Choosing a Parser for Anaphora Resolution

Judita Preiss

Computer Laboratory
JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
Judita.Preiss@cl.cam.ac.uk

Abstract

We investigate the performance changes in the Lappin and Leass (1994) anaphora resolution algorithm when we switch the parser. We compare anaphora resolution results obtained using the Briscoe and Carroll (1993) probabilistic LR parser, the maximum entropy based Charniak (2000) parser, and two versions of the Collins (1997) parser.

1. Introduction

The Lappin and Leass (1994) anaphora resolution algorithm requires its input to be fully parsed, and its performance is thus affected by which parser is chosen to do this preprocessing step. In this paper, we investigate to what extent the performance varies by switching between the Briscoe and Carroll (1993) probabilistic LR parser,[1] the Charniak (2000) parser[2] and two versions of the Collins (1997) parser.[3]

Anaphora resolution algorithms rarely use the same tools, often they are not even evaluated on the same anaphorically resolved corpus. These factors mean that it is usually difficult to compare performances of anaphora resolution algorithms. For example, consider the results of the Kennedy and Boguraev algorithm in the authors' paper (Kennedy and Boguraev, 1996), in a paper by Barbu and Mitkov (2001) and in our previous work (Preiss, 2002).

We use the four parsers in their entirety, including the recommended taggers etc. wherever this is appropriate. Thus we end up with a comparison involving complete changes in the parser component, but all other components of the anaphora resolution algorithm stay constant.

In Section 2. we describe the Lappin and Leass anaphora resolution algorithm. Section 3. outlines the four parsers, with Section 4. introducing our grammatical role extraction system. An evaluation, including a description of the evaluation corpus can be found in Section 5. We draw our conclusions in Section 6.

This work was supported by UK EPSRC project GR/N364 62/93: 'Robust Accurate Statistical Parsing'.

[1]Available from `http://www.cogs.susx.ac.uk/lab/nlp/rasp/`

[2]Available from `http://www.cs.brown.edu/people/ec/`

[3]Available from `http://www.cis.upenn.edu/~mcollins/home.html`

Factor	Weight
Sentence recency	100
Subject emphasis	80
Existential emphasis	70
Accusative emphasis	50
Indirect object/oblique	40
Head noun emphasis	80
Non-adverbial emphasis	50

Table 1: Salience weights

2. Lappin and Leass (1994) Anaphora Resolution Algorithm

We re-implement a non-probabilistic algorithm due to Lappin and Leass (1994). For each pronoun, this algorithm uses syntactic criteria to rule out noun phrases that cannot possibly corefer with it. An antecedent is then chosen according to a ranking based on salience weights.

For all pronouns, noun phrases cannot corefer if they have incompatible agreement features. Pronouns are split into two classes, lexical (reflexives and reciprocals) and non-lexical anaphors. There are additional syntactic filters for both of the two types of anaphors.

Candidates which remain after filtering are ranked according to their salience. A salience value corresponding to a weighted sum of the relevant feature weights (summarized in Table 1) is computed. If we consider the sentence *John walks*, the salience of John will be:

$$\begin{aligned} \text{sal}(\text{John}) &= w_{\text{sent}} + w_{\text{subj}} + w_{\text{head}} + w_{\text{non-adv}} \\ &= 100 + 80 + 80 + 50 \\ &= 210 \end{aligned}$$

Parser	Corpus	LR	LP	CB	0CB	2CB
$\leq$ 40 words						
Charniak	WSJ	90.1	90.1	0.74	70.1	89.6
Coll 1	WSJ	87.52	87.92	0.96	64.86	86.19
Coll 2	WSJ	88.07	88.35	0.95	65.84	86.64
$\leq$ 100 words						
Charniak	WSJ	89.6	89.5	0.88	67.6	87.7
Coll 1	WSJ	87.01	87.41	1.11	62.17	83.86
Coll 2	WSJ	87.60	87.89	1.09	63.20	84.60
B&C evaluation						
B & C	Susanne	74.0	73.0	1.03	59.6	–

Table 2: Summary of Published Results

The weights are scaled by a factor of $\left(\frac{1}{2}\right)^s$ where s is the distance (number of sentences) of the candidate from the pronoun.

We select the candidate with the highest salience to be the antecedent.

3. Parser Descriptions

In this experiment we use four state of the art probabilistic parsers, freely downloadable from the Internet. They were given the same tokenization for the test corpus[4] and their output was post-processed into a consistent format.

In the following sections we will list the parsers' grammar, parsing algorithm, tagger and training corpus. This information is relevant to their use in the present anaphora resolution experiment, and will help when we interpret the results of the experiment.

3.1. Briscoe and Carroll (1993)

Grammar: unification-based grammar of part of speech and punctuation labels.

Parsing algorithm: LR parser.

Tagger: Acquillex HMM tagger (using the CLAWS-II labels) (Elworthy, 1994).

Training corpus: the Susanne corpus (Sampson, 1995).

[4]However, Charniak's parser contains a tokenization component, so some words were further (wrongly) tokenized. For example, the text contained the word *Sánchez* instead of *Sánchez*. The Charniak parser created three words out of this: *S&aacute*, *;*, and *nchez*. The tokenization of the words was also sometimes already pre-processed for the Briscoe and Carroll parser, for example *so that* is tokenized as *so<blank>that* which was used in the other three parsers too.

3.2. Charniak (2000)

Grammar: Generative parsing model using a Markov grammar.

Parsing algorithm: Standard bottom-up best-first chart parser.

Tagger: comes with parser.

Training corpus: sections 2–21 of the Penn Wall Street Journal (WSJ) tree-bank (Marcus et al., 1993).

3.3. Collins (1997)

Grammar: Both parsers are generative models of lexicalized context-free grammar.

Parsing algorithm: CKY-style dynamic programming chart parser.

Tagger: Ratnaparkhi (1996).[5]

Training corpus: sections 2–21 of the Penn Treebank.

For our experiment we used two versions of the Collins parser, with different parsing models:

- Collins 1: a sequence of decisions is used to generate the parse tree in a top-down fashion.
- Collins 2: extends Collins 1 by introducing subcategorization frames, adjunct/complement distinction.

[5]Available from `http://www.cis.upenn.edu/~adwait/`

Sentence: John gave Mary the book.

Full parse:

```
(|T/txt-sc1/---|
 (|S/np_vp| (|NP/n1_name/-| (|N1/n| |John:1_NP1|))
  (|V/np_np| |give+ed:2_VVD| (|NP/n1_name/-| (|N1/n| |Mary:3_NP1|))
   (|NP/det_n| |the:4_AT| (|N1/n| |book:5_NN1|)))))
```

Grammatical relations:

```
(|ncsubj| |give+ed:2:2_VVD| |John:1:1_NP1| _)
(|dobj| |give+ed:2:2_VVD| |Mary:3:3_NP1| _)
(|obj2| |give+ed:2:2_VVD| |book:5:5_NN1|)
(|detmod| _ |book:5:5_NN1| |the:4:4_AT|)
```

Figure 1: Briscoe and Carroll output for *John gave Mary the book*

Sentence: John gave Mary the book

Full parse:

```
(S1 (S (NP (NNP John))
     (VP (VBD gave) (NP (NNP Mary)) (NP (DT the) (NN book)))
     (. .)))
```

Figure 2: Charniak output for *John gave Mary the book*

3.4. Summary of Published Results

Performance results, taken from the authors' original papers, are presented in Table 2. We present the following information:

Labelled Precision (LP) = $\frac{\textit{no. of correct constituents in proposed parse}}{\textit{number of constituents in proposed parse}}$

Labelled Recall (LR) = $\frac{\textit{no. of correct constituents in proposed parse}}{\textit{number of constituents in treebank parse}}$

Crossing Brackets (CB) = number of constituents which violate constituent boundaries with a constituent in the treebank parse

The Charniak and Collins parsers output Penn Treebank style trees, and are trained on sections 2–21 of the Penn Wall Street Journal treebank. The reported results are found from evaluating on section 23. The Briscoe and Carroll parser does not output a Penn Treebank style tree, and is therefore evaluated on 250 sentences bracketed in the Susanne style.

4. Grammatical Role Extraction

The Lappin and Leass anaphora resolution algorithm requires information about the suggested candidates' grammatical roles. In this section, we explain how we extracted this grammatical information from the output of the parsers we used. Dependency information of this type is already provided by the Briscoe and Carroll parser, which for example for the sentence *John gave Mary the book* produces the output shown in Figure 1.

However, the same information is not available for the Charniak or Collins' parsers. The parse for the same sentence (obtained from Charniak's parser) is shown in Figure 2.

We therefore a program which identifies the necessary grammatical roles in a Penn Treebank style parse by walking over the constructed tree and applying a set of hand-crafted rules. As the distinction between a subject and object is often the difference between choosing the correct candidate or not, we required our system to have high precision. We present a comparison of the grammatical roles extracted by our program to those extracted by the Briscoe and Carroll parser. The extraction software was evaluated on the 500 sentence grammatical relation evaluation corpus described in the work of Carroll et al. (1999).[6] The

[6]This evaluation corpus is available from

		B&C	Charniak	Collins 1	Collins 2
subj	precision (%)	84	91	89	90
	recall (%)	88	85	80	83
	F-measure	0.86	0.88	0.84	0.86
dobj	precision (%)	86	82	83	83
	recall (%)	84	67	62	55
	F-measure	0.85	0.74	0.71	0.66
inobj	precision (%)	39	60	50	50
	recall (%)	84	32	32	32
	F-measure	0.53	0.41	0.39	0.39

Table 3: Grammatical Relation Evaluation

results of this comparison are presented in Table 3.

5. Evaluation

5.1. Evaluation Corpus

The lack of anaphorically resolved corpora (reported by (Mitkov, 1999)) led us to manually annotate our own. We chose the initial 2400 sentences of the BNC, which we split into five roughly equal sized texts for a 5-fold cross validation. The full details of the annotation can be found in our earlier work (Preiss, 2002).

5.2. Results

We present the results of the anaphora resolution algorithm experiment in Table 4. The first five lines tell us the number of sentences, pronouns and the performance results in each of the five cross-validation texts. The table also includes the average performance and variance values. The baseline column describes the most recent subject baseline performance with the Briscoe and Carroll parser.

We also carried out a one-tailed t-test comparison of the performances, presented in Table 5. We illustrate how to interpret the table using the last line:

	1	2	3	4	5
Collins 2	–	70%	70%	–	

This states that the anaphora resolution algorithm using Collins 2 does not outperform the Briscoe and Carroll parser (columns 2 and 5). However, it outperforms both the Charniak and Collins 1 parser components with confidence 70% (columns 3 and 4).

5.3. Interpretation of the Results

The results obtained are surprising: intuitively, we would expect the performances of the anaphora resolution algorithm to mirror the performances of the parsers (see Table 2). In particular, we would expect the performance with the Charniak parser to be better than the performance with the Collins parsers. In this section, we suggest some possible reasons for the difference.

Already in Section 4., we can see that the performance of our complement to verb dependency results are lower than those reported by Collins in his thesis (Collins, 1999). He obtains 93.76% precision with 92.96% recall on the Penn Treebank. Now adding the observation that the performance of the anaphora resolution algorithm with various parsers does not match the parsers' relative performance, suggests to us that we may be dealing with what Gildea (2001) describes as "strong corpus effects with parsers". Due to being trained and tested on different corpora (and especially due to the fact that the test corpus is not as nicely uniform as the WSJ), we may be observing deviations from the cited parser performance.

It is however very interesting to note that the differences in performance of the anaphora resolution algorithm are not great. The change in performance between using the Briscoe and Carroll parser versus using the Charniak parser is only three percent. This may imply that it is difficult for different parsers to introduce much new information that would benefit the Lappin and Leass anaphora resolution algorithm.

The addition of adjunct information generated during parsing (as is the case in the Collins 2 parser), improves the performance of the anaphora resolution system only by one percent. The more detailed structure of the Briscoe and Carroll parse also does not yield great improvements to the accuracy of the anaphora resolution system.

It might be argued that we are not using the parsers in an optimal manner and if they were used differently we may obtain much better results with anaphora resolution. The following three problematic areas were noted during development:

`http://www.cogs.susx.ac.uk/lab/nlp/carroll/greval.html`

File	Sents	Prons	B&C	Collins 2	Collins 1	Charniak	Baseline
1	495	93	70%	68%	71%	68%	31%
2	489	44	61%	66%	55%	57%	49%
3	487	54	61%	57%	61%	61%	45%
4	482	198	64%	70%	66%	62%	44%
5	480	224	63%	55%	55%	58%	36%
Average	487	123	64%	63%	62%	61%	41
Variance	28	5545	10	34	41	14	43

Table 4: Percentage Accuracy with Various Parsers

	B&C	Collins 2	Collins 1	Charniak
B&C	–	60%	80%	97.5%
Collins 2	–	–	70%	70%
Collins 1	–	–	–	55%
Charniak	–	–	–	–

Table 5: One Tailed t-test Significance Results

1. As noted in Footnote 3., Charniak's algorithm may be decreasing it's own performance by making use of its tokenizer. For example, the word *Sánchez* is split into three words. This may account for the slightly lower performance of Charniak's model compared with Collins' models.

2. Both Charniak and Collins models suffer from the combined prepositions out<blank>of, which often get tagged as nouns rather than prepositions.

3. All parsers are losing out by not being run on a corpus in the same style as the one they were trained on: WSJ for Charniak and Collins, the Susanne corpus for Briscoe and Carroll.

6. Conclusion

From our preliminary investigation, it would appear that the difference in anaphora resolution performance with different state of the art parsers remains almost constant. Since the parser have very different performance figures on their respective evaluation domains, our work supports a conclusion of Gildea (2001), that standard evaluation metrics of parser performance may not be indicative of a parser's performance in different situations.

Acknowledgements

I would like to thank Ted Briscoe for advice and his insight into parsers. I have also had many productive discussions with Joe Hurd, Anna Korhonen and Stephen Clark.

7. References

C. Barbu and R. Mitkov. 2001. Evaluation tool for rule-based anaphora resolution methods. In *Proceedings of the 39th ACL/10th EACL*, pages 34–41.

E. J. Briscoe and J. Carroll. 1993. Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–60.

J. Carroll, G. Minnen, and E. J. Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of the EACL'99 Workshop on Linguistically Interpreted corpora*, pages 35–41.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL-2000*.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the ACL (jointly with the 8th Conference of the EACL)*, pages 16–23.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

D. Elworthy. 1994. Does baum-welch re-estimation help taggers? In *Proceedings of the 4th Conference on Applied NLP*, pages 53–58.

D. Gildea. 2001. Corpus variation and parser performance. In *Proceedings of 2001 Conference on Empirical Methods in Natural Language Processing*, pages 167–202.

C. Kennedy and B. Boguraev. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *Proceedings of the 16th International*

Conference on Computational Linguistics (COLING'96), pages 113–118.

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

M. Marcus, R. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguisitcs*, 19(2):313–330.

R. Mitkov. 1999. Anaphora resolution: the state of the art. Technical report, University of Wolverhampton.

J. Preiss. 2002. Anaphora resolution with memory based learning. In *Proceedings of CLUK 5*, pages 1–9.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

G. Sampson. 1995. *English for the computer*. Oxford University Press.

Sentence detector POS- tagging Partial parsing. Anaphora Resolution. Question analysis Question type Definition terms. Keywords.Â These tools compose the Slot Unification Parser for Anaphora Resolution (SUPAR) described in FerrÃ¡ndez (1999, 1998). SUPAR's architecture consists of three independent modules that interact with one other. These modules are lexical analysis, syntactic analysis, and a resolution module for NLP problems such as anaphora resolution. Queries and documents are pre-processed before entering question analysis and answer selection modules. Queries pre-processing consists on part-of-speech-tagging terms and parsing.