Evaluating State-of-the-Art Treebank-style Parsers for Coh-Metrix and Other Learning Technology Environments

Christian F. Hempelmann, Vasile Rus, Arthur C. Graesser, and **Danielle S. McNamara**
Institute for Intelligent Systems
Departments of Computer Science and Psychology
The University of Memphis
Memphis, TN 38120, USA
{chmplmnn, vrus, a-graesser, dsmcnamr}@memphis.edu

Abstract

This paper evaluates a series of freely available, state-of-the-art parsers on a standard benchmark as well as with respect to a set of data relevant for measuring text cohesion. We outline advantages and disadvantages of existing technologies and make recommendations. Our performance report uses traditional measures based on a gold standard as well as novel dimensions for parsing evaluation. To our knowledge this is the first attempt to evaluate parsers accross genres and grade levels for the implementation in learning technology.

1 Introduction

The task of syntactic parsing is valuable to most natural language understanding applications, e.g., anaphora resolution, machine translation, or question answering. Syntactic parsing in its most general definition may be viewed as discovering the underlying syntactic structure of a sentence. The specificities include the types of elements and relations that are retrieved by the parsing process and the way in which they are represented. For example, Treebank-style parsers retrieve a bracketed form that encodes a hierarchical organization (tree) of smaller elements (called phrases), while Grammatical-Relations(GR)-style parsers explicitly output relations together with elements involved in the relation (subj(John,walk)).

The present paper presents an evaluation of parsers for the Coh-Metrix project (Graesser et al., 2004) at the Institute for Intelligent Systems of the University of Memphis. Coh-Metrix is a text-processing tool that provides new methods of automatically assessing text cohesion, readability, and difficulty. In its present form, v1.1, few cohesion measures are based on syntactic information, but its next incarnation, v2.0, will depend more heavily on hierarchical syntactic information. We are developing these measures. Thus, our current goal is to provide the most reliable parser output available for them, while still being able to process larger texts in real time. The usual trade-off between accuracy and speed has to be taken into account.

In the first part of the evaluation, we adopt a constituent-based approach for evaluation, as the output parses are all derived in one way or another from the same data and generate similar, bracketed output. The major goal is to consistently evaluate the freely available state-of-the-art parsers on a standard data set and across genre on corpora typical for learning technology environments. We report parsers' competitiveness along an array of dimensions including performance, robustness, tagging facility, stability, and length of input they can handle.

Next, we briefly address particular types of misparses and mistags in their relation to measures planned for Coh-Metrix 2.0 and assumed to be typical for learning technology applications. Coh-Metrix 2.0 measures that centrally rely on good parses include:
causal and intentional cohesion, for which the main verb and its subject must be identified;

Proceedings of the 2nd Workshop on Building Educational Applications Using NLP,
pages 69–76, Ann Arbor, June 2005.

anaphora resolution, for which the syntactic relations of pronoun and referent must be identified;
temporal cohesion, for which the main verb and its tense/aspect must be identified.

These measures require complex algorithms operating on the cleanest possible sentence parse, as a faulty parse will lead to a cascading error effect.

1.1 Parser Types

While the purpose of this work is not to propose a taxonomy of all available parsers, we consider it necessary to offer a brief overview of the various parser dimensions. Parsers can be classified according to their general approach (hand-built-grammar-based versus statistical), the way rules in parses are built (selective vs. generative), the parsing algorithm they use (LR, chart parser, etc.), type of grammar (unification-based grammars, context-free grammars, lexicalized context-free grammars, etc.), the representation of the output (bracketed, list of relations, etc.), and the type of output itself (phrases vs grammatical relations). Of particular interest to our work are Treebank-style parsers, i.e., parsers producing an output conforming to the Penn Treebank (PTB) annotation guidelines. The PTB project defined a tag set and bracketed form to represent syntactic trees that became a standard for parsers developed/trained on PTB. It also produced a treebank, a collection of hand-annotated texts with syntactic information.

Given the large number of dimensions along which parsers can be distinguished, an evaluation framework that would provide both parser-specific (to understand the strength of different technologies) and parser-independent (to be able to compare different parsers) performance figures is desirable and commonly used in the literature.

1.2 General Parser Evaluation Methods

Evaluation methods can be broadly divided into non-corpus- and corpus-based methods with the latter subdivided into unannotated and annotated corpus-based methods (Carroll et al., 1999). The non-corpus method simply lists linguistic constructions covered by the parser/grammar. It is well-suited for hand-built grammars because during the construction phase the covered cases can be recorded. However, it has problems with capturing complexities occuring from the interaction of covered cases.

The most widely used corpus-based evaluation methods are: (1) the constituent-based (phrase structure) method, and (2) the dependency/GR-based method. The former has its roots in the Grammar Evaluation Interest Group (GEIG) scheme (Grishman et al., 1992) developed to compare parsers with different underlying grammatical formalisms. It promoted the use of phrase-structure bracketed information and defined Precision, Recall, and Crossing Brackets measures. The GEIG measures were extended later to constituent information (bracketing information plus label) and have since become the standard for reporting automated syntactic parsing performance. Among the advantages of constituent-based evaluation are generality (less parser specificity) and fine grain size of the measures. On the other hand, the measures of the method are weaker than exact sentence measures (full identity), and it is not clear if they properly measure how well a parser identifies the true structure of a sentence. Many phrase boundary mismatches spawn from differences between parsers/grammars and corpus annotation schemes (Lin, 1995). Usually, treebanks are constructed with respect to informal guidelines. Annotators often interpret them differently leading to a large number of different structural configurations.

There are two major approaches to evaluate parsers using the constituent-based method. On the one hand, there is the expert-only approach in which an expert looks at the output of a parser, counts errors, and reports different measures. We use a variant of this approach for the directed parser evaluation (see next section). Using a gold standard, on the other hand, is a method that can be automated to a higher degree. It replaces the counting part of the former method with a software system that compares the output of the parser to the gold standard,

highly accurate data, manually parsed − or automatically parsed and manually corrected − by human experts. The latter approach is more useful for scaling up evaluations to large collections of data while the expert-only approach is more flexible, allowing for evaluation of parsers from new perspectives and with a view to special applications, e.g., in learning technology environments.

In the first part of this work we use the gold standard approach for parser evaluation. The evaluation is done from two different points of view. First, we offer a uniform evaluation for the parsers on section 23 from the Wall Street Journal (WSJ) section of PTB, the community norm for reporting parser performance. The goal of this first evaluation is to offer a good estimation of the parsers when evaluated in identical environments (same configuration parameters for the evaluator software). We also observe the following features which are extremely important for using the parsers in large-scale text processing and to embed them as components in larger systems.

Self-tagging: whether or not the parser does tagging itself. It is advantageous to take in raw text since it eliminates the need for extra modules.

Performance: if the performance is in the mid and upper 80th percentiles.

Long sentences: the ability of the parser to handle sentences longer than 40 words.

Robustness: relates to the property of a parser to handle any type of input sentence and return a reasonable output for it and not an empty line or some other useless output.

Second, we evaluate the parsers on narrative and expository texts to study their performance across the two genres. This second evaluation step will provide additional important results for learning technology projects. We use *evalb* (http://nlp.cs.nyu.edu/evalb/) to evaluate the bracketing performance of the output of a parser against a gold standard. The software evaluator reports numerous measures of which we only report the two most important: labelled precision (LR), labelled recall (LR) which are discussed in more detail below.

1.3 Directed Parser Evaluation Method

For the third step of this evaluation we looked for specific problems that will affect Coh-Metrix 2.0, and presumably learning technology applications in general, with a view to amending them by postprocessing the parser output. The following four classes of problems in a sentence's parse were distinguished:

None: The parse is generally correct, unambiguous, poses no problem for Coh-Metrix 2.0.

One: There was one minor problem, e.g., a mislabeled terminal or a wrong scope of an adverbial or prepositional phrase (wrong attachment site) that did not affect the overall parse of the sentence, which is therefore still usable for Coh-Metrix 2.0 measures.

Two: There were two or three problems of the type one, or a problem with the tree structure that affected the overall parse of the sentence, but not in a fatal manner, e.g., a wrong phrase boundary, or a mislabelled higher constituent.

Three: There were two or more problems of the type two, or two or more of the type one as well as one or more of the type two, or another fundamental problem that made the parse of the sentence completely useless, unintelligible, e.g., an omitted sentence or a sentence split into two, because a sentence boundary was misidentified.

2 Evaluated Parsers

2.1 Apple Pie

Apple Pie (AP) (Sekine and Grishman, 1995) extracts a grammar from PTB v.2 in which S and NP are the only true non-terminals (the others are included into the right-hand side of S and NP rules). The rules extracted from the PTB have S or NP on the left-hand side and a flat structure on the right-hand side, for instance S → NP VBX JJ. Each such rule has the most common structure in the PTB associated with it, and if the parser uses the rule it will generate its corresponding structure. The parser is a chart parser and factors grammar rules with common prefixes to reduce the number of active nodes. Although the underlying model of the parser is simple, it can't handle sentences over 40 words due to the large variety of linguistic

constructs in the PTB.

2.2 Charniak's Parser

Charniak presents a parser (CP) based on probabilities gathered from the WSJ part of the PTB (Charniak, 1997). It extracts the grammar and probabilities and with a standard context-free chart-parsing mechanism generates a set of possible parses for each sentence retaining the one with the highest probability (probabilities are not computed for all possible parses). The probabilities of an entire tree are computed bottom-up. In (Charniak, 2000), he proposes a generative model based on a Markov-grammar. It uses a standard bottom-up, best-first probabilistic parser to first generate possible parses before ranking them with a probabilistic model.

2.3 Collins's (Bikel's) Parser

Collins's statistical parser (CBP; (Collins, 1997)), improved by Bikel (Bikel, 2004), is based on the probabilities between head-words in parse trees. It explicitly represents the parse probabilities in terms of basic syntactic relationships of these lexical heads. Collins defines a mapping from parse trees to sets of dependencies, on which he defines his statistical model. A set of rules defines a head-child for each node in the tree. The lexical head of the head-child of each node becomes the lexical head of the parent node. Associated with each node is a set of dependencies derived in the following way. For each non-head child, a dependency is added to the set where the dependency is identified by a triplet consisting of the non-head-child non-terminal, the parent non-terminal, and the head-child non-terminal. The parser is a CYK-style dynamic programming chart parser.

2.4 Stanford Parser

The Stanford Parser (SP) is an unlexicalized parser that rivals state-of-the-art lexicalized ones (Klein and Manning, 2003). It uses a context-free grammar with state splits. The parsing algorithm is simpler, the grammar smaller and fewer parameters are needed for the estimation. It uses a CKY chart parser which exhaustively generates all possible parses for a sentence before it selects the highest probability tree. Here we used the default lexicalized version.

3 Experiments and Results

3.1 Text Corpus

We performed experiments on three data sets. First, we chose the norm for large scale parser evaluation, the 2416 sentences of WSJ section 23. Since parsers have different parameters that can be tuned leading to (slightly) different results we first report performance values on the standard data set and then use same parameter settings on the second data set for more reliable comparison.

The second experiment is on a set of three narrative and four expository texts. The gold standard for this second data set was built manually by the authors starting from CP's as well as SP's output on those texts. The four texts used initially are two expository and two narrative texts of reasonable length for detailed evaluation:
The Effects of Heat (SRA Real Science Grade 2 Elementary Science): expository; 52 sentences, 392 words: 7.53 words/sentence;
The Needs of Plants (McGraw-Hill Science): expository; 46 sentences, 458 words: 9.96 words/sentence;
Orlando (Addison Wesley Phonics Take-Home Reader Grade 2): narrative; 65 sentences, 446 words: 6.86 words/sentence;
Moving (McGraw-Hill Reading - TerraNova Test Preparation and Practice - Teachers Edition Grade 3): narrative, 33 sentences, 433 words: 13.12 words/sentence.

An additional set of three texts was chosen from the Touchstone Applied Science Associates, Inc., (TASA) corpus with an average sentence length of 13.06 (overall TASA average) or higher.
Barron17: expository; DRP=75.14 (college grade); 13 sentences, 288 words: 22.15 words/sentence;
Betty03: narrative; DRP=56.92 (5th grade); 14 sentences, 255 words: 18.21 words/sentence;
Olga91: expository; DRP=74.22 (college grade); 12 sentences, 311 words: 25.92 words/sentence.

We also tested all four parsers for speed on a corpus of four texts chosen randomly from the Metametrix corpus of school text books, across high and low grade levels and across narrative and science texts (see Section 3.2.2).
G4: 4th grade narrative text, 1,500 sentences, 18,835 words: 12.56 words/sentence;
G6: 6th grade science text, 1,500 sentences, 18,237 words: 12.16 words/sentence;
G11: 11th grade narrative text, 1,558 sentences, 18,583 words: 11.93 words/sentence;
G12: 12th grade science text, 1,520 sentences, 25,098 words: 16.51 words/sentence.

3.2 General Parser Evaluation Results

3.2.1 Accuracy

The parameters file we used for *evalb* was the standard one that comes with the package. Some parsers are not robust, meaning that for some input they do not output anything, leading to empty lines that are not handled by the evaluator. Those parses had to be "aligned" with the gold standard files so that empty lines are eliminated from the output file together with their peers in the corresponding gold standard files.

In Table 1 we report the performance values on Section 23 of WSJ. Table 2 shows the results for our own corpus. The table gives the average values of two test runs, one against the SP-based gold standard, the other against the CP-based gold standard, to counterbalance the bias of the standards. Note that CP and SP possibly still score high because of this bias. However, CBP is clearly a contender despite the bias, while AP is not.[1] The reported metrics are Labelled Precision (LP) and Labelled Recall (LR). Let us denote by a the number of correct phrases in the output from a parser for a sentence, by b the number of incorrect phrases in the output and by c the number of phrases in the gold standard for the same sentence. LP is defined as $a/(a+b)$ and LR is defined as a/c. A summary of the other dimensions of the evaluation is offered in Table 3. A stability dimension is not reported because we were not able to find a bullet-proof parser so far, but we must recognize that some parsers are significantly more stable than others, namely CP and CBP. In terms of resources needed, the parsers are comparable, except for AP which uses less memory and processing time. The LP/LR of AP is significantly lower, partly due to its outputting partial trees for longer sentences. Overall, CP offers the best performance.

Note in Table 1 that CP's tagging accuracy is worst among the three top parsers but still delivers best overall parsing results. This means that its parsing-only performance is slighstly better than the numbers in the table indicate. The numbers actually represent the tagging and parsing accuracy of the tested parsing systems. Nevertheless, this is what we would most likely want to know since one would prefer to input raw text as opposed to tagged text. If more finely grained comparisons of only the parsing aspects of the parsers are required, perfect tags extracted from PTB must be provided to measure performance.

Table 4 shows average measures for each of the parsers on the PTB and seven expository and narrative texts in the second column and for expository and narrative in the fourth column. The third and fifth columns contain standard deviations for the previous columns, respectively. Here too, CP shows the best result.

3.2.2 Speed

All parsers ran on the same Linux Debian machine: P4 at 3.4GHz with 1.0GB of RAM.[2] AP's and SP's high speeds can be explained to a large degree by their skipping longer sentences, the very ones that lead to the longer times for the other two candidates. Taking this into account, SP is clearly the fastest, but the large range of processing times need to be heeded.

3.3 Directed Parser Evaluation Results

This section reports the results of expert rating of texts for specific problems (see Section 1.3). The best results are produced by CP with an average of 88.69% output useable for Coh-Metrix 2.0 (Table 6). CP also produces good output

[1] AP's performance is reported for sentences < 40 words in length, 2,250 out of 2,416. SP is also not robust enough and the performance reported is only on 2,094 out of 2,416 sentences in section 23 of WSJ.

[2] Some of the parsers also run under Windows.

Table 1: Accuracy of Parsers.

Parser	Performance(LP/LR/Tagging - %)		
	WSJ 23	Expository	Narrative
Applie Pie	43.71/44.29/90.26	41.63/42.70	42.84/43.84
Charniak's	84.35/88.28/92.58	91.91/93.94	93.74/96.18
Collins/Bikel's	84.97/87.30/93.24	82.08/85.35	67.75/85.19
Stanford	84.41/87.00/95.05	75.38/85.12	62.65/87.56

Table 2: Performance of parsers on the narrative and expository text (average against CP-based and SP-based gold standard).

File	Performance (LR/LP - %)			
	AP	CP	CBP	SP
Heat	48.25/47.59	91.96/93.77	92.47/94.14	92.44/91.85
Plants	41.85/45.89	85.34/88.02	78.24/88.45	81.00/85.62
Orlando	45.82/49.03	85.83/91.88	65.87/93.97	57.75/90.72
Moving	37.77/41.45	88.93/92.74	53.94/91.68	76.56/84.97
Barron17	43.22/42.95	89.74/91.32	80.49/89.32	87.22/86.31
Betty03	46.53/44.67	90.77/90.74	87.95/85.21	74.53/80.91
Olga91	32.29/32.69	77.65/80.04	61.61/75.43	61.65/70.60

Table 3: Evaluation of Parsers with Respect to the Criteria Listed at the Top of Each Column.

Parser	Self-tagging	Performance	Long-sentences	Robustness
AP	Yes	No	No	No
CP	Yes	Yes	Yes	Yes
CBP	Yes	Yes	Yes	Yes
SP	Yes	Yes	No	No

Table 4: Average Performance of Parsers.

Parser	Ave. (LR/LP - %)	S.D. (%)	Ave. on Exp+Nar (LR/LP - %)	S.D. on Exp+Nar (%)
AP	42.73/43.61	1.04/0.82	42.24/43.46	5.59/5.41
CP	90.00/92.80	4.98/4.07	87.17/89.79	4.85/4.66
CBP	78.27/85.95	9.22/1.17	74.36/88.31	14.24/6.51
SP	74.14/86.56	10.93/1.28	75.88/84.42	12.66/7.11

Table 5: Parser Speed in Seconds.

	G4	G6	G11	G12
#sent	619	3336	4976	2215
AP	144	89	144	242
CP	647	499	784	1406
CBP	485	1947	1418	1126
SP	449	391	724	651
Ave.	431	732	768	856

most consistently at a standard deviation over the seven texts of 8.86%. The other three candidates are clearly trailing behing, namely by between 5% (SP) and 11% (AP). The distribution of severe problems is comparable for all parsers.

Table 6: Average Performance of Parsers over all Texts (Directed Evaluation).

	Ave. (%)	S.D. (%)
AP	77.31	15.00
CP	88.69	8.86
CBP	79.82	18.94
SP	83.43	11.42

As expected, longer sentences are more problematic for all parsers, as can be seen in Table 7. No significant trends in performance differences with respect to genre difference, narrative (Orlando, Moving, Betty03) vs. expository texts (Heat, Plants, Barron17, Olga91), were detected (cf. also speed results in Table 5). But we assume that the difference in average sentence length obscures any genre differences in our small sample.

The most common non-fatal problems (type one) involved the well-documented adjunct attachment site issue, in particular for prepositional phrases ((Abney et al., 1999), (Brill and Resnik, 1994), (Collins and Brooks, 1995)) as well as adjectival phrases (Table 8)[3]. Similar misattachment issues for adjuncts are encountered with adverbial phrases, but they were rare

[3]PP = wrong attachment site for a prepositional phrase; ADV = wrong attachment site for an adverbial phrase; cNP = misparsed complex noun phrase; &X = wrong coordination

Table 7: Correlation of Average Performance per Text for all Parsers and Average Sentence Length (Directed Evaluation).

Text	perf. (%)	length (#words)
Heat	92.31	7.54
Plants	90.76	9.96
Orlando	93.46	6.86
Moving	90.91	13.12
Barron17	76.92	22.15
Betty03	71.43	18.21
Olga91	60.42	25.92

in our corpus.

Another common problem are deverbal nouns and denominal verbs, as well as *-ing*/VBG forms. They share surface forms leading to ambiguous part of speech assignments. For many Coh-Metrix 2.0 measures, most obviously temporal cohesion, it is necessary to be able to distinguish gerunds from gerundives and deverbal adjectives and deverbal nouns.

Table 8: Specific Problems by Parser.

	PP	ADV	cNP	&X
AP	13	10	8	9
CP	15	1	2	7
CBP	10	0	0	13
SP	22	6	3	4
Sum	60	17	13	33

Problems with NP misidentification are particularly detrimental in view of the important role of NPs in Coh-Metrix 2.0 measures. This pertains in particular to the mistagging/misparsing of complex NPs and the coordination of NPs. Parses with fatal problems are expected to produce useless results for algorithms operating with them. Wrong coordination is another notorious problem of parsers (cf. (Cremers, 1993), (Grootveld, 1994)). In our corpus we found 33 instances of miscoordination, of which 23 involved NPs. Postprocessing approaches that address these issues are currently under investigation.

4 Conclusion

The paper presented the evaluation of freely available, Treebank-style, parsers. We offered a uniform evaluation for four parsers: Apple Pie, Charniak's, Collins/Bikel's, and the Stanford parser. A novelty of this work is the evaluation of the parsers along new dimensions such as stability and robustness and across genre, in particular narrative and expository. For the latter part we developed a gold standard for narrative and expository texts from the TASA corpus. No significant effect, not already captured by variation in sentence length, could be found here. Another novelty is the evaluation of the parsers with respect to particular error types that are anticipated to be problematic for a given use of the resulting parses. The reader is invited to have a closer look at the figures our tables provide. We lack the space in the present paper to discuss them in more detail. Overall, Charniak's parser emerged as the most succesful candidate of a parser to be integrated where learning technology requires syntactic information from real text in real time.

ACKNOWLEDGEMENTS

This research was funded by Institute for Educations Science Grant IES R3056020018-02. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the IES. We are grateful to Philip M. McCarthy for his assistance in preparing some of our data.

References

S. Abney, R. E. Schapire, and Y. Singer. 1999. Boosting applied to tagging and pp attachment. *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 38–45.

D. M. Bikel. 2004. Intricacies of collins' parsing model. *Computational Linguistics*, 30-4:479–511.

E. Brill and P. Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics.*

J. Carroll, E. Briscoe, and A. Sanfilippo, 1999. *Parser evaluation: current practice*, pages 140–150. EC DG-XIII LRE EAGLES Document EAG-II-EWG-PR.1.

E. Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, AAAI Press/MIT Press, Menlo Park.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the North-American Chapter of Association for Computational Linguistics*, Seattle, Washington.

M. Collins and J. Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, Cambridge.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistic*, Madrid, Spain.

C. Cremers. 1993. *On Parsing Coordination Categorially.* Ph.D. thesis, Leiden University.

A. C. Graesser, D.S. McNamara, M. M. Louwerse, and Z. Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, and Computers*, 36-2:193–202.

R. Grishman, C. MacLeod, and J. . Sterling. 1992. Evaluating parsing strategies using standardized parse files. *In Proceedings of the Third Conference on Applied Natural Language Processing*, pages 156–161.

M. Grootveld. 1994. *Parsing Coordination Generatively.* Ph.D. thesis, Leiden University.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistic*, Sapporo, Japan.

D. Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1420–1427.

A. Ratnaparkhi, J. Renyar, and S. Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Workshop on Human Language Technology.*

S. Sekine and R. Grishman. 1995. A corpus-based probabilistic grammar with only two nonterminals. *Proceedings of the International Workshop on Parsing Technologies*, pages 216–223.