

Resolving Attachment and Clause Boundary Ambiguities for Simplifying Relative Clause Constructs

In Proceedings of the Student Research Workshop, 40th Meeting of the Association for Computational Linguistics (ACL 2002), pages 60-65

Advaith Siddharthan

Computer Laboratory, University of Cambridge
as372@cl.cam.ac.uk

Abstract

There are two important disambiguation problems to solve when dis-embedding relative clauses for simplifying text—finding the noun phrase the clause refers to and identifying the clause boundary. We argue that we can make these disambiguation decisions more reliably by using local context than by using general purpose tools like wide-coverage statistical parsers.

1 Introduction

The automatic dis-embedding of relative clauses is an important aspect of text simplification, an NLP task that aims to rewrite sentences, reducing their grammatical or lexical complexity while preserving their meaning and information content (Chandrasekar et al., 1996; Carroll et al., 1998). Text simplification is a useful NLP task for varied reasons. The group at UPenn (Chandrasekar et al., 1996; Chandrasekar and Srinivas, 1997) viewed text simplification as a preprocessing tool to improve the performance of their parser. The PSET project (Carroll et al., 1998; Carroll et al., 1999), on the other hand, focused its research on simplifying newspaper text for aphasics, who have trouble with long sentences, infrequent words and complicated grammatical constructs including embedded clauses (Devlin, 1999). Text simplification might also be useful to other groups of people with low reading ages or non-native English speakers using the internet.

Chandrasekar et al. (1996) divide the text simplification task into two stages—analysis and transformation. Previously published work on dis-embedding relative clauses make use of simplification rules (transformation) that act on either linear text or some form of parse tree (analysis). A simple

hand-crafted rule, which handles non-restrictive relative clauses and works on linear text, is:

$$V W:NP, X:Rel_Pr Y, Z. \rightarrow V W Z. W Y.$$

This rule can be interpreted as “If a sentence consists of any text V followed by a 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” and performs simplifications like:

Comments by John Major , who has succeeded Mr. Lawson , also failed to damp market concern. \rightarrow

Comments by John Major also failed to damp market concern. John Major has succeeded Mr. Lawson.

For the transformation stage to work correctly, two issues need to be resolved at the analysis stage—locating the noun phrase the clause refers to and locating the clause boundary. These are essentially disambiguation problems and need to be dealt with irrespective of the exact simplification rules used or the structures they act on. We argue that we can make these disambiguation decisions more reliably by targeting each problem individually and using local context than by relying on general purpose tools like wide-coverage statistical parsers. We describe these problems (including solutions and their evaluation) in sections 2 and 3. We present our conclusions and discuss future work in section 4.

2 Relative Clause Attachment

Current parsers like the ANLT (Briscoe and Carroll, 1995) take the view that determining what a relative pronoun refers to is not a problem that can always be solved in a syntactic framework; hence non-restrictive relative clauses are increasingly being treated by parsers as text adjuncts (Nunberg, 1990), leaving the attachment decisions to anaphora resolution algorithms. On the other hand, anaphora

resolution algorithms based on discourse oriented approaches (Hobbs, 1986; Grosz, 1978), global focus (Grosz and Sidner, 1986) and local focus (Carter, 1987; Webber, 1978) do not deal with relative clause attachment directly.

We discuss two types of ambiguities. The first type involves deciding local vs wide attachment when the noun phrase preceding the relative clause has the structure NP1 Prep NP2; for example, picking the wide attachment reading(underlined) in:

‘[The pace] of [life] was slower in [those days],’ says [51-year-old Cathy Tinsall] from [South London], *who had [five children].*

The second type involves picking the right noun phrase in the presence of appositives; for example, picking *Vaino Heikkinen* in:

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.*

An analysis of the Penn Treebank (Marcus et al., 1993) revealed that 21% of *who* and 27% of *which* relative clauses were preceded by complex noun phrases of the type NP1 Prep NP2. Further, 19% of *who* and 4% of *which* relative clauses were preceded by noun phrases with appositives after them.

2.1 Deciding Local vs Wide Attachment

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

The most important feature for determining relative clause attachment is agency. We made a distinction between *who* and *which* clauses. According to Quirk et al. (1985), the relative pronoun *who* is used to refer to something with *personality* and *which* to something without. In terms of the WordNet hierarchy (Miller et al., 1993), *who* can only refer to hyponyms of the following classes— *humans*,

Prep	P_{who}	P_{which}	Prep	P_{who}	P_{which}
about	0.58	0.43	against	0.53	0.47
among	0.62	0.42	as	0.57	0.43
at	0.46	0.57	before	0.51	0.52
between	0.75	0.41	by	0.63	0.43
during	0.44	0.56	from	0.62	0.53
for	0.55	0.50	in	0.52	0.52
into	0.51	0.51	like	0.39	0.54
near	0.50	0.50	of	0.52	0.52
on	0.62	0.49	over	0.61	0.50
to	0.66	0.61	under	0.34	0.54
with	0.52	0.51	without	0.37	0.54

Table 2: Probability of the preposition selecting for local attachment (for *who* and *which* clauses)

groups(organisations) or *animals*, while *which* cannot refer to *humans*. Features 3-10 and 26-23 in table 1 classify NP1 and NP2 according to the WordNet classes of their head nouns.

We included features for prepositions that the network can make use of when NP1 or NP2 do not have WordNet classes; proper nouns (that could be people, organisations or locations) are very common as arguments to prepositions. Lexicalization over prepositions (having the presence/absence of each preposition as a separate feature) was impractical due to data sparsity problems. We therefore assumed that prepositions only influence attachment indirectly, through their preferences for the agency of their arguments. We classified the subject and object of 15000 occurrences of prepositions (in any context, not just preceding relative clauses) according to their WordNet classes. We introduced two features (14 and 15) for prepositions. For *who* clauses, if the probability of the preposition’s object being *human*, *group(organisation)* or *animal* is greater than that of the preposition’s subject, then the preposition selects for local attachment and feature 14 is set, otherwise feature 15 is set. For *which* clauses, if the probability of the preposition’s object not being *human* is greater than the probability of the preposition’s subject not being *human*, then feature 14 is set, otherwise feature 15 is set. Table 2 gives the probability that the preposition select for local attachment for some common prepositions.

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

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

Table 1: List of Binary Features

2.1.1 Evaluation (*Who* clauses)

The precision of the machine learning approach for deciding attachment for *who* clauses when the preceding noun phrase has the structure NP1 Prep NP2 is shown below.

Data Set	Size	Baseline1	Baseline2	Winnow
Training Set	~200	66.5%	73.3%	91.6%
Test Set	~50	66.5%	73.3%	91.1%

Baseline1: Always attach locally.

Baseline2: Attach according to the preposition’s preferences

The 248 examples were divided into four sets of 50 and one set of 48. An experiment was run with each of the sets as test data (and the other four as training data). The results above are an average of the results of these five experiments. Our approach gives results that are roughly 25% better than the local attachment baseline. For another comparison, we converted the first 100 of these sentences to plain text and parsed them with the Briscoe and Carroll (1993) parser and Briscoe and Carroll (1995) grammar. An analysis of the parse trees gave a recall of 62% and a precision of 69.35%. The local attachment baseline for these 100 examples was 68%. The loss of recall was largely due to non-restrictive relative clauses being attached to the root node of the parse tree as an adjunct, though there were a few sentences that didn’t return meaningful parses.

Error Analysis - WINNOW

41% of the errors came from partitive expressions like {*thousands*| *dozens*| *a lot*| *a number*} of {*people*| *investors*|...} and other cases where NP1 doesn’t have a WordNet class (like *the* {*percentage*| *kind*} of {*Americans*| *guys*|...}). From the text simplification perspective, it is immaterial which at-

tachment is picked for partitives. Simplification is not possible when NP1 is {*kind*| *percentage*|...} and these cases need to be filtered out. 21% of errors arose because the network didn’t learn the rule that a relative clause cannot modify a proper noun without an intervening comma (in instances like *A former backup singer for Ms Midler who had...*). If this is enforced as a hard rule, the precision goes up by almost 2%. The remaining errors arose because the network had genuinely little to go on; for example, *Some 3.8 million of the 5 million who will...*

2.1.2 Evaluation (*Which* clauses)

The precision of the network in deciding attachment for *which* clauses when the preceding NP has the structure NP1 Prep NP2 is shown below.

Data Set	Size	Baseline1	Baseline2	Winnow
Training Set	~400	69.7%	62.7%	77.1%
Test Set	~50	69.7%	62.7%	76.6%

Baseline1: Always attach locally.

Baseline2: Attach according to the preposition’s preferences

The 466 examples were divided into eight sets of 50 and one set of 66. The results above are an average of nine experiments, with each of the sets as test data and the other eight as training data. *Which* clause attachments are not learnt as well as *who* clause attachments. The WordNet hierarchy is obviously useful when exactly one of NP1 and NP2 is *human*. In the majority of cases, however, neither is *human* and, as the second baseline suggests, the prepositions do not provide much of a clue either, so the network has very little to go on.

2.2 Handling Appositives

Deciding the noun phrase the clause refers to in the presence of appositives is an easier task. For a start,

most appositives refer to the same entity as the noun phrase they attach to. So, from the text simplification point of view, the problem is to pick the noun phrase that makes comprehension easier, as either noun phrase would preserve meaning; for example, picking *Laura Dobson* in the following sentence:

“She was an inspirational lady,” says Laura Dobson, a freshman at the University of South Carolina, who had Mrs. Yeargin in the teacher-cadet class last year.

2.2.1 The Algorithm

Given the pattern: $\dots NP_1, NP_2, \dots, NP_n, \text{who} \dots$, the task is to select the NP_i that the relative clause refers to. Our algorithm, based on a manual examination of the 121 sentences in the training set is described in algorithm 1:

Algorithm 1

1. IF only one of the NPs is a Proper Noun THEN select it
2. ELSE IF more than one NP is a Proper Noun THEN
 - (a) IF any of NP_2 through NP_n are a single word proper noun, THEN reject them
 - (b) Of the rest, IF exactly one of them does not contain a preposition THEN select it
 - (c) ELSE IF exactly one of NP_1 and NP_n is a Proper Noun THEN select it
3. ELSE By default, select NP_1

The main intuition behind the algorithm is that when there are appositives (NP_{2-n}) attached to a noun phrase (NP_1), either the noun phrase or one of the appositives (say, NP_i) is being described or specified by the rest. Our aim is to select NP_i . An examination of the WSJ Treebank suggested that (1) if only one of NP_{1-n} is a proper noun, that is usually what the rest are describing, (2b) appositives that contain prepositions tend to describe one of the other noun phrases and (2c) the relative clause tends to attach to either the noun phrase (NP_1) or the last appositive (NP_n). (2a) is a simplistic implementation of a rule to exclude locational proper nouns; for example, to avoid picking *Washington* or *D.C.* in *Company_x, Washington, D.C., who make synthesizers....* This works well on WSJ data but would need to be refined for other text genre.

2.2.2 Evaluation

The performance of algorithm 1 is tabled below.

Data Set	Size	Baseline* %		Algorithm 1 %	
		C	C or A	C	C or A
Training Set ¹	121	88.4	95.0	100.0	100.0
Test Set 1 ²	101	90.1	97.0	99.0	99.0
Test Set 2 ³	58	65.6	87.9	93.1	98.3

* Baseline: Always NP_1 . ¹ Only *who* clauses.

² Only *who* clauses. ³ Only *which* clauses.

C = Correct

A = Acceptable

Interestingly, the algorithm based on a manual analysis of only *who* clauses performs equally well on *which* clauses. These results are subjective, at one level, as we had to manually decide on the “most suitable” referent noun phrase. However, at another level, the results are objective in the sense that the selected noun phrase has to be “a” referent, if not “the simplest” referent of the relative clause. We provide some examples of right and wrong choices to illustrate our criteria. The correct noun phrase is italicised, acceptable ones are in bold font and what algorithm 1 picks is underlined.

- *...Joe Watson, the prosecutor in the case, who is...*
- *...**the smallest in terms of production**, Chateau Petrus, which costs...*
- *...at least one member of the court, Judge Robert Mayer, a former civil litigator, who served...*
- *...the microphone invented by my grandfather, Emile Berliner, which had...*
- *...its main product, bleached paperboard, which goes...*

3 Deciding Clause Boundaries

Determining where a relative clause ends is not always trivial. Non-restrictive relative clauses can extend to the end of the sentence or end with a comma. However, there might be commas internal to the clause so that at each comma after the clause starts, a decision needs to be made on whether the clause ends or not. We devised a set of heuristics for making this decision based on a manual examination of 290 non-restrictive *who* clauses and 846 non-restrictive *which* clauses in our training set derived from the Penn WSJ Treebank. These heuristics are encoded in algorithm 2 below.

Algorithm 2

1. LET n be the number of commas between “*{who|which}*” and the end of the sentence.
2. IF $n = 0$ THEN clause extends till the end of sentence

3. IF $n > 0$ THEN a decision needs to be made as follows
4. FOR each comma (scanning from left to right) DO
 - (a) IF followed by an Appositive¹
THEN INTERNAL comma
 - (b) IF followed by a Verb Group THEN
IF the verb has POS “VB{N|G}”
THEN INTERNAL comma
ELSE END of clause
 - (c) IF an implicit conjunction of adjectives or adverbs
like “JJ, JJ” or “RB, RB”
THEN INTERNAL clause
 - (d) IF its a *who* clause THEN
IF “, CC who”
THEN END of clause
IF “, {which|when|where}”
THEN INTERNAL comma
 - (e) IF its a *which* clause THEN
IF “, CC which”
THEN END of clause
IF “, {who|when|where}”
THEN INTERNAL comma
5. ELSE by default end clause on first comma

This is essentially a formalisation of various observations we made about the training sets, for example: most “ambiguous” commas were followed by either noun groups (15%) or verb groups (67%); All appositives attached locally within the clause. This is because when a relative clause and an appositive attach to the same noun phrase, the appositive always precedes the relative clause. The verb groups always ended the clause unless they were past participle, present participle or gerund in which case they acted like appositives and attached locally.

3.1 Evaluation

We performed two evaluations of our algorithm. The first evaluation was on the Penn WSJ Treebank corpus. The results are shown below. Clauses are ambiguous if there is at least one comma between the relative pronoun and the end of the sentence.

Data Set	Size	Accuracy ¹	Accuracy ²
Training (<i>who</i>)	290	98.97%	97.84%
Training (<i>which</i>)	846	98.34%	96.80%
Test (<i>who</i>)	236	98.31%	96.75%
Test (<i>which</i>)	696	96.70%	94.20%

¹ For all clauses. ² For only ambiguous clauses.

The second evaluation was against the six systems that participated in the CoNLL-2001 clause identification workshop at ACL-2001 (Daelemans

¹We define appositive to mean anything enclosed in commas, not starting with a conjunction and not containing a verb.

and Zajac, 2001). This comparison, against systems tackling the harder task of identifying all clauses in text, not just relative clauses, illustrates the point that disambiguation algorithms aimed at a specific tasks perform better on that task than more general purpose approaches. The workshop provided training and test sets and the output of six systems on the test set are downloadable at the website. The test set contained 26 non-restrictive *who* clauses and 77 non-restrictive *which* that did not end unambiguously in a full stop. The table below compares our algorithm against the average performance of the six systems (Carreras and Màrquez; Déjean; Hammer-ton; Molina and Pla; Patrick and Goyal; Sang and Erik F. in Daelemans and Zajac (2001)) and the system that performed best (by a large margin) on these 103 examples (Carreras and Màrquez, 2001). The ANLT parser (Briscoe and Carroll, 1995) gave a recall of 77% and precision of 75% on the 26 ambiguous *who* clauses and a recall of 87% and precision of 78% on the 77 ambiguous *which* clauses.

Data Set	Average	Best ¹	Our Algo ²
<i>Who</i> Clauses	28.84%	80.77%	96.15%
<i>Which</i> Clauses	29.65%	80.52%	96.10%

¹ (Carreras and Màrquez, 2001) ² Algorithm 2

4 Conclusions and Future Work

Our results suggest that local context plays an important role in making disambiguation decisions. Machine learning techniques incorporating WordNet semantic categories as features can be used effectively in making some disambiguation decisions, like relative clause attachment. Simple algorithms based on the local context described by part of speech tags, like the one presented for deciding clause boundaries, can be better at disambiguation than sophisticated wide coverage parsers. These techniques, that require only POS tagged text, can be used to aid parse selection. We found that machine learning techniques turned out to be useful where there were many features that were potentially relevant for disambiguation and a methodology was required to determine their relative importance. When features consistently selected one way or the other, sequential decision based algorithms that took minimal effort to devise sufficed.

The results presented in this paper pertain to the *analysis* and *transformation* stages suggested by

Chandrasekar et al. (1996). In practice, we require a third stage—*generation*. An important realisation issue arises in the generation stage. When splitting a sentence into two by dis-embedding a relative clause, the referent noun phrase gets duplicated, occurring once in each simplified sentence. We need to generate a referring expression the second time, as duplicating the whole noun phrase can make the text stilted. Other generation issues include determiner choice for the referring expression and deciding the order in which to output the simplified sentences.

Other future work involves evaluating the whole dis-embedding relative clauses task by checking how many sentences containing relative clauses get simplified correctly.

Acknowledgements

We'd like to thank Ted Briscoe for many fruitful discussions about our work and for answering all our queries about the ANLT parser, John Carroll for introducing us to the text simplification problem and sharing both insights and data from the PSET project with us and four anonymous referees for providing constructive feedback on this paper.

References

- Ted Briscoe and John Carroll. 1993. Generalised probabilistic LR parsing for unification-based grammars. *Computational Linguistics*, 19(1):25–60.
- Ted Briscoe and John Carroll. 1995. Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels. In *Proceedings of the ACL/SIGPARSE 4th International Workshop on Parsing Technologies, Prague / Karlovy Vary, Czech Republic*, pages 48–58.
- Andrew J. Carlson, Chad M. Cumby, Jeff L. Rosen, and Dan Roth. 1999. The SNoW learning architecture. Technical report, Tech. Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May.
- Xavier Carreras and Lufís Márquez. 2001. Boosting trees for clause splitting. In Walter Daelemans and Rémi Zajac, editors, *Proceedings of CoNLL-2001*, pages 73–75. Toulouse, France.
- John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of English newspaper text to assist aphasic readers. In *Proceedings of AAAI98 Workshop on Integrating Artificial Intelligence and Assistive Technology, Madison, Wisconsin*.
- 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*.
- David Carter. 1987. *Interpreting Anaphors in Natural Language Texts*. Ellis Horwood, Chichester.
- 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.
- Walter Daelemans and Rémi Zajac, editors. 2001. *Proceedings of CoNLL-2001*. Toulouse, France.
- Siobhan Devlin. 1999. Simplifying natural language for aphasic readers. Technical report, Ph.D. thesis, University of Sunderland, UK.
- Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.
- Barbara J. Grosz. 1978. Discourse analysis. In D.E. Walker, editor, *Understanding Spoken Language*, pages 235–268. North-Holland, New York.
- Jerry R. Hobbs. 1986. Resolving pronoun references. In Barbara J. Grosz, Karen Sparck-Jones, and Bonnie L. Webber, editors, *Readings in Natural Language Processing*, pages 339–352. Morgan Kaufmann, Los Altos, California.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large natural language corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.
- George A. Miller, Richard Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine Miller. 1993. Five Papers on WordNet. Technical report, Princeton University, Princeton, N.J.
- Geoffrey Nunberg. 1990. *The Linguistics of Punctuation*. Number 18 in CSLI Lecture Notes. Stanford University Press.
- Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.
- Bonnie L. Webber. 1978. A formal approach to discourse anaphora. Technical Report 3761, Cambridge, Massachusetts.