘Statistics-Based Summarization — Step One: Sentence Compression’. In *Proceeding of The 17th National Conference of the American Association for Artificial Intelligence (AAAI-2000)*, pages 703–710.

Emiel Krahmer and Mariët Theune. 2002. ‘Efficient context-sensitive generation of referring expressions’. In Kees van Deemter and Rodger Kibble, editors, *Information Sharing: Givenness and Newness in Language Processing*, pages 223–264. CSLI Publications, Stanford,California.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. 'Graph-Based Generation of Referring Expressions'. *Computational Linguistics*, 29(1):53–72.

Shalom Lappin and Herbert Leass. 1994. 'An Algorithm for Pronominal Anaphora Resolution'. *Computational Linguistics*, 20(4):535–561.

Hans Peter Luhn. 1958. 'The Automatic Creation of Literature Abstracts'. *IBM Journal of Research and Development*, 2(2):159–165.

E.A. Lunzer and W.K. Gardner, editors. 1979. *The effective use of reading. School council project report*. Heinemann, London, U.K.

Inderjeet Mani and Mark Maybury. 1999. 'Introduction'. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*, pages ix–xv. MIT Press, Cambridge, Massachusetts.

William C. Mann and Sandra A. Thompson. 1988. 'Rhetorical Structure Theory: Towards a functional theory of text organization'. *Text*, 8(3):243–281.

Daniel Marcu. 1997. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. Ph.D. thesis, University of Toronto, Canada.

Daniel Marcu. 2000. 'The rhetorical parsing of unrestricted texts: A surface-based approach'. *Computational Linguistics*, 26(3):395–448.

J. Mason and J. Kendall. 1979. 'Facilitating reading comprehension through text structure manipulation'. *Alberta Journal of Medical Psychology*, 24:68–76.

Kathleen McKeown, Jacques Robin, and Karen Kukich. 1995. 'Generating concise natural language summaries'. *Information Processing and Management*, 31(5):703–733.

Andrei Mikheev, Claire Grover, and Marc Moens. 1999. 'XML tools and architecture for Named Entity recognition'. *Journal of Markup Languages: Theory and Practice*, 1(3):89–113.

Andrei Mikheev. 1997. 'Automatic Rule Induction for Unknown Word Guessing'. *Computational Linguistics*, 23(3):405–423.

Andrei Mikheev. 1998. 'Feature lattices for maximum entropy modelling'. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)*, Montreal, Canada. pages 845–848.

George A. Miller, Richard Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine Miller. 1993. 'Five Papers on WordNet'. Technical report, Princeton University, Princeton, N.J.

Ruslan Mitkov. 1998. 'Robust Pronoun Resolution with Limited Knowledge'. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)*, Montreal, Canada. pages 869–875.

Geoffrey Nunberg. 1990. *The Linguistics of Punctuation.* Number 18 in CSLI Lecture Notes. CSLI Publications, Stanford, California.

S. Parr. 1993. *Aphasia and Literacy.* Ph.D. thesis, University of Central England.

Joseph Pollock and Antonio Zamora. 1975. 'Automatic Abstracting Research at Chemicals Abstracts Service'. *Journal of Chemical Information and Computer Sciences*, 15(4):226–232.

Richard Power. 2000. 'Planning texts by constraint satisfaction'. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany. pages 642–648.

Judita Preiss. 2002. 'Choosing a Parser for Anaphora Resolution'. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC'02)*, Lisbon, Portugal. pages 175–180.

Stephen P. Quigley and Peter V. Paul. 1984. *Language and Deafness.* College-Hill Press, San Diego, California.

Stephen P. Quigley, Des Power, and M.W. Steinkamp. 1977. 'The language structure of deaf children'. *The Volta Review*, 79(2):73–84.

Philip Quinlan. 1992. *The Oxford Psycholinguistic Database.* Oxford University Press, U.K.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language.* Longman, London and New York.

G. J. Rath, A Resnick, and T. R. Savage. 1961. 'The Formation of Extracts by the Selection of Sentences'. *American Documentation*, pages 139–141.

Adwait Ratnaparkhi. 1998. 'Statistical Models for Unsupervised Prepositional Phrase Attachement'. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)*, Montreal, Canada. pages 1079–1085.

Ehud Reiter and Robert Dale. 1992. 'A fast algorithm for the generation of referring expressions'. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, Nantes, France. pages 232–238.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems.* Cambridge University Press.

Ehud Reiter, R. Robertson, and Liesl Osman. 1999. 'Types of Knowledge Required to Personalise Smoking Cessation Letters'. In Werner Horn et al, editor, *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, Berlin, Germany. Springer-Verlag, pages 389–399.

Ehud Reiter, R Robertson, and Liesl Osman. 2003. ‘Lessons from a Failure: Generating Tailored Smoking Cessation Letters’. *Artificial Intelligence*, 144:41–58.

Ehud Reiter. 1990. ‘The Computational Complexity of Avoiding Conversational Implicatures’. In *Proceedings of the 28th Annual Meeting of Association for Computational Linguistics (ACL’90)*, Pittsburgh, Pennsylvania. pages 97–104.

Stefan Riezler, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. ‘Statistical Sentence Condensation using Ambiguity Packing and Stochastic Disambiguation Methods for Lexical-Functional Grammar’. In *Proceedings of the 3rd Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL’03)*, Edmonton, Canada.

Nancy Lee Robbins and C.W. Hatcher. 1981. ‘The effects of syntax on the reading comprehension of hearing-impaired children’. *The Volta Review : journal of the Alexander Graham Bell Association for the Deaf*, 83:105–115.

James Shaw. 1998. ‘Clause Aggregation Using Linguistic Knowledge’. In *Proceedings of the 9th International Workshop on Natural Language Generation (INLG’98)*, Niagara-on-the-Lake, Canada. pages 138–147.

Cynthia Shewan and Gerald Canter. 1971. ‘Effects of vocabulary, syntax and sentence length on auditory comprehension in aphasic patients’. *Cortex*, 7:209–226.

Advaith Siddharthan and Ann Copestake. 2002. ‘Generating Anaphora for Simplifying Text’. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC’02)*, Lisbon, Portugal. pages 199–204.

Advaith Siddharthan and Ann Copestake. 2004. ‘Generating Referring Expressions in Open Domains’. In *Proceedings of the 42th Meeting of the Association for Computational Linguistics Annual Conference (ACL 2004)*, Barcelona, Spain. pages 408–415.

Advaith Siddharthan. 2002a. ‘An Architecture for a Text Simplification System’. In *Proceedings of the Language Engineering Conference (LEC’02)*, Hyderabad, India. pages 64–71.

Advaith Siddharthan. 2002b. ‘Resolving Attachment and Clause Boundary Ambiguities for Simplifying Relative Clause Constructs’. In *Proceedings of the Student Workshop, 40th Meeting of the Association for Computational Linguistics (ACL’02)*, Philadelphia, USA. pages 60–65.

Advaith Siddharthan. 2003a. ‘Preserving Discourse Structure when Simplifying Text’. In *Proceedings of the European Natural Language Generation Workshop (ENLG), 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL’03)*, Budapest, Hungary. pages 103–110.

Advaith Siddharthan. 2003b. ‘Resolving Pronouns Robustly: Plumbing the Depths of Shallowness’. In *Proceedings of the Workshop on Computational Treatments of*

Anaphora, 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03), Budapest, Hungary. pages 7–14.

Advaith Siddharthan. To appear. 'Syntactic Simplification and Text Cohesion'. *Journal of Language and Computation.*

Sidney Siegel and N. John Jr. Castellan. 1988. *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill, 2nd edition.

George Snedecor and William Cochran. 1989. *Statistical Methods.* Iowa State University Press, Ames, IA.

Karen Sparck Jones and Julia Galliers. 1996. *Evaluating Natural Language Systems.* Springer Verlag.

Karen Sparck Jones. 1999. 'Automatic summarising: factors and directions'. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*, pages 1–12. MIT Press, Cambridge, Massachusetts.

Bangalore Srinivas. 1997. *Complexity of Lexical Descriptions and its Relevance to Partial Parsing.* Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Michael Strube. 1998. 'Never Look Back: An Alternative to Centering'. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING'98)*, pages 1251–1257.

Joel Tetreault. 1999. 'Analysis of syntax-based pronoun resolution methods'. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99) Student Session*, Maryland, USA. pages 602–605.

Henry Thompson. 1977. 'Strategy and tactics: A model for language production'. In W. Beach, S. Fox, and S. Philosoph, editors, *Papers from the 13th Regional Meeting of the Chicago Linguistics Society*, pages 651–668.

Sandra Williams, Ehud Reiter, and Liesl Osman. 2003. 'Experiments with Discourse-Level Choices and Readability'. In *Proceedings of the European Natural Language Generation Workshop (ENLG), 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary. pages 127–134.

Richard Wojcik and James Hoard. 1996. 'Controlled Languages in Industry'. In Ronald Cole, editor, *Survey of the state of the art in Human Language Technology*, pages 274–276. http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Richard Wojcik, James Hoard, and Kim Holzhauser. 1990. 'The Boeing Simplified English Checker'. In *Proceedings of the International Conference in Human Machine Interaction and Artificial Intelligence in Aeronautics and Space*, Centre d'Etude et de Recherche de Toulouse, France. pages 43–57.