Preserving Discourse Structure when Simplifying Text

Advaith Siddharthan
Natural Language and Information Processing Group
Computer Laboratory, University of Cambridge
as372@cl.cam.ac.uk

Abstract

Text simplification involves restructuring sentences by replacing particular syntactic constructs (like embedded clauses and appositives). The aim is to make the text easier to read for some target group (like aphasics and people with low reading ages) or easier to process by some program (like a parser or machine translation system). However, sentence-level syntactic restructuring can wreak havoc with the discourse structure of a text, actually making it harder to comprehend, and possibly even altering its meaning. In this paper, we present and evaluate techniques for detecting and correcting disruptions in discourse structure caused by syntactic restructuring. In particular, we look at the issues of preserving the rhetorical relationships between the original clauses and phrases and preserving the anaphoric link structure of the text.

1 Introduction

Syntactic restructuring involves replacing particular syntactic constructs (like embedded clauses and appositives) in sentences. The aim is usually to reduce their grammatical complexity to make the text either easier to read for some target group (like aphasics and people with low reading ages) or easier to process by some program (like parsers or machine translation systems). When we constrain the restructuring operations to preserve the meaning and information content of the original text, we call the process *text simplification*. Syntactic transforms for dis-embedding relative clauses were first suggested as a preprocessing step for parsers (Chandrasekar et al., 1996; Chandrasekar and Srinivas, 1997) as they reduce sentence length and hence improve parser throughput. They were later used as part of a text simplification project aimed at making newspaper text accessible to aphasics (Carroll et al., 1999; Devlin, 1999). We illustrate syntactic simplification with an example. The sentence (1) a. contains two relative clauses and one conjoined verb phrase. Our text simplification system can simplify (1) a. to (1) b.

(1) a. 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.

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

A broad coverage text simplification system is expected to be useful to people with language disabilities like aphasia (Carroll et al., 1999; Devlin, 1999), 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, text simplification is useful as a preprocessing tool to improve the performance of other applications like parsing and machine translation (where performance deteriorates rapidly with

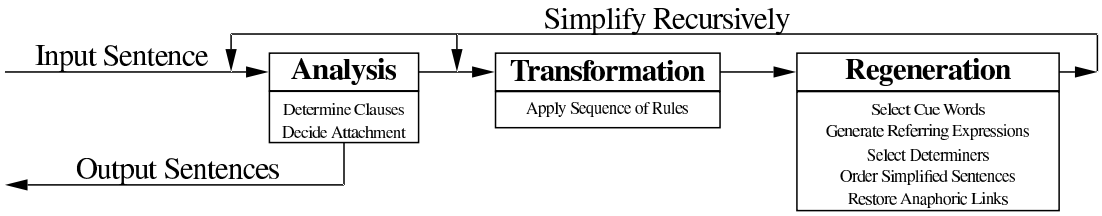

Figure 1: An Architecture for Syntactic Simplification

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 and Srinivas (1997), 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 of the sort illustrated by example 1. Not considering these discourse implications could result in the resultant text losing coherence, thus making it harder to read, or alter the intended meaning; in either case, making the text harder to comprehend. Our architecture (figure 1) therefore uses a third stage—*regeneration*, that we describe in this paper.

In section 2, we describe how to preserve the rhetorical relations (Mann and Thompson, 1988) that existed between clauses and phrases in the original text and ensure that we do not introduce spurious relations and conversational implicatures.

Applying syntactic transforms on text containing pronouns can cause further discourse level problems. In section 3, we discuss how syntactic transforms can result in discourse referents getting introduced in different orders, with different grammatical relations, and how this could make it hard for a reader (or program) to correctly resolve pronouns further in the text.

In section 4 we conduct a corpus evaluation of the techniques described in sections 2 and 3.

2 Preserving Rhetorical Relations

In this section, we discuss how generation issues like cue-word selection, referring expression generation, determiner choice and sentence ordering can be resolved so as to minimise disruption in the text's rhetorical structure. Then, in section 3, we show that the process of preserving rhetorical structure can unavoidably result in the destruction of the anaphoric link structure of a document and provide techniques to restore this structure.

2.1 Using Cue Words

Subordinating conjunctions connect clauses and make one of the clauses subordinate. Subordinating conjunctions also act as cue words that define the relationship between the conjoined clauses. When separating out the conjoined clauses, we can preserve the rhetorical relation between them by introducing a new cue word like *however* or *then*:

(2) 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. **However**, it's the tritest lip service.

(3) 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. A car bomb blast outside the US embassy in Nairobi killed 219 people **then.**

The table below gives a list of conjunctions (coordinating, subordinating and correlative) and the corresponding cue word that our algorithm introduces:

Conjunctions	Cue Word
although, though, whereas, but	however
or, or else	otherwise
even though	still
if, if...then	suppose...then
when	then
not only...but also	also
because, since, as	hence
and	

Our algorithm does not separate out conjoined clauses in cases where there is no appropriate cue

word; for example, when the conjunction is *unless*. The placement of cue words is sentence initial, apart from *then* that is placed sentence finally (a sentence initial *then* would erroneously trigger a *chronological sequence* relation). The ordering of simplified sentences is described in section 2.4.

2.2 Generating Referring Expressions

In the examples above, the extracted clause had a subject and could be made into a stand alone sentence trivially. However, when splitting a sentence into two by dis-embedding a relative clause, we need to provide the clause with a subject. The referent noun phrase hence gets duplicated, occurring once in each simplified sentence. This phenomenon also occurs when separating out conjoined verb phrases and extracting appositives. We 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:

(4) 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.

(4) 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.

Existing referring expression generation algorithms (Reiter and Dale, 1992; Dale and Haddock, 1991) can't cope with open domains like newspaper text as they assume a classification of adjectives which is possible only for very restricted domains. We have proposed a new algorithm (Siddharthan and Copestake, 2002) that relies on WordNet synonym and antonym sets and gives equivalent results on the examples cited in the literature and improved results in other cases that prior approaches cannot handle. This algorithm is suitable for open domains like newspaper text and has been evaluated on the text-simplification task using Wall Street Journal data with promising results (summarised in section 4).

2.3 Determiner Choice

In example 4, the relative clause attached to a proper noun. However, in general, we have to decide on what determiners to use. This decision depends on the rhetorical relation between the extracted clause or phrase and its referent NP.

In the non-restrictive case (for either appositives or relative clauses), the rhetorical relation is that of *elaboration*. This relation continues to hold when we make the clause into the second sentence:

(5) 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.

For extracting non-restrictive constructs, we only need to ensure that the referring expression contains a definite determiner. The determiner *this* is stronger than *the* and can only be used if there is no future reference that uses the determiner *the*.

When simplifying restrictive clauses, the relationship between the clause and the referent noun phrase is that of *specification*; that is, 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:

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

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

The indefinite article is not introduced if the NP contains a numerical attribute (eg. ***...two conversions*** *which turned out to be crucial.*). The referring expression contains a definite determiner as usual. The algorithm for selecting determiners is:

Algorithm *select_determiner*

1. `IF` restrictive clause `THEN`
   `IF` head noun is not a proper noun `AND` NP does not contain a numerical attribute `THEN`
   introduce indefinite determiner (*a* or *some*) in NP in the first sentence

2. `IF` no future references to the NP `THEN`
   introduce *this* or *these* in referring expression
   `ELSE` introduce *the* in referring exp.

2.4 Sentence Order

In general, the clause order should be preserved in the transformed sentences. However, there are a few exceptions. In the following example, the lack of a suitable cue word for the *reason* relation forces us to change the clause ordering and use the cue word for the *consequence* relation.

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

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

We also need to reverse the clause ordering when extracting non-restrictive clauses that attach to noun phrases in the subject position; the *elaboration* clause or phrase has to come second. This is illustrated in example 5 above. Also, the *elaboration* relation tends to get lost if the second sentence is separated from the noun phrase being elaborated by too much text. This can happen if the first sentence is very long, or if it contains another construct to be simplified. Consider:

(8) 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.

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.

c. 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.

It is obvious that the ordering (8) b. is less disruptive than (8) c. In such cases, using sentence order to preserve rhetorical relations is counterproductive and we make the extracted clause the first sentence.

Algorithm *sentence_order*

1. *order* = "preserve"
2. `IF` cue word introduction changes clause order `THEN` *order* = "reverse"
3. `IF` non-restrictive clause `THEN` `IF` referent NP is a subject `THEN` *order* = "reverse"
4. `IF` length($sent_1$)-length($sent_2$) > threshold `THEN` *order* = "reverse"
5. `IF` $sent_2$ can be simplified further `THEN` *order* = "reverse"
6. `RETURN` *order*

3 Preserving Anaphoric Structure

Syntactic restructuring that involves splitting sentences or changing their voice can change the grammatical function of NPs and alter the order in which they are introduced into the discourse. This can affect the reader's ability to correctly resolve pronouns further in the text. If we cannot ensure that the most salient (Lappin and Leass, 1994; Kennedy and Boguraev, 1996) entities before simplification remain the most salient after simplification, we have to consider the possibility of broken anaphoric links. We do this in section 3.2.

When syntactic restructuring reverses the original clause order, this disruption in the anaphoric link structure can become evident in the restructured sentences themselves. We illustrate this in section 3.1.

In both cases, our approach is the same; we make use of a pronoun-resolution algorithm in deciding what to generate. The discussion in this section is based on *salience* and our implementation uses a shallow version of the Lappin and Leass (1994) algorithm. It is worth pointing out in advance that in the examples that follow, we use the term *salience* to mean "salience, as calculated by our algorithm", which may differ slightly from other calculations that use differently weighted features.

We use a three sentence discourse window containing the sentence to be simplified and the two previous sentences and calculate the salience of entities at the end of this window. We then simplify the required sentence, splitting it into two or changing its voice from passive to active. We then check that any pronouns in this sentence resolve to the same antecedents in the original and simplified text. If not, we need to replace them with referring expressions. This process needs to continue till the relative salience of entities in the original and restructured text is the same, at which point we know that the resolution of future pronouns will not be affected by our restructuring. We now illustrate the process with examples.

3.1 Problems with reversing Clause Order

Consider:

(9) a. Incredulity is an increasingly lost art.

b. It requires a certain self-confidence to go on holding the line that Elvis Presley [1] isn't in an underground recording studio somewhere.

c. David Beckham[2] is prone to provoking revisionist hints because the virtues he[2] represents are rare not only in the general population but especially so in football.

When we restructure sentence (9) c. into (9) c'. below, we need to check that the pronouns continue to refer to the same antecedents.

(9) c'. The virtues **he**[1] represents are rare not only in the general population but especially so in football. Hence, David Beckham is prone to provoking revisionist hints.

Our salience-based pronoun resolution system resolves *he* to *David Beckham* in the original text, but incorrectly to *Elvis Presley* in the restructured text. We therefore need to replace *he* by *David Beckham* (its antecedent in the original text). We then check whether the *David Beckham* in the second sentence would, if replaced by the pronoun *he*, still be interpreted correctly. Our pronoun resolution system tells us it will. Hence we can safely simplify sentence (9) c. to (9) c''. below:

(9) c''. The virtues David Beckham[2] represents are rare not only in the general population but especially so in football. Hence, he[2] is prone to provoking revisionist hints.

Algorithm *fix_restructured_sentence*

1. FOR every pronoun in restructured sentences DO 2-3
2. resolve pronoun in original and restructured text.
3. IF they are not the same THEN
   (a) replace pronoun in restructured text with referring expression for antecedent in original text
   (b) IF that antecedent NP has ended up in second sentence THEN
      i. temporarily replace that NP with pronoun
      ii. check that it resolves correctly.
      iii. IF it does THEN
         make the replacement in i permanent
      ELSE
         withdraw replacement in i

We still need to ensure that future anaphoric links are not affected. This is described next.

3.2 Fixing future Anaphoric Links

We now describe how we can tell when future anaphoric links will be affected, and how we can fix disrupted links.

3.2.1 Transforms that preserve Relative Salience

In example 9 above, the five most salient classes at the end of sentence (9) c. in the original text are:

David Beckham, revisionist hints, virtues, general population, football

The five most salient classes at the end of sentence (9) c''. in the restructured text are:

David Beckham, revisionist hints, virtues, general population, football

We find that the relative salience of entities is preserved. This tells us that the reader will be able to resolve future pronouns correctly.

For another example, consider:

(10) a. The Supreme Court agreed to decide whether the federal Pension Benefit Guaranty Corp. may require LTV Corp. to reassume funding responsibility for a $2.3 billion shortfall in the company's pension plans .

b. The high court's decision may affect the stability of many large corporate pension plans that have relied on the availability of pension insurance provided by the federal insurance agency.

c. The agency[1], 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.

At the end of sentence (10) c., the top 5 salience classes are (in order):

agency, pension benefits, 30 million private-sector workers, part, single-employer pension plans

When we split sentence (10) c. the first time, we choose to order the simplified sentences as (10) c'. (sentence (10) c. is the same as sentence 8 that was dealt with in section 2.4).

(10) c'. The agency[1] is funded through insurance premiums from employers. The agency[1] insures pension benefits for some 30 million private-sector workers who take part in single-employer pension plans.

When sentence (10) c. is replaced by (10) c'. the top 5 salience classes are:

agency, pension benefits, 30 million private-sector workers, part, single-employer pension plans

Again, we find that the relative salience of entities is preserved by this transform and hence future anaphoric links will not be disturbed. As an illustration of this, consider the sentence following the simplified sentence (10) c'. in the original text:

(10) d. It[1] recently reported assets of $2.4 billion and liabilities of $4 billion.

Our anaphora resolution algorithm resolves the pronoun (*it*) in sentence (10) d. identically (to *agency*) for the simplified and original texts, suggesting that we can safely leave it as it is.

3.2.2 Transforms that alter Relative Salience

If a clause attaches to a non-subject NP, the discourse structure is invariably disturbed. Consider:

(11) a. Back then, scientists[1] had no way of ferreting out specific genes, but under a microscope they[1] could see the 23 pairs of chromosomes in the cells that contain the genes.

b. Occasionally, gross chromosome damage was visible.

c. Dr. Knudson[2] found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent[3], who had necessarily had the disease.

At the end of sentence (11) c., the top 5 salience classes are:

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 (11) c'. or (11) c".

(11) c'. A parent[3] had necessarily had the disease. Dr. Knudson[2] found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from the parent.

c". Dr. Knudson[2] found that some children with the eye cancer had inherited a damaged copy of chromosome No. 13 from a parent. The parent[3] had necessarily had the disease.

When sentence (11) c. is replaced by (11) c'., the top 5 salience classes are:

Dr. Knudson, children, damaged copy, parent, eye cancer

When sentence (11) c. is replaced by (11) c"., the top 5 salience classes are:

parent, disease, Dr. Knudson, children, damaged copy

There is now a conflict between preserving the discourse structure in terms of anaphoric links and preserving the discourse structure in terms of rhetorical relations. The non-restrictive relative clause has an *elaboration* relationship with the referent NP. To maintain this *elaboration* relationship after simplification, the dis-embedded clause needs to be the second sentence, as in (11) c". However, this ordering significantly disrupts the relative salience of different entities that is more or less preserved by the ordering (11) c'. This conflict between picking the ordering that preserves anaphoric links and the ordering that preserves rhetorical structure 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 solution is to select the ordering that preserves rhetorical structure ((11) c".) and detect and then fix broken anaphoric links as described next.

We detect and fix broken anaphoric links as follows. We consider each sentence following the simplified sentence. For each pronoun we encounter, we use our anaphora resolution procedure to find its antecedent in both the original and simplified texts. If the antecedents differ, we replace the pronoun by a referring expression for its correct antecedent (determined using the original text). The salience scores are then recomputed. This process continues until the relative salience of entities in the original and simplified text are the same again.

Now consider the sentence that follows the simplified sentence (11) c.

(11) d. Under a microscope he$^{original:2,\ simplified:3}$ could actually see that a bit of chromosome 13 was missing.

Our anaphora resolution algorithm resolves the pronoun *he* in sentence (11) d. to *Dr. Knudson* in the original text, but incorrectly to *parent* in the simplified text. To preserve the meaning of the original text, we need to replace the pronoun in the simplified text with a new referring expression

for its antecedent in the original text. Thus we, replace (11) d. with (11) d'. below:

(11) d'. Under a microscope Dr. Knudson[2] could actually see that a bit of chromosome 13 was missing.

Now, we find that at the end of this sentence, the five most salient classes are the similar for the original text:

Dr. Knudson, microscope, bit, chromosome, children

and the simplified text:

Dr. Knudson, microscope, bit, chromosome, parent

This tells us that future anaphoric link will not be disrupted by our simplification process.

This process of fixing anaphoric links looks quite daunting. However, in practice, as salience decreases rapidly at sentence boundaries, we rarely (in only 2% of the cases; refer to section 4 on evaluation) have to consider more than just the one sentence succeeding the transformed one. Hence the loop in step 2 below is rarely executed more than once.

Algorithm *fix_future_links*

1. `IF` relative salience of entities in original and transformed text is same, `THEN` *flag* = 0 `ELSE` *flag* = 1
2. `WHILE` *flag* `DO` steps 3 and 6
3. `FOR` every pronoun in next sentence `DO` 4-5
4. resolve pronoun in original and transformed text.
5. `IF` they are not the same `THEN` replace pronoun in transformed text with referring expression for antecedent in original text
6. `IF` relative salience of entities in original and transformed text is same, `THEN` *flag* = 0 `ELSE` *flag* = 1

4 Evaluation

For many of the algorithms presented in this paper, evaluation is difficult. It is hard to quantify the effects of text restructuring on a text's discourse-level structure. The effects of many regeneration decisions (eg. cue word selection and sentence ordering) on the regenerated text are largely stylistic, which makes evaluation subjective.

The referring expression generator gives *correct* results on ~81%, *acceptable* results on ~12% and *wrong* results on ~7% of cases, when evaluated on WSJ data (Siddharthan and Copestake, 2002). A generated referring expression was labelled as *correct* if it was optimal and factually accurate, as *acceptable* if the generated expression was accurate but suboptimal and as *wrong* if the generated expression was nonsensical or ambiguous with a distractor. The mistakes mainly arose due to multi-word expressions being incorrectly analysed as multiple attributes to generate, for example, *the care products* from *personal care products*.

That evaluation considered only examples where there were one or more distractors in context. However, in over 90% of the cases for which we need to generate referring expressions, the contrast set of distractors is empty, which means the error rate for our application is less than 1%.

For a preliminary evaluation of the other regeneration components, we used a corpus of newspaper columns and news reports, travelogues, medical articles and literary extracts and manually examined the output of our text simplification algorithm on the first 250 embedded clauses.

Our method for selecting determiners gave wrong results on ~4% of examples. The following examples show the output of our algorithm for two sentences. In example 12, the adjectival pronoun *his* would have been preferable to *this* in the referring expression. In example 13, the referring expression should have had the indefinite determiner *a*.

(12) a. Puckett played in 10 All-Star games during his career, which was cut short by glaucoma.

b. Puckett played in 10 All-Star games during his career. **This** career was cut short by glaucoma.

(13) a. Petroleum companies were also popular because of expectations of a weaker dollar, which cuts crude-oil prices.

b. Petroleum companies were also popular because of expectations of a weaker dollar. **This** weaker dollar cuts crude-oil prices.

The algorithms on preserving anaphoric links can be evaluated more objectively. 20% the cases contained pronouns in the sentence to be simplified. Assuming that salience based anaphora resolution algorithms perform with an accuracy of ~0.65 on open domains (Barbu

and Mitkov, 2001; Preiss, 2002), algorithm *fix_restructured_sentence* can be expected to have an error rate of around $20 \times 0.35\% = \sim 7\%$. In practice, our algorithm made mistakes in only ~2.5% of the cases. This is because our anaphora resolution algorithm performs with an accuracy of ~0.80 on our corpus, and because intra-sentential pronouns are relatively easy to resolve.

The loop in algorithm *fix_future_links* needed to be executed only once in 98% of the cases. In the remaining 2% cases, the loop needed to be executed twice. 15% of the cases contained anaphora in the sentence following the simplified sentence. Assuming again that salience based anaphora resolution algorithms perform with an accuracy of ~0.65 on open domains, algorithm *fix_future_links* can be expected to have an error rate of around $15 \times 0.35\% = \sim 5\%$. Experimentally, using our anaphora resolution on this corpus, we report an error rate of ~4%.

5 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 discourse structure. We have presented and evaluated techniques for detecting and correcting these disruptions in discourse structure. In particular, we have examined the issues of preserving the rhetorical relationships between the original clauses and phrases and preserving the text's anaphoric link structure. We believe that the techniques we have described to analyse the simplified discourse might prove useful to other NLP applications that involve transforming text; in particular, summarisation and translation.

We have tried to evaluate our algorithms intrinsically. Future work includes an extrinsic evaluation of these algorithms, using comprehension tests on subjects. This would be more useful than intrinsic evaluations in judging the benefits of text restructuring to target groups like aphasics.

Acknowledgement

This work was partially supported by the EC Fifth framework project Deep Thought IST-2001-37836.

References

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 2001), Toulouse, France*, pages 34–41.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying English text for language impaired readers. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Bergen, Norway.*

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 Sixteenth International Conference on Computational Linguistics (COLING '96)*, Copenhagen, Denmark.

Robert Dale and N. Haddock. 1991. Generating referring expressions involving relations. In *Proceedings of EACL-91, Berlin*, pages 161–166.

Siobhan Devlin. 1999. Simplifying natural language for aphasic readers. Technical report, Ph.D. thesis, University of Sunderland, UK.

Christopher Kennedy and Branimir Boguraev. 1996. Anaphora in a wider context: Tracking discourse referents. In *European Conference on Artificial Intelligence*, pages 582–586. John Wiley and Sons, Ltd, London/New York.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

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

Judita Preiss. 2002. Choosing a parser for anaphora resolution. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2002), Lisbon, Portugal*, pages 175–180.

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, Nantes, France*, pages 232–238.

Advaith Siddharthan and Ann Copestake. 2002. Generating anaphora for simplifying text. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2002), Lisbon, Portugal*, pages 199–204.