An Architecture for a Text Simplification System

in Proceedings of the Language Engineering Conference 2002 (LEC 2002)

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

Abstract

We present a pipelined architecture for a text simplification system and describe our implementation of the three stages— analysis*,* transformation *and* regeneration*. Our architecture allows each component to be developed and evaluated independently. We lay particular emphasis on the discourse level aspects of syntactic simplification as these are crucial to the process and have not been dealt with by previous research in the field. These aspects include generating referring expressions, deciding determiners, deciding sentence order and preserving rhetorical and anaphoric structure.*

1 Introduction

Text Simplification is an NLP task that aims to rewrite sentences to reduce their syntactic or lexical complexity while preserving their meaning. We illustrate syntactic simplification with an example. The sentence (1) a. contains two relative clauses and one conjoined verb phrase. Our system is capable of simplifying (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 the full purchasing agents report. The Chicago report gives an indication of what the full report might hold. The full report is due out today.

We list the potential uses of a text simplification system in section 2. We then describe syntactic constructs that can be simplified in section 3 and present our architecture in section 4, describing the individual modules in sections 5, 7 and 8.

2 The Uses of Text Simplification

A broad coverage text simplification system is expected to be useful to the following classes of users:

- **People with Language Disabilities like Aphasia**
  While there are a variety of language problems associated with aphasia, depending on the extent and location of brain damage and the level of pre-aphasia literacy amongst other things, aphasics in general have trouble with long sentences, infrequent words and complicated grammatical constructs including embedded clauses and passive voice [1, 5].

- **Adults learning English**
  Elementary English texts tend to be aimed at children. While teaching English to adults, as part of an Adult literacy programme or as a foreign language, a text simplification toolkit could be used to construct texts that are relevant to adults, while of the desired linguistic complexity. Simplification of arbitrary internet text would also be useful to non-native English speakers and people with low reading ages.

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

Further, text simplification is useful as a pre-processing tool to improve the performance of other NLP applications like:

- **Parsing**
  Simplification tools that generate shorter sentences

could be used to pre-process texts before feeding them to a full-blown parser. This was the motivation for some early work [2, 3] on simplification. Long sentences are problematic for parsers due to their high levels of ambiguity. Shortening sentences prior to parsing would increase parser throughput and reduce parser timeouts. It has been suggested [3] that the parses of simplified sentences can be combined to give the parse for the original sentence.

- **Machine Translation**
  The performance of machine translation systems deteriorates rapidly with increased sentence length. Simplified sentences would be easier to translate correctly. If the generation process for the target language can pick out sentences that can be combined, the resulting text might be more readable. This can be achieved by opportunistically packing information into sentences [10].

- **Text Summarisation**
  Text simplification results in less information per sentence. This is likely to improve the performance of summarisation systems that are based on sentence extraction as smaller units of information are being extracted.

3 Syntactic Simplification

We now illustrate how various syntactic constructs [12] can be simplified.

Adjectival (or Relative) Clauses

Relative clauses modify noun phrases and always follow them. They can be either restrictive or non-restrictive. Non-restrictive relative clauses are usually marked by punctuation:

(2) a. The jury also commented on the Fulton court, *which has been under fire for its practices in the appointment of appraisers, guardians and administrators*.
  b. The jury also commented on the Fulton court. The Fulton court has been under fire for its practices in the appointment of appraisers, guardians and administrators.

Adverbial clauses

Adverbial clauses modify a verb. Unlike adjectival clauses, they can appear in different positions in a sentence and don't need to immediately follow the verb. Introductory adverbial clauses are followed by a comma and adverbial clauses in the middle of a sentence are enclosed by commas.

(3) a. *Needing money to pay my rent*, I forced myself to beg my parents.
  b. I needed money to pay my rent. I forced myself to beg my parents.

Coordinated Clauses

Coordinating conjunctions connect words, phrases or clauses of equal importance. The most common coordinating conjunctions are *and, but* and *or*.

(4) a. The army also arrested 32 Palestinians in the West Bank overnight ***and*** *claimed 23 of those were "wanted suspects" in attacks on Israel*.
  b. The army also arrested 32 Palestinians in the West Bank overnight. The army claimed 23 of those were "wanted suspects" in attacks on Israel.

Subordinated clauses

Subordinating conjunctions only connect clauses and make one of the clauses subordinate. Some subordinating conjunctions are *after*, *rather than*, *although*, *since*, *as*, *so*, *as if*, *as long as*, *as though*, *though*, *because*, *unless*, *before*, *until*, *even if*, *when*, *even though*, *whenever*, *whereas*, *if only*, *wherever*, *in order that*, *while*.

(5) 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.

Correlated clauses

Correlative conjunctions operate in pairs to connect sentence elements. Examples are *either...or*, *neither...nor*, *both...and*, *if...then*, *not only...but also*, *whether...or*.

(6) a. This will undermine our country's independence **not only** in matters of security **but also** *with regard to the exercise of foreign and domestic policy options*.
  b. This will undermine our country's independence in matters of security. This will also undermine our country's independence with regard to the exercise of foreign and domestic policy options.

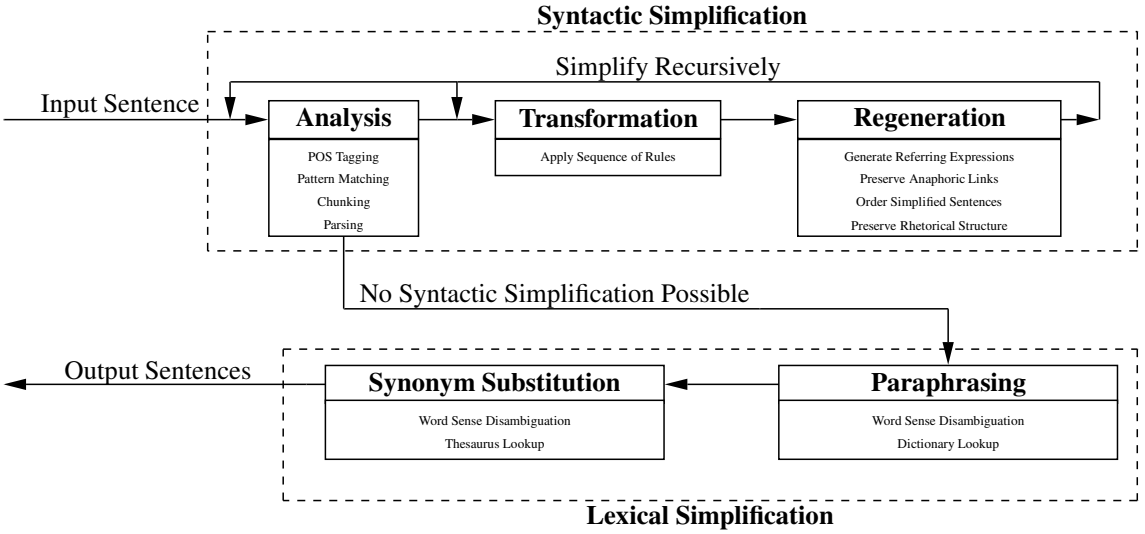

Figure 1. The Architecture of a Text Simplification System

Participial Phrases

Participles do not function as verbs unless preceded by an auxiliary. The lack of an auxiliary suggests the presence of a participial phrase. In general, sentences can be simplified by making a new sentence with the preceding NP as subject and the Participial Phrase (with an appropriate auxiliary) as VP.

(7) a. Alicia, *running down the street*, tripped and fell.
b. Alicia was running down the street. Alicia tripped and fell.

Appositive Phrases

Appositive phrases further explain a noun phrase:

(8) a. The company chose John Rice, *vice president of public affairs*, as its new chief executive officer.
b. The company chose John Rice as its new chief executive officer. John Rice was vice president of public affairs.

Voice

Aphasics often have trouble with closed class words like *was* and *by*. As a result they tend to misinterpret passive voice constructs. For example, they might interpret *Mary was punched by John* as *Mary punched John* which is the standard Subject-Verb-Object form. Passive voice constructs can be simplified by changing them to active voice.

4 Architecture

Previous research [3] treats syntactic simplification as a two-stage process— *analysis* followed by *transformation*. The first stage provides a structural representation of a sentence and the second stage uses a sequence of rules to transform this representation and flatten the resulting structures into plain text. In practice, syntactic simplification does not work very well without taking inter-sentential discourse considerations into account. Our architecture, therefore, uses a third stage (*regeneration*) in addition to the *analysis* and *transformation* stages. Figure 1 shows a simplified block diagram of our architecture. The upper dotted box shows the syntactic simplification component and the lower box shows the lexical simplification component. Lexical simplification can involve paraphrasing words (especially verbs) with their dictionary definitions [6] and replacing words with simpler synonyms [5]. In this paper, we concentrate on the architecture for syntactic simplification and in particular, the *regeneration* stage that has not received treatment in the literature. We now describe in detail each of the three stages in the syntactic simplification component and the interactions between them.

5 Analysis

There are different levels of analysis available to us. The more detailed analyses (like full parses) are less robust and computationally more expensive than shallower analyses (like chunking and POS tagging). Unfortunately, sentences that need simplification tend to cause parsers problems due to their long length and high degree of ambiguity. Our strat-

egy is therefore to use only as detailed an analysis as is required.

The analysis stage performs two functions. Firstly, it identifies syntactic structures that can be simplified. This can be done reliably using pattern matching techniques on POS tagged text with noun and verb groups chunked; for example, consider:

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

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

The pattern:

`NounGroup ", who" VerbGroup * "," *`

indicates the presence of a non-restrictive relative clause in the example (9) a.

Once a sentence has been identified as suitable for simplification, the analysis stage provides the transformation stage with the required analysis. When splitting sentences by dis-embedding relative clauses, separating out conjoined clauses or making new sentences out of appositives, the analysis stage needs to decide attachment and clause/phrase boundaries. For example, to simplify (9) a. to (9) b., we need to know that the relative clause attaches to *Cathy Tinsell* rather than *South London*. We also need to know that the relative clause does not end at the first comma, but extends to the end of the sentence. Relative clause attachment decisions can be made reliably in a machine learning framework [14], using WordNet hierarchies [11] and prepositional preferences (refer to section 6 for performance details). Further, clause boundaries can be determined reliably (refer again to section 6) using very shallow processing when the clause is enclosed in punctuation [14]. Therefore, the simplification of non-restrictive relative clauses and appositives as well as other constructs marked by punctuation can be accomplished using just POS tagging and chunking at the analysis stage.

However, the clause boundary problem is more complex for restrictive dependant clauses and other constructs not marked by punctuation. A parser output is then required for the correct analysis.

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

Output Specification for Analysis Stage

1. Words should be POS tagged
2. The boundaries of the clauses or phrases to be extracted should be marked
3. The noun phrases that the clauses or phrases to be extracted attach to should be marked.
4. Elementary noun groups should be marked and annotated with information about grammatical function and whether they are within oblique or embedded constructs.

The specifications 2 and 3 are required by the transformation stage and the specifications 1 and 4 are required by the regeneration stage.

6 Evaluating the Analysis Stage

Our previous work on the analysis stage deals with two kinds of clause attachment ambiguities. We have shown [14] that deciding between local and wide attachment when the noun phrase preceding the clause has the structure `NP Prep NP` can be done with an accuracy of ˜82% .

For deciding attachment when the clause is preceded by appositives, for example:

(10) One man who is likely to reap the benefits is Vaino Heikkinen, aged 67, a farmer in Lieksa, 10km from the Soviet border, *who claims a Finnish record for shooting 36 bears since 1948.*

we report [14] an accuracy of ˜99%.

An analysis of the Penn Treebank [9] revealed that ˜25% of relative clauses were preceded by complex noun phrases of the type `NP1 Prep NP2`. ˜10% of relative clauses were preceded by noun phrases with appositives after them. The rest were unambiguous. Therefore, our analysis component can make attachment decisions with a precision of $80 \times .25 + 99 \times .10 + 100 \times .65\% = \tilde{}95\%$.

Clause boundary identification in the presence of punctuation can be made with a precision of ˜97% [14]. For other clauses, we need to rely on parsers, and the performance of the analysis stage becomes dependent on the performance of the parser (˜80% on open domains).

7 Transformation

The transformation stage takes as input a representation that marks the boundaries of the construct to be extracted as well as the noun phrase that the construct attaches to. The transformation stage consists of straightforward handcrafted rules like the following:

V W:NP$_{ant}$, Rel_Clause(X:Rel_Pr Y), Z.

↓

V W Z. W Y.

which can be interpreted as "If a sentence consists of any text V followed by the antecedent noun phrase W, a relative clause (consisting of a relative pronoun X and a sequence of words Y) enclosed in commas and a sequence of words Z, then the embedded clause can be made into a new sentence with W as the subject NP"

It is possible that there is more than one construct that can be simplified in a sentence. In this case, constructs are simplified in the order in which they occur in the sentence; that is, from left to right. Our experiments suggest that this results in the best ordering of the simplified sentences in the regeneration stage. For the example in the introduction, this ordering results in (1) a. getting simplified to (1) b.

8 Regeneration

There are many issues that crop up when regenerating transformed text. Some of them are largely stylistic, such as the use of referring expressions. Others, such as deciding sentence order and preserving rhetorical and anaphoric link structure, are vital to preserve the coherence and meaning of the text.

8.1 Generating Referring Expressions

When splitting a sentence into two by dis-embedding a relative clause, the referent noun phrase gets duplicated, occurring once in each simplified sentence. This also happens with conjoined clauses and appositives. We 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 (9) c. below with (9) b. above.

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

Existing referring expression generation algorithms [13, 4] 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 [15] 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 [15] 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 9).

8.2 Selecting Determiners

When extracting phrases or clauses, we introduce a definite determiner in the referring expression that is the subject of the second sentence:

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

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

Further, when simplifying restrictive clauses, we introduce an indefinite determiner in the noun phrase that the clause attaches to:

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

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

8.3 Preserving Discourse Structure

We look at three regeneration issues that affect discourse structure:

1. **Deciding Sentence Order**
   When the simplification rule splits a sentence into two, we need to decide the order in which to output the simplified sentences.

2. **Preserving Rhetorical Relations**
   We need to ensure that the rhetorical relations that held between clauses and phrases in the original sentence are not altered by the simplification process. For example, non-restrictive relative clauses and appositive phrases have an *elaboration* relationship with the NPs they attach to. This needs to be preserved by the simplification process.

3. **Preserving the Relative Salience of Entities**
   Splitting sentences or changing their voice can change the grammatical function of noun phrases and alter the order in which they are introduced into the discourse. This can affect the reader's ability to correctly resolve pronouns further in the text. We need to ensure that the most salient [8, 7] entities before simplification remain the most salient after simplification.

We use a three sentence discourse window containing the sentence to be simplified and the two previous sentences. We first measure the salience of entities in this window. We then simplify the required sentence. This involves splitting it into two or changing its voice from passive to active. We

then recalculate the salience of the entities, once for each possible ordering of the simplified sentences. We discuss two different cases below.

8.3.1 Clause Attaches to Subject NP

Consider:

(13) 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 .

(14) 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.

(15) 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 15, 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 15 the first time, we have the choice of ordering the simplified sentences as either of (15) a. or (15) b.

(15) a. 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.

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

When sentence 15 is replaced by (15) a. the top 5 salience classes are:

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

When sentence 15 is replaced by (15) b., the top 5 salience classes are:

agency, insurance premiums, employers, pension benefits, 30 million private-sector workers

We see that the first ordering (15) a. better preserves the relative salience of entities. This should therefore be the preferred ordering.

8.3.2 Clause Attaches to Non-Subject NP

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

(16) 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.

(17) Occasionally, gross chromosome damage was visible.

(18) 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 18, 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 (18) a. or (18) b.

(18) a. 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.

b. 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 18 is replaced by (18) a., the top 5 salience classes are:

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

When sentence 18 is replaced by (18) b., 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 (18) b. However, this ordering significantly disrupts the relative salience of different entities that is more or less preserved by the ordering (18) a. 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 ((18) b.) and detect and fix broken anaphoric links later. We show how this can be done in section 8.3.3.

8.3.3 Fixing Broken Anaphoric Links

As the section above shows, dis-embedding relative clauses can result in altering the relative salience of entities. This also happens when changing passive voice constructs to active. This means that subsequent pronouns might have to be replaced by referring expressions to prevent problems with disrupted anaphoric links.

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.

In the example in section 8.3.1 above, the sentence following the simplified sentence (15) a. is:

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

Our anaphora resolution algorithm resolves the pronoun in sentence 19 identically for the simplified and original texts. This suggests that we can safely leave it as it is.

Further, we find that after considering this sentence, the five most salient classes are the same for the original text and the simplified text:

agency, assets, liabilities, $4 billion, $2.4 billion

This tells us that future anaphoric link will not be disrupted by our simplification process. Hence, in this example, there are no anaphoric links that need fixing.

Now consider the sentence that follows the simplified sentence (18) b. in the example in section 8.3.2:

(20) 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 20 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 20 with (20) a. below:

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

Now, we find that after considering 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.

9 Evaluating the Regeneration Stage

The referring expression generator gives *correct* results on ˜81%, *acceptable* results on ˜13% and *wrong* results on ˜7% of cases, when evaluated on the text simplification task [15]. 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 proucts*.

For a preliminary evaluation of the other regeneration components, we examined 100 WSJ Treebank examples.

Our method for determining determiners gave wrong results on 9 examples— cases like *a weaker dollar* where the referring expression should have an indefinite determiner and cases like *non-ferrous metals* where the referring expression should have no determiner.

Our method of deciding sentence order gave acceptable results on all 100 cases, though in 8 cases, the other ordering would have been preferable. In all the cases, the pattern of salience change was similar to the examples in section 8. 14 cases contained anaphora in the sentence following the simplified sentence. As current salience based anaphora resolution algorithms perform with an accuracy of 65% on open domains, the module for preserving discourse structure can be expected to have an error rate of $.14 \times 65\% = \tilde{}9\%$.

10 Conclusions and Future Work

We have proposed a general architecture for a text simplification system and described our implementation of

each component. In particular, we have highlighted the discourse level problems that could arise from syntactic text simplification and described techniques of overcoming them. We have also tried to quantify the limits on the performance of the system based on the technology we use in each stage. Future work involves evaluating the performance of the system as a whole. This could be done intrinsically, using user evaluations, or extrinsically, by measuring its affect on the performance of parsers and machine translation systems.

References

[1] J. Carroll, G. Minnen, D. Pearce, Y. Canning, S. Devlin, and J. 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), Bergen, Norway.*, 1999.

[2] R. Chandrasekar, C. Doran, and B. Srinivas. Motivations and methods for text simplification. In *Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING '96)*, Copenhagen, Denmark, 1996.

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

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

[5] S. Devlin. Simplifying natural language for aphasic readers. Technical report, Ph.D. thesis, University of Sunderland, UK., 1999.

[6] N. Kaji, D. Kawahara, S. Kurohash, and S. Sato. Verb paraphrase based on case frame alignment. In *Proceedings of the 40th Annual Meeting of the Assosiation for Computational Linguistics, Philadelphia, USA*, pages 215–222, 2002.

[7] C. Kennedy and B. Boguraev. 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, 1996.

[8] S. Lappin and H. J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.

[9] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large natural language corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330, 1993.

[10] K. McKeown, J. Robin, and K. Kukich. Generating concise natural language summaries. *Information Processing and Management*, 31(5):703–733, 1995.

[11] G. A. Miller, R. Beckwith, C. D. Fellbaum, D. Gross, and K. Miller. Five Papers on WordNet. Technical report, Princeton University, Princeton, N.J., 1993.

[12] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. *A Comprehensive Grammar of the English Language*. Longman, 1985.

[13] E. Reiter and R. Dale. A fast algorithm for the generation of referring expressions. In *Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France*, pages 232–238, 1992.

[14] A. 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 2002), Philadelphia, USA*, pages 60–65, 2002.

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