Parser Evaluation and the BNC: Evaluating 4 constituency parsers with 3 metrics

Jennifer Foster and Josef van Genabith

National Centre for Language Technology
Dublin City University
Ireland
jfoster,josef@computing.dcu.ie

Abstract

We evaluate discriminative parse reranking and parser self-training on a new English test set using four versions of the Charniak parser and a variety of parser evaluation metrics. The new test set consists of 1,000 hand-corrected British National Corpus parse trees. We directly evaluate parser output using both the Parseval and the Leaf Ancestor metrics. We also convert the hand-corrected and parser output phrase structure trees to dependency trees using a state-of-the-art functional tag labeller and constituent-to-dependency conversion tool, and then calculate label accuracy, unlabelled attachment and labelled attachment scores over the dependency structures. We find that reranking leads to a performance improvement on the new test set (albeit a modest one). We find that self-training using BNC data leads to significantly better results. However, it is not clear how effective self-training is when the training material comes from the North American News Corpus.

1. Introduction

We evaluate state-of-the-art constituency parsing techniques using four different versions of the Charniak parser and a new English test set consisting of 1,000 sentences taken from the British National Corpus. The parsers are evaluated using three metrics: the oft-employed Parseval metric, the less well known Leaf-Ancestor metric, and a dependency evaluation which relies on an automatic functional tag labeller and a constituency-to-dependency conversion program to convert the parser output and gold standard phrase structure trees to dependency trees. We present the evaluation results and highlight some areas where there is room for improvement.

The paper is organised as follows: in Section 2., we present the parsers which will be evaluated. In Section 3., we describe the new test data. In Section 4., we present the evaluation results. In Section 5., we present and analyse a more detailed breakdown of evaluation results, before summarizing and concluding in Section 6..

2. The Parsers

We evaluate four different versions of the Charniak parser, a constituency parser with state-of-the-art performance on the standard English test set, Section 23 of the Wall Street Journal section of the Penn Treebank (WSJ23) (Marcus et al., 1994). The first parser (*parser1*) is Charniak's lexicalized history-based generative statistical parser which achieves a Parseval f-score of 89.1% on WSJ23 (Charniak, 2000). The second parser (*parser2*) extends the first parser by incorporating a discriminative reranker which uses features ranging over the entire parse tree to re-order the *n*-best parses returned by *parser1* (Charniak and Johnson, 2005). The reranking parser achieves an f-score of 91.3% on WSJ23, a significant improvement over the first-stage parser.

The third parser (*parser3*) is the self-trained parser reported in McClosky et al. (2006a; 2006b): 1.75 million sentences from the North American News Corpus (NANC) are parsed with *parser2*, and *parser1* is retrained on a combination of its original training material (Sections 2-21 of the WSJ) and the NANC trees produced by *parser2*. The resulting parser, *parser3*, is the re-trained *parser1* combined with the discriminative reranker and it achieves an f-score of 92.1% on WSJ23. To obtain the fourth parser (*parser4*) we repeat the self-training procedure used to produce *parser3*, but we use sentences from the BNC instead of the NANC (Foster et al., 2007). The f-score of *parser4* on WSJ23 is 91.7%. Table 1 summarises the results for all four parsers on WSJ23.

	parser1	*parser2*	*parser3*	*parser4*
F-Score	89.1	91.3	92.1	91.7

Table 1: Parseval Results on WSJ23

3. BNC Test Set

The new English test set consists of 1,000 sentences taken from the British National Corpus (BNC) (Burnard, 2000). The BNC is a one hundred million word balanced corpus of British English from the late twentieth century. Ninety per cent of it is written text, and the remaining 10% consists of transcribed spontaneous and scripted spoken language.

The BNC sentences that are in the test set are not chosen completely at random. Each sentence in the test set has the property of containing a word which appears as a verb in the BNC but not in the usual training sections of the Wall Street Journal section of the Penn Treebank (WSJ02-21). Sentences were chosen in this way so that the resulting test set would be a difficult one for WSJ-trained parsers. Approximately 6% of the BNC test set consists of "non-standard" text such as spoken language, captions, headlines, lines from poems, etc. Examples are given in Table 2.

In order to produce the gold standard parse trees, the test sentences were manually parsed by one annotator, using as references the Penn Treebank trees themselves and the Penn Treebank bracketing guidelines (Bies et al., 1995).

Text Type	#	Example
Highlighted	34	*Podvig also prominent in the Crime and Punishment notebooks, gets relegated in the final text to the Epilogue where it is seen at its simplest in the mitigating circumstance that the murderer is discovered at his trial to have burnt himself rescuing two little children from a blazing house.*
Dramatic	21	*Tommy Johnson dribbled past the Oxford keeper, shot towards an empty net but up popped Matt Elliott to clear off the line.*
Quote	10	*I know that's not your fault but all the same, God damn you*
Spoken	10	*The seconder of formally seconded*
Poem	9	*Groggily somersaulting to get airborne*
List Item	8	*If you're really this thirsty, drink something non-alcoholic to quench thirst*
Caption	4	*Community Personified*
Headline	2	*Drunk priest is nicked driving to a funeral*

Table 2: Some examples of non-standard text from BNC test set sentences

When the two references did not agree, the guidelines took precedence over the Penn Treebank trees. Due to time constraints, the annotator did not mark functional tags or traces. The annotator made two passes through the data, and annotated between 10 and 20 sentences per hour. Difficult parsing decisions were documented. Some pre-processing was carried on the BNC test sentences to ensure that they were tokenized in a similar way to Penn Treebank sentences (see (Wagner et al., 2007) for details).

4. Parser Evaluation

4.1. Parseval Evaluation

The Parseval metric (Black et al., 1991) calculates precision and recall over the constituents in a parse tree. According, to the stronger version of the metric, labelled Parseval, a constituent in a parser output tree is correct if there is a constituent in the corresponding gold parse tree which dominates the same sequence of terminal symbols *and* has the same label. The weaker version, unlabelled Parseval, considers a constituent to be correct if there is a constituent in the gold parse tree which dominates the same sequence of terminal symbols. We use the stricter labelled Parseval measure. In order to separate the evaluation of parsing and part-of-speech tagging, the Parseval metric does not calculate the accuracy of pre-terminal constituents, e.g. *(NN man)*. Precision is the number of correct constituents produced by the parser divided by the total number of constituents produced by the parser. Recall is the number of correct constituents produced by the parser divided by the total number of constituents in the set of gold standard parse trees. The f-score is the harmonic mean of precision and recall.

The Parseval results for the four versions of the Charniak parser are shown in Table 3. McClosky et al. (2006b) report that *parser2* achieves a labelled f-score of 85.2% on sentences from Brown Corpus. The performance for the same parser is worse for the BNC — this is not unexpected, not only because the BNC contains sentences from a wide variety of text genres but also because the BNC test set is a difficult one. As with the WSJ23 test set, each successive version of the parser improves performance, with *parser4* achieving the most significant improvement.

The significant improvement for *parser4* demonstrates that self-training on in-domain data has the potential to be used to adapt a parser to a new domain. McCloskey et al.(2006b) claim that self-training a parser on material from the same material as its original training material can be used to carry out domain adaptation, since *parser3*, the parser self-trained on NANC data, performs significantly better on the Brown corpus than *parser2*. This claim is not completely borne out by our results for *parser3* — there is an improvement over *parser1* and *parser2*, but a relatively modest one.

	Precision	Recall	F-Score
parser1	82.5	82.6	**82.5**
parser2	83.5	83.3	**83.4**
parser3	84.0	83.9	**83.9**
parser4	85.6	85.2	**85.4**

Table 3: Parseval Results on BNC Test Set

4.2. Leaf-Ancestor Evaluation

The drawbacks of the Parseval metric have been noted by many (Lin, 1998; Carroll et al., 2002). Some of these criticisms relate to phrase-structure-based evaluation in general, i.e. evaluation based on phrase-structure constituents abstracts away from basic predicate-argument relationships which are important for correctly capturing the semantics of the sentence. Other criticisms relate to the Parseval metric in particular, e.g. it penalises certain attachment errors too harshly, and is too sensitive to the treebank annotation scheme (Rehbein and van Genabith, 2007). Taking these criticisms into account and in order to carry out a balanced evaluation, we employ a second phrase-structure-based evaluation metric, the Leaf-Ancestor metric, and we also perform a dependency-based evaluation (Section 4.3.). The Leaf-Ancestor metric (Sampson and Babarczy, 2002), assigns a score to every word in the test sentence. The score is obtained by comparing the lineage of the word in the parser output tree to the lineage of the same word in the gold parse tree using a Levenshtein or edit-distance measure. The lineage is the sequence of non-terminal symbols from the root node to the word. Sampson and Barbarczy (2002) argue that the Leaf-Ancestor metric is closer to

people's intuitive notion of what constitutes a good parse. Fig 4 shows the Leaf-Ancestor results for the four parsers on the BNC test set. There are slight differences between *parser1*, *parser2* and *parser3*, and, as with the Parseval metric, the greatest improvement is for *parser4*, the version of the parser that has been self-trained on BNC sentences.

	parser1	*parser2*	*parser3*	*parser4*
LA	0.8807	0.8821	0.8810	0.8900

Table 4: Leaf-Ancestor Results on BNC Test Set

4.3. Dependency Evaluation

	UAS	**LabAcc**	**LAS**
parser1	85.8	89.9	82.5
parser2	86.1	90.2	82.8
parser3	86.2	90.7	83.0
parser4	87.4	91.0	84.2

Table 5: Dependency Evaluation Results on BNC Test Set

Proponents of dependency grammar argue that dependency relations between words are a more useful source of information than constituent structure. For parser evaluation, the use of dependencies has also been advocated (Lin, 1998; Kübler and Telljohann, 2002). We can evaluate constituent parsers using a dependency-based evaluation by automatically extracting dependency relationships from constituent structure. The quality of the dependencies produced will depend, not only on the quality of the phrase structure trees, but also on the quality of the automatic constituent-to-dependency conversion procedure. However, any noise introduced by the conversion procedure will also appear in the "gold standard" dependency graphs produced by applying the conversion procedure to the gold standard phrase structure trees.

To extract dependencies, we use the conversion procedure provided by Johansson and Nugues (2007). This is the procedure used in the CONLL 2007 Shared Task on dependency parsing (Nivre et al., 2007), and it improves upon the constituent-to-dependency conversion procedure provided by Yamada and Matsumotot (2003) by using more sophisticated head-finding rules and by making use of functional tags and traces, if present, to resolve long-distance dependencies. Because the BNC gold standard trees have not yet been annotated with functional tags and traces, we apply the machine-learning based functional tag labeller of Chrupala et al. (2007) to both the gold standard trees and the parser output trees before applying the constituent-to-dependency conversion tool. This WSJ-trained labeller takes phrase-structure trees as input and labels the non-terminal symbols with functional tags such as SUBJ, LOC, TMP, etc. It is the best-performing functional tag labeller for WSJ23.

We use the evaluation script provided for the CONLL 2007 Shared Task to compute three scores: the labelled attachment score (LAS) which is the percentage of words with the correct head and dependency label, unlabelled attachment score (UAS) which is the percentage of words assigned the correct head and the labelled accuracy score (LabAcc), which is the percentage of words with the correct dependency label. The results are shown in Fig. 5. The dependency-based evaluation shows similar findings to the Parseval evaluation : each successive parser version improves upon the previous version, with modest improvements for *parser2* and *parser3* and a more significant improvement for the BNC self-trained *parser4*. This improvement manifests itself particularly in the unlabelled attachment score.

5. Error Analysis

Dependency	**F-score**			
	parser1	*parser2*	*parser3*	*parser4*
ADV	63.2	63.7	63.4	64.9
AMOD	67.8	67.6	69.6	70.5
CC	73.4	75.2	71.0	77.0
CLR	72.7	73.4	75.6	75.1
COORD	66.8	68.1	63.5	68.9
DEP	33.3	31.6	33.3	32.4
IOBJ	59.4	55.9	55.9	56.9
LGS	83.5	86.1	86.1	85.3
LOC	68.0	69.0	70.1	73.4
NMOD	89.3	89.7	90.8	91.1
OBJ	83.3	84.0	85.3	85.6
PMOD	92.0	92.5	93.5	93.3
PRD	81.4	80.5	80.9	82.5
PRN	36.6	35.8	33.8	41.8
PRT	65.5	65.9	64.0	65.3
ROOT	88.5	88.9	88.3	90.4
SBJ	90.8	91.3	92.6	93.5
VC	92.3	91.0	91.5	92.1
VMOD	86.8	87.1	87.6	87.8

Table 6: Breakdown by Dependency Type

F-scores for individual dependency relationships are shown in Table 6. A dependency relationship is considered correct if both the attachment and the label are correct. From this breakdown, we can make the following observations:

- All parsers perform relatively badly on the dependency relations: ADV, DEP, IOBJ, PRN, PRT.
- All parsers perform relatively badly on co-ordinate constructions but the NANC self-trained parser, *parser3*, performs worse than the other three parsers. The self-training procedure cannot be blamed for this because the BNC self-trained parser, *parser4*, performs better. This seems to suggest that there are differences in co-ordination phenomena between American newspaper text and the sentences in the BNC.
- All parsers perform well on the frequently occurring dependency relations: NMOD, SUBJ, PMOD.
- The ranking in parser performance

$$parser1 < parser2 < parser3 < parser4$$

holds for the following relations: LOC, NMOD, OBJ, SUBJ, VMOD.

- The BNC self-trained parser, *parser4*, performs better than the other three parsers for all dependency relations apart from the following: CLR, DEP, IOBJ, LGS, PMOD, PRT, VC. The dependency relations ADV, LOC, CC and ROOT seem to be particularly helped by the BNC self-training.

The following 77-word sentence is an example of a sentence which poses a challenge for all four parsers:
The fact is that in the primeval struggle of the jungle , as in the refinements of civilized warfare , we see in progress a great evolutionary armament race – whose results , for defense , are manifested in such devices as speed , alertness , armor , spinescence , burrowing habits , nocturnal habits , poisonous secretions , nauseous taste , and -LRB- camouflage and other kinds of protective coloration -RRB- ; and for offense , in such counter-attributes as speed , surprise , ambush , allurement , visual acuity , claws , teeth , stings , poison fangs , and -LRB- lures -RRB- .
The parsers *parser1* and *parser2* incorrectly analyse the word *claws* as a third person singular verb — encouragingly, both self-trained parsers, *parser3* and *parser4*, have learned to analyse it as a plural noun.

6. Conclusion

We have evaluated four different versions of the Charniak parser on a new 1,000 sentence English test set. The sentences in the test set come from the British National Corpus, and have been chosen in such a way that they tend to differ in theme from the Wall Street Journal sentences of the Penn Treebank. The first version of the parser is the generative, lexicalised parser, the second version combines the first version with a discriminative reranker, and the third and fourth versions employ the technique of self-training – the third version is self-trained on American newspaper text, and the fourth version is self-trained on BNC data. We evaluate the parsers using three different evaluation metrics. The results of the evaluation confirm previous results obtained for WSJ test sets: both re-ranking and self-training improve parser performance. Also, self-training using parser output trees for sentences from the target domain appears to be more effective than self-training using data from the original seed domain.
The new test set is available to other researchers with a BNC license. In the future, we hope to use it to evaluate other parsers, e.g. the Berkeley parser (Petrov et al., 2006).

Acknowledgments We thank the IRCSET Embark Initiative (postdoctoral fellowship P/04/232) and Science Foundation Ireland (Principal Investigator grant 04/IN/1527) for supporting this research. Thanks to Joachim Wagner for providing the BNC pre-processing scripts.

7. References

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II style, Penn Treebank project. Technical Report Tech Report MS-CIS-95-06, University of Pennsylvania, Philadelphia, PA.

Ezra Black, Steve Abney, Dan Flickinger, Claudia Gdaniec, Robert Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Fred Jelinek, Judith Klavans, Mark Liberman, Mitchell Marcus, Salim Roukos, Beatrice Santorini, and Tomek Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the 1991 DARPA Speech and Natural Language Workshop*, pages 306–311.

Lou Burnard. 2000. User reference guide for the British National Corpus. Technical report, Oxford University Computing Services.

John Carroll, Anette Frank, Dekang Lin, Detlef Prescher, and Hans Uszkoreit, editors. 2002. *Proceedings of the "Beyond Parseval - Towards Improved Evaluation Measures for Parsing Systems" Workshop at the 3rd International Conference on Linguistic Resources and Evaluation (LREC-02)*, Las Palmas, Gran Canaria.

Eugene Charniak and Mark Johnson. 2005. Course-to-fine n-best-parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the ACL (ACL-05)*, pages 173–180, Ann Arbor, Michigan, June.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the Annual Meeting of the North American Association for Computational Linguistics (NAACL-00)*, pages 132–139, Seattle, Washington.

Grzegorz Chrupała, Nicolas Stroppa, Josef van Genabith, and Georgiana Dinu. 2007. Better training for function labeling. In *Proceedings of the Recent Advances in Natural Language Processing Conference (RANLP-07)*, pages 133–138, Borovets, Bulgaria.

Jennifer Foster, Joachim Wagner, Djamé Seddah, and Josef van Genabith. 2007. Adapting WSJ-trained parsers to the British National Corpus using in-domain self-training. In *Proceedings of the Tenth International Workshop on Parsing Technologies (IWPT-07)*, pages 33–35, Prague, Czech Republic.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek, and Mare Koit, editors, *Proceedings of NODALIDA 2007*, pages 105–112, Tartu, Estonia.

Sandra Kübler and Heike Telljohann. 2002. Towards a dependency-oriented evaluation for partial parsing. In Carroll et al. (Carroll et al., 2002), pages 9–16.

Dekang Lin. 1998. Dependency-based evaluation of MINIPAR. In John Carroll, editor, *Proceedings of The Evaluation of Parsing Systems Workshop at the 3rd International Conference on Linguistic Resources and Evaluation (LREC)*, Cognitive Science Research Papers 489, pages 48–56. University of Sussex, Brighton, England.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the ARPA Workshop on Human Language Technology*, pages 110–115, Princeton, NJ.

David McClosky, Eugene Charniak, and Mark Johnson. 2006a. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference and North American chapter of the ACL annual meeting (HLT-NAACL-06)*, pages 152–159, New York, June.

David McClosky, Eugene Charniak, and Mark Johnson. 2006b. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (COLING-ACL-06)*, pages 337–344, Sydney, Australia, July.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan Mac Donald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic, June.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia.

Ines Rehbein and Josef van Genabith. 2007. Treebank annotation schemes and parser evaluation for German. In *Proceedings of the Joint EMNLP-CoNLL 2007*, pages 630–639, Prague, Czech Republic.

Geoffrey Sampson and Anna Babarczy. 2002. A test of the leaf-ancestor metric for parse accuracy. In Carroll et al. (Carroll et al., 2002).

Joachim Wagner, Djamé Seddah, Jennifer Foster, and Josef van Genabith. 2007. C-structures and f-structures for the British National Corpus. In *Proceedings of the 12th International Workshop on Lexical Functional Grammar*, Stanford, CA.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT-03)*, pages 195–206, Nancy, France.