Syntactic Simplification and Text Cohesion

ADVAITH SIDDHARTHAN

Department of Computer Science, Columbia University[1],
E-mail: as372@cs.columbia.edu

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 paper, we 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. We 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 and present the results of an evaluation of our syntactic simplification system.

KEYWORDS: *Syntactic simplification, discourse structure, text cohesion, anaphoric structure, sentence ordering, cue-word selection, determiner choice*

[1]The research reported in this paper was carried out at the University of Cambridge, U.K.

 © Hermes Science Publishing LTD

1 Introduction

Syntactic simplification is the process of reducing the grammatical complexity of a text, while retaining its information content and meaning. Syntactic simplification involves replacing particular syntactic constructs (like relative clauses, apposition and conjunction) in sentences in order to make the text either easier to read for some target group (people with aphasia, deafness or low reading ages have trouble understanding long sentences and complex grammar [3, 14, 16]) or easier to process by some program (like parsers or machine translation systems). Syntactic simplification was originally proposed as a preprocessing step for parsers [5, 6] as the reduction in sentence length was expected to improve parser throughput. Later, the PSET (Practical Simplification of English Text) project [4, 7] used text simplification to try and make newspaper text accessible to aphasics.

A broad coverage text simplification system is expected to be useful to people with language disabilities like aphasia, adults learning English (by aiding the construction of texts that are of the desired linguistic complexity, while being relevant to adults), non-native English speakers surfing a predominantly English internet and users of limited channel devices (software that displays text in short sentences that fit on small screens could improve the usability of these devices). Further, syntactic simplification has potential uses as a preprocessing tool for improving the performance of other applications like parsing and machine translation (where performance deteriorates rapidly with sentence length) and text summarisation systems based on sentence extraction (as simplified sentences contain smaller units of information).

Previous research on text simplification has not considered the discourse level issues that arise from applying syntactic transforms at the sentence level. Chandrasekar et al. [6], for example, use an architecture with two stages—*analysis* and *transformation*. There are various discourse level issues that arise when carrying out sentence-level syntactic restructuring. Not considering these discourse implications could result in the simplified text losing coherence, or even changing the intended meaning, in either case, making the text harder to comprehend. For example, consider the sentence:

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

The clause, *but he isn't sure it should be strictly regulated* is rhetorically linked to 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.

conjunctive cohesion (rhetorical cohesion achieved through a conjunction) is adversely affected as the final sentence is erroneously linked 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*. It appears on first sight that the issues of anaphoric and conjunctive cohesion are interlinked, as the situation can be partially remedied by replacing the pronoun *it* with its

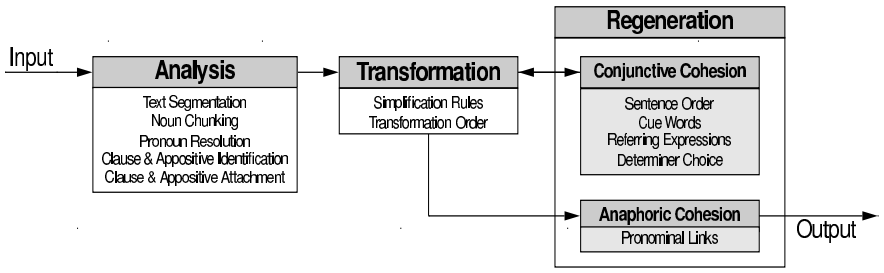


FIG. 1. An architecture for syntactic simplification

antecedent *program trading*. One contribution of this paper is the demonstration that the issues of conjunctive and anaphoric cohesion can be treated independently, with anaphoric cohesion handled as a post-process.

In §3, we 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, for this example, generating:

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

Our approach to preserving conjunctive cohesion can still result in broken 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. Our theory of how to detect and correct these breaks in anaphoric cohesion is detailed in section 5. Before describing the discourse level effects of syntactic simplification, we overview the architecture of our system in §2.

2 Overview of the System

We divide the simplification task into three stages— *analysis*, *transformation* and *regeneration*, as shown in the block diagram in figure 1. The text is analysed in the

analysis stage and then passed on to the *transformation* stage. The transformation stage applies rules for syntactic simplification and calls the *regeneration* stage as a subroutine to address issues of conjunctive cohesion. When no further simplification is possible, the transformation stage pipes the simplified text to the regeneration stage, which then addresses issues of anaphoric cohesion as a post-process.

2.1 Analysis Stage

The output specification of our analysis module is shown below:

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.

We use the LT Text Tokenization Toolkit [10] 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. We then mark up syntactic structures that can be simplified in each sentence. This mark-up has two components— clause/appositive identification and clause/appositive attachment. We resolve attachment and boundary ambiguities using techniques based on local context and lexical knowledge resources such as WordNet [13]. 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 when it needs to replace a pronoun with a referring expression in order to preserve anaphoric cohesion (as mentioned in §1 and expanded on in §5). Details of our implementations of the analysis module can be found in [18], [19] and [20].

2.2 Transformation and Regeneration Stages

The transformation stage receives as input the output of the analysis stage. Our implementation uses XML tags to represent the specification described above. For ease of presentation, we display the markup differently in our examples.

The primary function of the transformation module is to apply syntactic-simplification rules to the analysed text. We use a set of hand-crafted rules like the following:

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

This rule states that if in a sentence, a relative clause *RELPR Y* attaches to a noun phrase W, then we can extract *W Y* into a new sentence. We use superscript *#n* to indicate attachment to the noun phrase with superscript *n*. Each rule also specifies the relation between the two simplified sentences in the form of a triplet *(a, R, b)*, where the sentence *a* is the nucleus of the relation *R* and *b* the satellite. The relations that we assign are partly based on rhetorical relations [12], and are elaborated on in §3.2. Rhetorical Structure Theory (RST) postulates that for most relations, the involved clauses have a nucleus-satellite relationship. In our approach, for the case of conjunction, the nucleus and satellite are determined from the cue-word and its position in the original sentence (whether it precedes the first or second clause). In the case of embedding, the embedded construct is always the satellite. We use seven rules in total, three for conjunction and two each for relative clauses and apposition (details in [20]). Within a sentence, these rules are applied sequentially in a top-down manner, as described below.

The transformation stage implements an algorithm that recursively simplifies the analysed text. The analysed sentences (output of the analysis stage) are represented as a *stack* with the first sentence at the top. This stack is then transformed recursively as follows. At each transformation step, the first sentence in the stack is popped. In the base case, when the popped sentence contains no simplifiable construct, it is added to the end of an output queue. In the recursive case, a transformation rule is applied to the popped sentence and the two resultant simplified sentences are sent to the regeneration stage, which addresses issues of conjunctive cohesion (cf. §3). These two (regenerated) sentences are then pushed onto the top of the transformation stack in the order specified by the regeneration stage. When there are multiple constructs that can be simplified within a sentence, the simplification is carried out in a top-down manner (we discuss this further in §3.1.3).

When the transformation stack is empty, the simplified text is contained in the output queue. The transformation stage then invokes the regeneration stage on the output queue for fixing pronominal links (cf. §5), before outputting the simplified text.

As described above, the regeneration stage has two modules. The module for handling conjunctive cohesion is called (repeatedly) by the transformation stage as a subroutine. The module that handles pronominal links (anaphoric cohesion) is the third stage of the pipeline and is invoked at the end of the transformation stage.

3 Preserving Conjunctive Cohesion

Having overviewed the system, we now describe how the issues of sentence ordering, cue-word selection and determiner choice can be resolved in a manner that preserves conjunctive cohesion and connectedness. We then address anaphoric cohesion in §5.

3.1 Sentence Ordering

3.1.1 Constraint Based Text Planning

We formulate the sentence ordering task as a constraint satisfaction problem. The constraint satisfaction approach was first used in planning text structure by the ICONOCLAST [15] project. A key issue in natural language generation is that of realising a discourse structure, represented as a Rhetorical Structure Theory [12] 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 [15] described how this task could be formalised as a constraint satisfaction problem. A constraint satisfaction problem [11] 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 $X_6 = 0$).

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 between multiple potential 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.

In ICONOCLAST, the rules for 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 [15] included a set of soft stylistic constraints; for example, that single sentence paragraphs are undesirable.

Power [15] 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.

For our application, we do not require full blown text planning, and only need to order sentences. This only requires us to consider text-sentences (TEXT-LEVEL=2). Further, we do not consider typographic features like indentation and itemised lists. Power [15] only considered relations that can be realised by a conjunction, and demonstrated that text planning can be formulated as a CSP by exploiting interactions between the choices of cue-words and the potential orderings of textual units. We extend this by offering a treatment of relative clauses and apposition. Further, we use 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).

3.1.2 Local vs Global Sentence Ordering

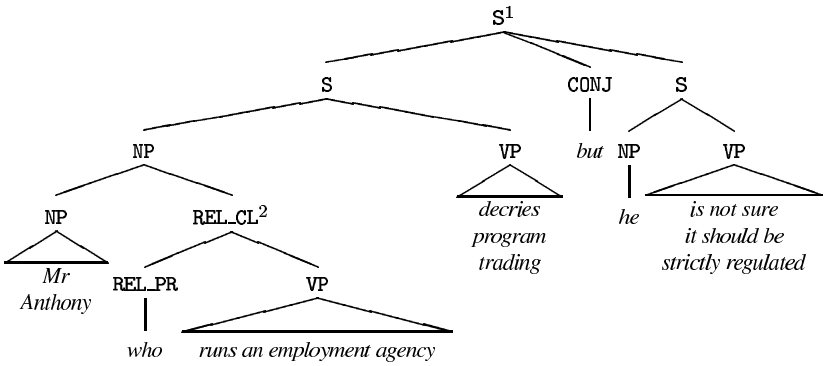
We can simplify the sentence ordering problem further by making ordering decisions locally rather than globally when there is more than one construct that can be simplified in a sentence. Consider the sentence:

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

Our transformation module applies simplification rules in a top-down manner (the conjunction (construct 1) is simplified before the relative clause (construct 2) in figure 2), and it is possible to resolve sentence-ordering constraints locally, rather than globally. Global sentence ordering would involve deciding the relative order of the three simplified 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 sen-



[1 Mr. Anthony, [2 who runs an employment agency 2], decries program trading1][1, but he isn't sure it should be strictly regulated1].

FIG. 2: Top down transform application. The tree is shown for illustration purposes only, our analysis module only marks up embedded clauses as shown below the tree.

tences generated by the first transform (that simplifies the *but* clause):

(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):

(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. It is also desirable that clauses that were adjacent in the original text remain adjacent in the simplified text. The local ordering approach ensures this, and is equivalent to a global approach with the preservation of adjacency encoded as a hard constraint. We now describe our local approach to sentence ordering, where a decision is made at every transform application on the optimal order of the two generated simplified sentences.

3.1.3 Local Sentence Ordering and Recursive Transformation

In our formalisation of local sentence ordering as a constraint satisfaction problem, 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 the discourse relation between

the two sentences, as well as from considerations of referential cohesion and connectedness. Constraints can get passed down the recursion during recursive transform application, as elaborated on later in this section. The sentence-ordering algorithm is called by the transformation stage after each application of a simplification rule and receives two inputs:

1. A triplet (a, R, b) of the simplified sentences a and b and the relation R between them. The relations that we assign are partly based on rhetorical relations, and are elaborated on in §3.2.
2. A set C of inherited constraints (introduced by previously applied transforms) on sentence order.

The algorithm forms new constraints from the relation R, 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 passed down the recursion in the transformation stage and made available to future calls to the sentence ordering algorithm.

We now describe the constraints that different relations R add to the sets C, C_a and C_b. With the exception of three (*cause*, *elaboration* and *identification*[2]), every 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 relation between the two simplified sentences. The other two constraints arise because this 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:

[2]Following the approach in [17], we use the *elaboration* relation to relate non-restrictive relative clauses and appositives to the main clause. However, we postulate an *identification* relation for restrictive relative clauses, thus deviating from the RST treatment in [17], where all embedded clauses are hypothesised to realize *elaboration*. To motivate our approach, consider the restrictive relative clause in *the man who had brought it in for an estimate returned to collect it*. The relation between the relative clause and the main clause is not strictly *elaboration*; rather, its purpose is referential - to identify one man from the set of all men. This distinction is important to us and we thus postulate an *identification* relation that is referential rather than conjunctive (thus diverging from RST, which does not attempt to model referential relations).

(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 (*R*=*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 *R*=*cause*. 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 we transform the *cause* relation into a *result* relation (cf. §3.2 for the rationale) and the *result* clause has to be second; for example, we 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 *R*=*elaboration*, the constraints introduced are *soft*. Elaboration clauses 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 the *elaboration* relation 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, 3.1(b') is only marginally (if at all) less acceptable than 3.1(b) below[3]:

(3.1) 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 *R=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 *elaboration* or *identification* relations can be violated either to enforce a hard constraint or to improve text connectedness.

3.1.4 The Algorithm for Sentence-Ordering

We now present our algorithm for deciding sentence order. Algorithm 3.1 receives a constraint set C, the simplified sentences a and b and the relation R 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 relation R 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.

[3] In general, making a new sentence out of an embedded clause does affect discourse structure. In this example, both simplified versions elevate the importance of Boone teaching art at Earlham College. However, many classes of struggling readers have fundamental problems with comprehending relative clauses; for example, the sentence *The boy who hit the girl ran home*, is likely to be interpreted as *the girl ran home* by the deaf [16]. These readers are likely to benefit from syntactic simplification, despite such discourse level concerns.

ALGORITHM 3.1 (Deciding Sentence Order Locally)

Order-Sentences((a,R,b),C)

1. Initialise C_a and C_b to the constraints in C
2. Process R 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

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:

(3.2) 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 [8, 9]. 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.

While centering theory can be used to justify our 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 3.2 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].

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:

(3.3) 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.

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:

(3.3) 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:

(3.3) 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.

Conjunctions	Relation	Cue-Words
although, though, whereas, but, however	(a, Concession, b)	but
or, or else	(a, Anti-Conditional, b)	otherwise
because	(a, Cause, b)→(b, Result, a)	so
and	(a, And, b)	and
X	(a, X, b)	this AUX X

TABLE 1. Relations triggered by conjunctions, and regenerated cue-words

Again, centering theory can be used to reason that 3.3(b) is preferable to both 3.3(b') and 3.3(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).

3.2 Cue-Word Selection

To preserve the relation between conjoined clauses that have been simplified into separate sentences, it is necessary to introduce new cue-words in the simplified text. In our architecture, cue-word selection is resolved using an input from the transformation stage of the form *(a, R, b)*, where *R* is the relation connecting the two simplified sentences *a* and *b*. The set of relations that we use is motivated by RST, but has been extended to suit the requirements of our specific task.

Table 1 shows the relation associated with each subordinating conjunction that we simplify, and the regenerated cue-word. The final row is a default that arises because RST is in some cases not suited for our application. For example, RST provides the rhetorical relation *circumstance* where the satellite clause provides an interpretive context of situation or time. However, we need to be able to distinguish between *when*, *before* and *after* clauses, all of which have the *circumstance* relation with their nucleus. We therefore introduce our own relations *(a, when, b)*, *(a, before, b)* and *(a, after, b)*. 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 *cause* or a *circumstance* relation. As our analysis module does not disambiguate rhetorical relations, we define our own relations *(a, as, b)* and *(a, since, b)* that capture the underspecified relation. The *(a, and, b)* relation is similarly underspecified. Our application allows us to transfer ambiguity in discourse cues from the input to the output, and we therefore do not need to perform rhetorical analysis of the input.

We have a choice of cue-words available for signalling some relations. Williams et al. [22] 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. also reported that the presence of punctuation along with the cue-word resulted in faster reading times. We therefore

restrict ourselves to using simple cue-words like *so* for the *result* relation and *but* for the *concession* relation and also include punctuation wherever possible.

All the cue-words that we introduce are positioned at the beginning of the second sentence. Every *concession* relation is realised by the cue-word *but*; for example:

(3.4) 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.

We convert the *cause* relation to a *result* relation in order to use the simple cue-word *so*. This also results in reversing the original clause order (refer to §3.1 on sentence-ordering). An example is:

(3.5) 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.

For each of these relations `X`, we 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:

(3.6) 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.

(3.7) 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.

(3.8) 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.

3.3 Determiner Choice

Simplifying relative clauses and appositives requires in the duplication of a noun phrase. We need to use a referring expression the second time, a topic we discuss in §4. We also need to decide on what determiners to use. This decision depends on the 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 *R=elaboration*. The only constraint here is that there should be a definite determiner in the referring expression. We use *this* or *these* depending on the whether the noun phrase is singular or plural; for example, in:

(3.9) 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 relation is that of *identification* - identifying a member (or some members) from a larger set. To preserve this, we 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:

(3.10) 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:

(3.11) 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.

We do not 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 pronoun present (for example, in *his latest book*).

3.4 Evaluation

Evaluating issues of conjunctive cohesion is non-trivial. One 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

(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):

FIG. 3. An example from the data-set for the evaluation of correctness

syntactic simplification (Chandrasekar et al. [5] or the PSET project [1]) considered issues of conjunctive cohesion, there is no precedent for evaluation that we can follow.

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, we conducted a human evaluation using three native-English speakers with a background in computational linguistics as subjects. We presented the three subjects with 95 examples. Each example consisted of a sentence from a corpus of 15 Guardian news reports that was simplified by our program, the corresponding simplified sentences that were generated and boxes for scoring grammaticality and semantic parity. An example from the evaluation is presented in figure 3.

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:

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 reason for using a single scale for both meaning preservation and coherence is that the two are related. Indeed, in a pilot evaluation, judges found it difficult to distinguish between

Judges	Grammatical (G)	Meaning Preserving (MP)	G and MP
Unanimous	80.0%	85.3%	67%
Majority vote	94.7%	94.7%	88.7%

TABLE 2: Percentage of examples that are judged to be grammatical and meaning-preserving

extreme incoherence and meaning change. Meaning change can be considered a particularly dangerous form of incoherence, because not only is the intended meaning inaccessible to the reader, but the reader is misled into an incorrect interpretation.

Grammaticality

The evaluation results for grammaticality and meaning preservation are summarised in table 2. 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:

> 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:

> 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.

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 case 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 by our analysis module; for example, 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). These invariably arose from errors in the analysis module, in either clause identification or clause attachment.

As table 2 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.

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 our system preserves coherence despite the lack of agreement between judges, we 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.

We 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. However, there are various problems that arise when trying to construct these bounds.

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. When asked to revise the simplified text to improve it, there were examples where judges stated that they could not rewrite the simplified sentences in a manner that preserved the subtleties of the original. Further, 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 (an example follows later in this section). 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. Therefore, while we can speculate that the upper bound might be less than 3.00, we cannot quantify what that bound might be.

In order to find a lower bound, we would have had to ask the experimental subjects to judge the output of a baseline algorithm; for example, one that used no cue words and ordered the simplified sentences in accordance with the original clause order. As the evaluation described above was both labour and time intensive, it was not feasible to ask the judges to perform another evaluation for a baseline algorithm. As a compromise, we tried to assess the utility of only our sentence ordering algorithm, by extrapolating from the results of the original evaluation. There were 17 examples (18%) where our 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, our data set did not contain any instance of a *because* clause, which is the only instance of conjunction where our algorithm reverses clause order. On the 17 examples where our 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.

To try and pin the errors on particular algorithms in our simplification system, we 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 our 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 our 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 another 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 our 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.

News Source	Flesch Reading Ease	Flesch Reading Age	Av. Sent. 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 3: Flesch readability scores and average sentence lengths before and after syntactic simplification (shown as *original* —→ *simplified*)

Readability

Table 3 compares a few objective readability measures for news reports from different sources (we used 15 reports per source) before and after simplification by our program. Our program appears to reduce average sentence lengths to around 15 words across 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). Of the newspapers surveyed, the Wall Street Journal was judged the least readable. This was largely due to the abnormally high number of proper names (companies and people), which increased the number of syllables per word.

4 Generating Referring Expressions

The previous section dealt with the issue of preserving conjunctive cohesion. We now turn our attention to issues of anaphoric cohesion. In this section, we consider the use of referring expressions as an anaphoric device. Then, in §5, we consider issues relating to pronominalisation in rewritten text.

When splitting a sentence into two by dis-embedding a relative clause, we 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. We need to generate a referring expression for the second sentence. Referring-expression generation is an important aspect of natural-language generation, but existing approaches are unsuited for open domains. We have elsewhere [21] described a lexicalised incremental approach that can generate referring expressions in open domains. Our approach does not rely on the availability of an attribute classification scheme and uses WordNet [13] antonym and synonym lists instead. It is also, as far as we know, the first algorithm that allows for the incremental incorporation of relations in a referring expression. Due to space constraints, we cannot describe our referring expression generator here. We make do with emphasising

that open-domain referring expression generation is important to text simplification—including too much information in a referring expression makes the text stilted and can convey unwanted and possibly wrong conversational implicatures, while including too little information can result in ambiguity. Consider the sentence:

> Also contributing to the firmness in copper, the analyst noted, was *a report by Chicago purchasing agents*, which precedes *the full purchasing agents report* that is due out today and gives an indication of what the full report might hold.

Our algorithm simplifies the above to:

> Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. A full purchasing agents report is due out today. The Chicago report precedes the full report and gives an indication of what the full report might hold.

Contrast the above with the stiltedness of generating full references:

> Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. A full purchasing agents report is due out today. The report by Chicago purchasing agents precedes the full purchasing agents report and gives an indication of what the full report might hold.

or the ambiguity that results from generating only head nouns:

> Also contributing to the firmness in copper, the analyst noted, was a report by Chicago purchasing agents. A full purchasing agents report is due out today. The report precedes the report and gives an indication of what the full report might hold.

5 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 we cannot ensure that the attentional state (the model of the reader's focus of attention) at every point in the discourse remains the same before and after simplification, we have to consider the possibility of broken pronominal links. In this section, we 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 correct pronominal links that have been broken by the rewriting operations.

5.1 Pronominalisation, Cohesion and Coherence

Our interest in pronominalisation stems from our desire to ensure that the simplified text retains anaphoric cohesion. In particular, our objective is different from that of [2] in the PSET project, who aimed to replace every 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, we only aim to correct 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. Our 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, our 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.

We use a salience function to model attentional state. For the rest of this paper, we use the term *salience list* (*S*) to refer to a list of discourse entities that have been sorted according to the salience function used by our anaphora resolution program [19]. As an illustration, consider example 5.1 below:

(5.1) 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:

(5.1) 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:

S_{orig} = {Mr Blunkett, the security breach, a comedian, Prince William's 21st birthday party, Prince William, Windsor Castle, ... }

S_{simp} = {this breach, a comedian, Prince William's 21st birthday party, Prince William, Windsor Castle, Mr Blunkett, ... }

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. We therefore want to replace as few pronouns as possible. We do this by relaxing our original objective of preserving pronominal cohesion to only preserving pronominal coherence. We now run our pronoun-resolution algorithm on the simplified text and deem pronominal coherence to be lost if the pronoun-resolution algorithm returns different antecedents for a pronoun in the original and simplified texts. For the highlighted *he* in example 5.1, our 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 5.2 below:

(5.2) 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:

(5.2) 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[a] then took **him** quite properly to a police point.

[a]This example exposes a limitation of our referring expression generator, in that it does not recognise multiword expressions like *member of staff*.

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, our pronoun resolution algorithm returns *Mr Barschak* as the antecedent of *him* in both texts (as binding constraints 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, we can relax our objective further, to only preserve *local* pronominal coherence. Our pronoun-resolution algorithm [19] is significantly more accurate when finding the immediate antecedent than when finding the absolute antecedent. We therefore do not replace a pronoun if the immediate antecedent is the same in both texts. In example 5.2 above, the immediate antecedent of *him* is *he* in both texts. We assume that this is sufficient to preserve local coherence. Algorithm 5.1 formalises our approach to detecting and correcting broken anaphoric links.

ALGORITHM 5.1 (Detecting and correcting pronominal links)

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

Our theory only aims to correct broken anaphoric links in a text and does not attempt to replace the existing anaphoric structure with a new one. In particular, algorithm 5.1 can only replace pronouns in a text and cannot, in any situation, introduce pronouns. Consider:

(5.3) 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 5.3(c) is transformed to 5.3(c') below:

(5.3) 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.

Our pronoun-resolution algorithm resolves *he* to *David Beckham* in the original text, but incorrectly to *Elvis Presley* in the simplified text. Our anaphoric post-processor therefore replaces *he* with *David Beckham* to give:

(5.3) 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 5.3(c''), it might be desirable to pronominalise the subject, to give:

(5.3) 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.

Algorithm	No. Replaced	No. of Errors	Accuracy
Cohesion Preserving	68	19	.72
Coherence Preserving	17	5	.70
Local-Coherence Preserving	11	3	.73

TABLE 4. Precision results for pronoun replacement

We 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"").

5.2 Attentional States and the Reader

As we have mentioned before, the correct resolution of pronouns by readers depends on their maintaining an accurate focus of attention. In our approach to correcting broken pronominal links, we 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. We 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 we can adapt our anaphoric post-processor to different readers, simply by changing our pronoun-resolution algorithm.

In algorithm 5.1, we 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 we need to do is use a different pronoun resolution algorithm on the simplified text. We discuss two possibilities below. Note that we still need to use the best available pronoun resolution algorithm on the original text to locate the correct antecedent.

If we use our pronoun-resolution algorithm without the agreement and syntax filters, our 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.

If we do away with the pronoun-resolution algorithm completely, our approach reduces to one in which all pronouns are replaced. This is similar to the anaphoric simplification carried out in the PSET project [2].

5.3 Evaluation

We now evaluate three different approaches to pronoun-replacement that we have described— cohesion preserving, coherence preserving and local-coherence preserv-

ing. These approaches are implemented using algorithm 5.1 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 4 shows the results of these approaches on our corpus of 15 Guardian news reports. We do not attempt pronoun replacement for occurrences of the pronoun *it*. This is because 85% of *it*s in Guardian news reports are not anaphoric [20].

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 our 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. Our approach resulted in the introduction of only three errors.

6 Conclusions and Future Work

In this paper, we 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. We have formalised the interactions between syntax and discourse during the text simplification process and shown that in order to preserve conjunctive cohesion and anaphoric coherence, it is necessary to model both intentional structure and attentional state. Our 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 our objective of preserving anaphoric coherence. Consider again:

(6.1) 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 6.1(c), the attentional state is:

S = {Dr. Knudson, children, damaged copy, parent, eye cancer, ...}

When we split the last sentence, we have the choice of ordering the simplified sentences as either of 6.1(c') or 6.1(c"):

(6.1) 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 6.1(c) is replaced by 6.1(c'), the attentional state is:

S = {Dr. Knudson, children, damaged copy, parent, eye cancer, ...}

When sentence 6.1(c) is replaced by 6.1(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 disembedded clause needs to be the second sentence, as in 6.1(c"). This ordering also leads to a more connected text, as described in §3.1. However, this ordering significantly disrupts the attentional state that is more or less preserved by the ordering 6.1(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.

Our theory allows us to handle issues of conjunctive and anaphoric cohesion separately. It allows us to select the ordering that preserves conjunctive cohesion (6.1(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 6.1(c) is:

(6.1) 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*. We 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:

(6.1) 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 6.1(d) (sentence 6.1(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, ...}

We feel that our 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.

Future work would include implementing a lexical simplification module and performing a comprehension-based evaluation on end users with low reading ages. In addition to extending and evaluating our text simplification system, we are also interested in researching the use of text simplification as a preprocessor for other NLP tasks; in particular, parsing, translation and summarisation.

Acknowledgements

Thanks are due to thank Ann Copestake, Simone Teufel and John Carroll for many fruitful discussions and comments on the research described here and to two anonymous reviewers for useful feedback on preliminary versions of this paper.

References

[1] Yvonne Canning. *Syntactic simplification of Text*. PhD thesis, University of Sunderland, UK., 2002.

[2] Yvonne Canning, John Tait, Jackie Archibald, and Ros Crawley. 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, pages 145–150, Brno, Czech Republic, 2000. Springer-Verlag.

[3] David Caplan. *Language: Structure, Processing, and Disorders*. MIT Press, Cambridge, Massachusetts, 1992.

[4] John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 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)*, pages 269–270, Bergen, Norway, 1999.

[5] Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. Motivations and methods for text simplification. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING '96)*, pages 1041–1044, Copenhagen, Denmark, 1996.

[6] Raman Chandrasekar and Bangalore Srinivas. Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10:183–190, 1997.

[7] Siobhan Devlin. *Simplifying natural language for aphasic readers*. PhD thesis, University of Sunderland, UK., 1999.

[8] Barbara Grosz, Aravind Joshi, and Scott Weinstein. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226, 1995.

[9] Barbara Grosz and Candace Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[10] Grover, Claire and Matheson, Colin and Mikheev, Andrei and Moens, Marc. LT TTT - A Flexible Tokenisation Tool. *Proceedings of Second International Conference on Language Resources and Evaluation*, pages 1147–1154, Athens, Greece, 2000

[11] P Van Hentenryck. *Constraint satisfaction in logic programming*. MIT Press, Cambridge, Mass, 1989.

[12] William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Towards a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[13] George A. Miller, Richard Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine Miller. Five Papers on WordNet. Technical report, Princeton University, Princeton, N.J., 1993.

[14] S. Parr. *Aphasia and Literacy*. PhD thesis, University of Central England, 1993.

[15] Richard Power. Planning texts by constraint satisfaction. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pages 642–648, Saarbrücken, Germany, 2000.

[16] Stephen P. Quigley and Peter V. Paul. *Language and Deafness*. College-Hill Press, San Diego, California., 1984.

[17] Donia Scott and Clarisse Sieckenius de Souza. Getting the Message Across in RST-based Text Generation. In Robert Dale, Chris Mellish and Michael Zock, Eds., *Current Research in Natural Language Generation*, pages 47–73, Academic Press.

[18] Advaith Siddharthan. 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)*, pages 60–65, Philadelphia, USA, 2002.

[19] Advaith Siddharthan. 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)*, pages 7–14, Budapest, Hungary, 2003.

[20] Advaith Siddharthan. *Syntactic simplification and Text Cohesion*. PhD thesis, University of Cambridge, UK., 2003.

[21] Advaith Siddharthan and Ann Copestake. Generating referring expressions in open domains. To appear in *Proceedings of the 42th Meeting of the Association for Computational Linguistics Annual Conference (ACL 2004)*, Barcelona, Spain, 2004.

[22] Sandra Williams, Ehud Reiter, and Liesl Osman. 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)*, pages 127–134, Budapest, Hungary, 2003.

Received June 17 2004