Grammar & Parser Evaluation in the XTAG Project

Srinivas Bangalore[†]**, Anoop Sarkar, Christine Doran and Beth Ann Hockey**

[†]AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932
srini@research.att.com

Institute for Research in Cognitive Science
3401 Walnut Street, Suite 400A
Philadelphia, PA 19104
{anoop,cdoran,beth}@linc.cis.upenn.edu

Abstract

In this paper we discuss several methods used to evaluate the XTAG parser and English grammar. We consider the methods proposed in the literature for grammar and parser evaluation, and give some empirical reasons for electing to use certain methods over others. We propose a general framework for evaluation, which is then used to evaluate the English grammar and parser developed as part of the XTAG project. We describe methods to evaluate and extend the coverage and performance of both the grammar and parser.

1. Introduction

A parsing system can be evaluated along different dimensions ranging from grammatical coverage to average number of parses produced to average number of correct constituents in a parse produced by a system. Given this variety of metrics, there have been a number of proposals for evaluating parsing systems. A comprehensive survey of parsing metrics is provided in [Briscoe *et al.*, 1996]. In this paper we first discuss some of the methods proposed in the literature for grammar and parser evaluation and give some empirical reasons why we chose to use certain methods over others. We then propose a general framework for parser evaluation. Finally, we discuss the use of some of these methods to evaluate and extend the coverage and performance of the grammar and parser developed as part of the XTAG project.

We group the proposals for evaluation into three classes:

1. extrinsic,
2. intrinsic, and
3. comparative

We will discuss each of these approaches in turn.

2. Intrinsic Evaluation

Intrinsic evaluation measures the performance of a parsing system in the context of the framework in which it is developed. This kind of evaluation is applicable to both grammar-based and statistical parsing systems, and helps system developers and maintainers to measure the performance of successive generations of the system. For grammar-based systems, intrinsic evaluation helps identify the shortcomings and weaknesses in the grammar, and provides a direction for productive development of the grammar. For statistical parsers, intrinsic evaluation provides a measure of performance of the underlying statistical model and helps to identify needed improvements. Since the evaluation is performed in the context of the framework that the parsing system is developed in, the metrics used for intrinsic evaluation can be made sensitive to the features and output representations of the parsing system.

Approaches to intrinsic evaluation can be divided into test suite-based and corpus-based methods. The corpus based methods are further divided into those which use annotated data and those which use unannotated data. In the sections that follow, we review each metric and discuss its strengths and weaknesses. Once any of the following evaluation protocols have been performed, the results can be compared with the results of other systems on the same test-suites or corpora. However, unlike the “comparative evaluation” techniques discussed in Section below, it will rarely be the case that such results are directly comparable across systems.

Test suite-based Evaluation In this traditional method of parsing system evaluation, a list of sentences for each syntactic construction that is covered and not covered by the grammar is maintained as a database [Alshawi *et al.*, 1992; Grover *et al.*, 1993; XTAG-Group, 1995; Oepen *et al.*, forthcoming]. The test suite is used to track improvements and verify consistency between successive generations of grammar development in a system. Although this method of evaluation has been mostly used for hand-crafted grammars, it could also be used to track the improvements in performance of statistical parsers from changes in the underlying statistical model. The advantage of this method of evaluation is that it is relatively straightforward and the information provides a direction for improving the system. The disadvantage is that it does not quantify how the performance of a parsing system would scale up when parsing unrestricted text data.

Unannotated Corpus-based Evaluation These methods use unrestricted texts as corpora for evaluating parsing systems. The corpora consist of sentences which are not annotated with any linguistic information. One example of this method is measuring *coverage* which is a measure of the percentage of sentences in the corpus that can be assigned one or more parses by a parsing system [Briscoe and Carroll, 1995; Doran *et al.*, 1994]. It is a weak measure since it does not guarantee that the analysis found is indeed the correct one. A similar metric is that of *Average Parse Base* (APB) [Black *et al.*, 1993a] which is defined as the geometric mean of the number of analysis divided by the number of input tokens in each sentence parsed. This metric provides a method of predicting the average number of parses that a parsing system would produce for a sentence of length *n* tokens. The metric is useful in comparing different versions of a parsing system that is under development, when tested on the same data. The disadvantage is that the metric does not measure the performance of correct analysis provided by the system and systems that have very low coverage would perform very well on this measure. Also, this metric cannot be used for cross-system comparative evaluation.

Annotated Corpus-based Evaluation The following methods use unrestricted texts with linguistic annotation as corpora for evaluating parsing systems. The corpora consists of sentences which are annotated with information such as Part-of-speech tags, constituency labels, subject-verb-object triples. The annotation in the corpus is termed as the *gold standard*. The usefulness of the following methods is dependent on the accuracy and consistency of annotation in the gold standard corpus. One such evaluation method is to measure *tagging accuracy*, i.e accuracy of part-of-speech and/or syntactic tagging [Church, 1988; Karlsson *et al.*, 1994] output by the parser. Other methods try to measure parsing accuracy more directly, for instance, the *zero crossing brackets* measure [Black *et al.*, 1993b] counts the number of constituents that are inconsistent with the gold standard. One can also measure count of the *exact match* of the structure produced by the parse with that in the gold standard. Tree similarity measures, employed by Sampson [1989], uses a variety of metrics including the ratio of rules correctly deployed in a derivation to all rules in that derivation computed from the gold standard. This method is tolerant to certain errors in the gold standard.

3. Extrinsic Evaluation

Extrinsic evaluation is possible when a parsing system is embedded in an application and the parsing system's contribution to the overall performance of the application can be measured. Extrinsic evaluation can be used as an indirect method for comparing parsing systems even if they produce different representations for their outputs as long as the output can be converted into a form usable by the application in which the parser is embedded. The objective of this type of evaluation is an Adequacy Evaluation[1] of a parser. Adequacy Evaluation involves determining the fitness of a system for a purpose. It provides users (application developers) information to determine whether a parser is adequate for their particular task, and if so, whether it is more suited than others. The result of this evaluation is a report which provides comparative information of the parsers and does not necessarily identify the best parser, thus allowing the user to make an informed choice.

4. Comparative Evaluation

A third method of evaluation is *comparative evaluation*. The objective here is to directly compare the performance of parsing systems that use different grammar formalisms and/or different statistical models. Comparative evaluation helps in identifying the relative strengths and weaknesses of the systems and may suggest possibilities of combining different approaches. However, such an evaluation scheme requires a metric that is insensitive to the representational differences in the output produced by different parsers. To achieve this the metric may have to abstract away from individual representations in order to be able to directly compare them. However, as a result of the abstraction process, the strengths of representations of certain parsers might be lost completely.

A metric that was designed with the aim of comparative evaluation of parsing system, Parseval [Harrison *et al.*, 1991] is an alternate mechanism of relaxing exact match to the parse in the gold standard as the success criterion. This scheme utilizes only bracketing information to compute three figures: crossing bracket, the number of brackets in the system's parse that cross the treebank parse; recall, a ratio of the number of correct brackets in the system's parse to the total number of brackets in the treebank parse and precision, a ratio of the number of correct brackets in the system's parse to the total number of brackets in the system's parse. The Parseval scheme served its purpose of providing a cross-representational evaluation metric using which the performance of various parsers could be quantified and compared – a previously impossible task.

There have been several objections to the Parseval scheme. In this section, we summarize some of the objections and limitations of this scheme. First, it was observed that crossing brackets measure penalizes mis-attachments more than once [Lin, 1995]. Consider for example, the sentence (1) with the annotation as the gold standard and the sentences (2) and (3) as parses from two parsing systems.

(1) [She [bought [[an incredibly expensive coat] [with [[gold buttons] and [fur lining]]]] [at [the store]]]]

(2) [She [bought [[an incredibly expensive coat] [with [[gold buttons] and [[fur lining] [at [the store]]]]]]]]

(3) [She [bought [an incredibly expensive coat] with [gold buttons] and [fur lining] [at [the store]]]]

Due to the mis-attachment of the phrase [at [the store]] to the phrase [fur lining] in the parse 2, three crossing brackets are created.

[1] See [Cole *et al.*, 1996].

1. Between [[gold buttons] and [fur lining]] and [[fur lining] [at [the store]]]

2. Between [with [[gold buttons] and [fur lining]]] and [[gold buttons] and [[fur lining] [at [the store]]]]

3. Between [[an incredibly expensive coat] [with [[gold buttons] and [fur lining]]]] and [[an incredibly expensive coat] [with [[gold buttons] and [[fur lining] [at [the store]]]]]]

The recall of 2 is $\frac{6}{10}$=60% and the precision is $\frac{7}{10}$=63.6%. In contrast, the shallow parse in 3 with its minimal bracketing has a recall of 70%, a precision of 100% and no crossing brackets with the parse in 1. Although the parse 2 contains more information and is closer to the intended parse 1, it scores lower on the Parseval metric.

A second objection of the Parseval metric is that the precision measure penalizes a parser for inserting extra, possibly correct, brackets, if annotation in the treebank is skeletal [Srinivas *et al.*, 1995]. Consider a finer analysis of the phrase "an incredibly expensive coat" as shown in 4.

(4) [an [incredibly expensive] coat]

Since the structure in 4 is more detailed than the structure for the phrase in 1, the analysis in 4 results in a recall of 100% but a precision of 50%.

Due to the above mentioned limitations of the Parseval metric, it is unclear as to how the score on this metric relates to success in parsing. It is also not clear if this metric can be used for comparing parsers with different degrees of structural fineness since the score on this metric is tightly related to the degree of structural detail in the gold standard.

A significant limitation of the Parseval metric is that it has not been designed to evaluate parsers that produce partial parses. A partial parser could potentially produce constituents/chunks that may not be connected in a complete tree. In cases where an attachment of a constituent is hard to make, it might be equally useful to leave the constituent unattached than to force the parser to always attach it to some node. However, the bracket representation used by Parseval is inadequate for representing disconnected trees. In the bracket notation, there is an implicit assumption that the unattached constituents are attached to the root of the tree.

Another limitation of the Parseval metric is that it does not compare analyses to parsing tasks or applications. Parseval computes crossing brackets, recall and precision as a way of approximating the exact match criterion (identity of the output parse with the gold standard parse). Performance using the exact match criterion has a direct interpretation in terms of performance on tasks that a parser may be put to use. However, a measure that is an approximation to the exact match criterion, such as Parseval, has a less direct interpretation since the definition of approximation varies from task to task. Applications that use a parser may be interested in specific structures, Noun Phrases, Appositives, Predicative Nominatives, Subject-Verb-Object relations and so on. The Parseval metricis not fine-grained enough to evaluate parses with respect to specific syntactic phenomena. It divorces the parser from the application the parser is embedded in.

5. A General Framework for Parser Evaluation: Our Proposal

In [Srinivas *et al.*, 1996], we proposed an alternative comparative parser evaluation method which overcomes the limitations of the Parseval scheme, and which can be used to evaluate both full and partial parsers.

In the preceding sections, we have identified two kinds of parser evaluations – intrinsic, application independent evaluation, and extrinsic, application dependent evaluation. In this section, we review the general framework presented in [Srinivas *et al.*, 1996] called the Relation Model of parser evaluation that allows for both kinds of evaluations. In this framework, a parse is viewed in multiple dimensions with each dimension expressing a relation R. A parser can be evaluated along any one (or all) of these dimensions. A gold standard used for evaluation is viewed as expressing one particular relation. Performance of the parser in expressing the relation R is measured in terms of recall, precision and F-measure [Appelt *et al.*, 1993]. F-measure provides a parameter β that can be set so as to take into account the relative importance of recall to precision. A summary of the evaluation framework is shown in 1.

- Let x R y represent that x is in a relation R with respect to y .
- Let S$_{gold}$ be the relation, R, expressed in the key (annotated corpus).
- Let S$_{out}$ be the relation, R, expressed in the output of the parser.
- Recall = $|S_{gold} \cap S_{out})|/|S_{gold}|$
- Precision = $|(S_{gold} \cap S_{out})|/|S_{out}|$
- F-Measure = $((\beta^2 + 1) * P * R)/(\beta^2 * P + R)$ where β is the relative importance of Recall and Precision.

Figure 1: Summary of the Relation Model of Parser Evaluation

A few instantiations of this general framework are as follows: For evaluating chunks of type t where t could be a noun chunk, verb chunk, preposition phrase and so on; R is defined as the relation: x *starts* and y *ends* the chunk type t. A similar instantiation could be used to evaluate the parser on specific constructions. For evaluating dependencies, the relation R is defined as x *depends on* y. The dependency links could be evaluated further based on their labels.

The objective of this evaluation is to provide a measure that can be used to compare parsers irrespective of their output representation. Parsers are either constituency-based or dependency-based systems. Constituency-based parsers produce a hierarchically organized constituent structure, while dependency-based parsers produce a set of dependency linkages between the words of a sentence. Furthermore, parsers could produce full parses that span the entire input string or partial parses that are a set of locally consistent parses for non-overlapping substrings of the input. Also, in terms of the detail of a parse, statistical constituency-based parsers produce relatively flat structures for Noun Phrases and Verb groups since the treebanks they are trained off of do not annotate for the internal structures of these elements. Hence, parsers that do produce internal structure for these elements need to be normalized first so that non-recursive phrasal constituents are flattened to chunks such as noun chunks, verb chunks, preposition chunks.

Also, hierarchically organized constituency notation does not have the provision to represent partial parses since an "unattached constituent" is implicitly assumed to attach to the highest constituent. Hence to facilitate the comparison of partial and full parsers, we propose that the accuracy of hierarchical structures be measured in terms of the accuracy of the relations between chunks that are expressed by dependency links between certain words (typically heads) of the chunks. Partial parses are thus penalized by their not being able to express certain dependency links due to unattached constituents.

Thus we propose a normalized representation that is based on chunks and dependency links. This assumes that all systems either directly represent their output using some set of standard syntactic constituents, such as NP, PP, and S, or can their convert their output into such elements.[2] Evaluating this representation amounts to first evaluating the flat phrasal structures such as noun chunks, verb groups, preposition phrases (disregarding the attachment location) and then evaluating the correctness of hierarchical structures using dependencies between words (typically, heads) of the two chunks. If the dependency links are labeled then evaluation could be performed with and without the labels.

One of the dimensions of adequacy evaluation is to evaluate parsers from the point of view of specific grammatical constructions such as minimal and maximal Noun phrases, Appositives, Preposition Phrase modifiers, Predicative constructions, Relative Clauses, Parentheticals and Predicate-Argument relations. The degree of difficulty and accuracy of identifying these grammatical constructions, given a parse, depends on the representation adopted by the grammar underlying the parser. The appropriateness of the representation for a task can only be evaluated by evaluating the accuracy with which these grammatical constructions can be identified. Hence we recommend that parsers be evaluated based on their performance in identifying specific grammatical constructions.

6. Evaluation of the XTAG English Grammar

6.1. Intrinsic: Parsing Corpora and the TSNLP test-suite

In the XTAG project, we have used corpus analysis in two main ways: (1) to measure the performance of the English grammar on a given genre and (2) to identify gaps in the grammar. The former analysis has been performed on Wall Street Journal, IBM Manual and ATIS data; those results are reported in [Doran *et al.*, 1994], and simply reflect the percentage of sentences for which any/a correct parse is produced. We also reported our performance on a test set of Indian newswire sentences in [Srinivas *et al.*, 1996].

The second type of evaluation involves performing detailed error analysis on the sentences rejected by the parser, and we have done this several times on WSJ and Brown data. Based on the results of such analysis, we prioritize upcoming grammar development efforts. The results of a recent error analysis are shown in Table 1. The table does not show errors in parsing due to mistakes made by the POS tagger which contributed the largest number of errors: 32. At this point, we have added a treatment of punctuation to handle #1, an analysis of time NPs (#2), a large number of multi-word prepositions (part of #3), gapless relative clauses (#7), bare infinitives (#14) and have added the missing subcategorization (#3) and missing lexical entry (#12). We are in the process of extending the parser to handle VP coordination (#9) [Sarkar and Joshi, 1996]. We find that this method of error analysis is very useful in focusing our research efforts in a productive direction.

Rank	No of errors	Category of error
#1	11	Parentheticals and appositives
#2	8	Time NP
#3	8	Missing subcat
#4	7	Multi-word construction
#5	6	Ellipsis
#6	6	Not sentences
#7	3	Relative clause with no gap
#8	2	Funny coordination
#9	2	VP coordination
#10	2	Inverted predication
#11	2	Who knows
#12	1	Missing entry
#13	1	Comparative?
#14	1	Bare infinitive

Table 1: Results of Corpus Based Error Analysis

To ensure that we are not losing coverage of certain phenomenon as we extend the grammar, we have a benchmark set of grammatical and ungrammatical sentences from our technical report [XTAG-Group, 1995]. We parse these sentences periodically to ensure that in adding new features and constructions to the grammar, we are not blocking previous analyses. There are 167 example sentences in this set.

6.1.1. TSNLP

In addition to corpus-based evaluation, we have also run the English Grammar on the Test Suites for Natural Language Processing (TSNLP) English corpus [Lehmann *et al.*, 1996]. The corpus is intended to be a systematic col-

[2] If the set of constituent labels of the two parsers being compared are not the same then the parsers can be evaluated on unlabeled constituents.

lection of English grammatical phenomena, including complementation, agreement, modification, diathesis, modality, tense and aspect, sentence and clause types, coordination, and negation. It contains 1409 grammatical sentences and phrases and 3036 ungrammatical ones.

There were 42 examples which we judged ungrammatical, and removed from the test corpus. These were sentences with conjoined subject pronouns, where one or both were accusative, e.g. *Her and him succeed.* Overall, we parsed 61.4% of the 1367 remaining sentences and phrases. The errors were of various types, broken down in Table 2. As with the error analysis described above, we used this information to help direct our grammar development efforts. It also highlighted the fact that our grammar is heavily slanted toward American English—our grammar did not handle *dare* or *need* as auxiliary verbs, and there were a number of very British particle constructions, e.g. *She misses him out*.

One general problem with the test-suite is that it uses a very restricted lexicon, and if there is one problematic lexical item it is likely to appear a large number of times and cause a disproportionate amount of grief. *Used to* appears 33 times and we got all 33 wrong. However, it must be noted that the XTAG grammar has analyses for syntactic phenomena that were not represented in the TSNLP test suite such as sentential subjects and subordinating clauses among others. This effort was, therefore, useful in highlighting some deficiencies in our grammar, but did not provide the same sort of general evaluation as parsing corpus data.

6.2. Comparative Evaluation via Text Chunking

Following the model of parser evaluation described in Section and illustrated in Figure 1, we evaluated the XTAG parser for the text chunking task [Abney, 1991]. In particular, we compared NP chunks and verb group (VG) chunks[3] produced by the XTAG parser with the NP and VG chunks from the Penn Treebank [Marcus *et al.*, 1993]. The test involved 940 sentences of length 15 words or less from sections 17 to 23 of the Penn Treebank, parsed using the XTAG English grammar. The results are given in Table 3.

As described earlier, the results cannot be directly compared with other results in chunking such as in [Ramshaw and Marcus, 1995] since we do not train from the Treebank before testing. However, in earlier work, text chunking was done using a technique called supertagging [Srinivas, 1997] (which uses the XTAG English grammar) which can be used to train from the Treebank. The comparative results of text chunking between supertagging and other methods of chunking is shown in Figure 4.[4]

We also performed experiments to determine the accuracy of the derivation structures produced by XTAG on WSJ text, where the derivation tree produced after parsing XTAG is interpreted as a dependency parse. We took sentences that were 15 words or less from the Penn Treebank [Marcus *et al.*, 1993]. The sentences were collected from sections 17–23 of the Treebank. 9891 of these sentences were given at least one parse by the XTAG system. Since XTAG typically produces several derivations for each sentence we simply picked a single derivation from the list for this evaluation. Better results might be achieved by ranking the output of the parser using various techniques [Srinivas *et al.*, 1995].

There were some striking differences in the dependencies implicit in the Treebank and those given by XTAG derivations. For instance, often a subject NP in the Treebank is linked with the first auxiliary verb in the tree, either a modal or a copular verb. Also XTAG produces some dependencies with a NP, while a large number of NPs in the Treebank are directly dependent on the verb. To normalize for these facts, we took the output of the NP and VG chunker described above and accepted as correct any dependencies that emanated from one chunk to another.

For example, for the sentence *Borrowed shares on the Amex rose to another record*, the XTAG and Treebank chunks are shown below.

```
XTAG chunks:
 [Borrowed shares] [on the Amex] [rose]
    [to another record]
Treebank chunks:
 [Borrowed shares on the Amex] [rose]
    [to another record]
```

Using these chunks, we can normalize for the fact that in the dependencies produced by XTAG *borrowed* is dependent on *shares* while in the Treebank *borrowed* is directly dependent on the verb *rose*. The dependencies for the sentence are given below.

XTAG dependency	Treebank dependency
Borrowed::shares	Borrowed::rose
shares::rose	shares::rose
on::shares	on::shares
the::Amex	the::Amex
Amex::on	Amex::on
rose::NIL	rose::NIL
to::rose	to::rose
another::record	another::record
record::to	record::to

After this normalization, testing simply consisted of counting how many of the dependency links produced by XTAG matched the Treebank dependency links. Due to some tokenization and subsequent alignment problems we could only test on 835 of the original 9891 parsed sentences. There were a total of 6135 dependency links extracted from the Treebank. The XTAG parses also produced 6135 dependency links for the same sentences. Of the dependencies produced by the XTAG parser, 5165 were correct giving us an accuracy of 84.2%.

7. Conclusion

[3] We treat a sequence of verbs and verbal modifiers, including auxiliaries, adverbs, modals as constituting a verb group.

[4] It is important to note in this comparison that the supertagger uses lexical information on a per word basis only to pick an initial set of supertags for a given word.

Error Class	%	Example
POS Tag	19.7%	She adds to/V it , He noises/N him abroad
Missing lex item	43.3%	*used* as an auxiliary V, *calm NP down*
Missing tree	21.2%	*should've, bet NP NP S, regard NP as Adj*
Feature clashes	3%	*My every firm, All money*
Rest	12.8%	*approx, e.g.*

Table 2: Breakdown of TSNLP Errors

	NP Chunking	VG Chunking
Recall	82.15%	74.51%
Precision	83.94%	76.43%

Table 3: Text Chunking performance of the XTAG parser

In this paper, the various methods used to evaluate the XTAG parser and English grammar were described. We considered the methods proposed in the literature for grammar and parser evaluation and gave some empirical reasons why we chose to use certain methods over others. We described a general framework which was then used to evaluate the English grammar and parser developed as part of the XTAG project. We also gave results based on the framework proposed. In particular, we gave results on text chunking using the XTAG grammar and parser.

8. *

References

[Abney, 1991] Steven Abney. Parsing by chunks. In Robert Berwick, Steven Abney, and Carol Tenny, editors, *Principle-based parsing*. Kluwer Academic Publishers, 1991.

[Alshawi *et al.*, 1992] Hiyan Alshawi, David Carter, Richard Crouch, Steve Pullman, Manny Rayner, and Arnold Smith. *CLARE – A Contextual Reasoning and Cooperative Response Framework for the Core Language Engine*. SRI International, Cambridge, England, 1992.

[Appelt *et al.*, 1993] D. Appelt, J. Hobbs, J. Bear, D. J. Israel, and M. Tyson. FASTUS: a finite-state processor for information extraction from real-world text. In *Proceedings of IJCAI-93*, Chambery, France, September 1993.

[Black *et al.*, 1993a] Ezra Black, R. Garside, and G. Leech (eds.). *Statistically-driven computer grammars of English: The IBM/Lancaster approach*. Rodopi, Amsterdam, 1993.

[Black *et al.*, 1993b] Ezra Black, Fred Jelinek, John Lafferty, David M. Magerman, Robert Mercer, and Salim Roukos. Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings of the* 31^{st} *Conference of Association of Computational Linguistics*, 1993.

[Briscoe and Carroll, 1995] Ted Briscoe and John Carroll. Developing and Evaluating a Probabilistic LR Parser of Part-of-Speech and Punctuation Labels. In *Proceedings of the Fourth International Workshop on Parsing Technologies (IWPT95)*, Prague, Czech Republic, 1995.

[Briscoe *et al.*, 1996] Ted Briscoe, John Carroll, Nicoletta Calzolari, Stefano Federici, Simonetta Montemagni, Vito Pirrelli, Greg Grefenstette, Antonio Sanfilippo, Glenn Carroll, and Mats Rooth. Shallow parsing and knowledge extraction for language engineering – work package 1. Specification of Phrasal Parsing, Prefinal Report, May 1996.

[Church, 1988] Kenneth Ward Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *2nd Applied Natural Language Processing Conference*, Austin, Texas, 1988.

[Cole *et al.*, 1996] Ronald A. Cole, Joseph Mariani, Hans Uszkoreit, Annie Zaenen, and Victor Zue. Survey of the state of the art in human language technology, 1996. http://www.cse.ogi.edu/CSLU/HLTsurvey/.

[Doran *et al.*, 1994] Christy Doran, Dania Egedi, Beth Ann Hockey, B. Srinivas, and Martin Zaidel. XTAG System - A Wide Coverage Grammar for English. In *Proceedings of the* 17^{th} *International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan, August 1994.

[Grover *et al.*, 1993] Claire Grover, John Carroll, and Ted Briscoe. *The Alvey Natural Language Tools Grammar*, 4th release edition, 1993.

[Harrison *et al.*, 1991] P. Harrison, S. Abney, D. Fleckenger, C. Gdaniec, R. Grishman, D. Hindle, B. Ingria, M. Marcus, B. Santorini, and T. Strzalkowski. Evaluating syntax performance of parser/grammars of English. In *Proceedings of the Workshop on Evaluating Natural Language Processing Systems, ACL.*, 1991.

[Karlsson *et al.*, 1994] Karlsson, Voutilainen, Heikkilä, and Anttila. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mourton de Gruyter, Berlin and New York, 1994.

System	Training Size	Recall	Precision
Ramshaw & Marcus	Baseline	81.9%	78.2%
Ramshaw & Marcus (without lexical information)	200,000	90.7%	90.5%
Ramshaw & Marcus (with lexical information)	200,000	92.3%	91.8%
Supertags	Baseline	74.0%	58.4%
Supertags	200,000	93.0%	91.8%
Supertags	1,000,000	93.8%	92.5%

Table 4: Performance comparison of the transformation based noun chunker and the supertag based noun chunker

[Lehmann *et al.*, 1996] Sabine Lehmann, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, Hervé Compagnion, Judith Baur, Lorna Balkan, and Doug Arnold. TSNLP — Test Suites for Natural Language Processing. In *Proceedings of COLING 1996*, Kopenhagen, 1996.

[Lin, 1995] Dekang Lin. A Dependency-based Method for Evaluating Broad-Coverage Parsers. In *Proceedings of IJCAI-96*, Montreal, Canada, August 1995.

[Marcus *et al.*, 1993] Mitchell M. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19.2:313–330, June 1993.

[Oepen *et al.*, forthcoming] Stephan Oepen, Klaus Netter, and Judith Klein. *TSNLP – Test Suites for Natural Language Processing*. CSLI Lecture Notes, forthcoming. http://cl-www.dfki.uni-sb.de/tsnlp/.

[Ramshaw and Marcus, 1995]
Lance Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, MIT, Cambridge, Boston, 1995.

[Sampson *et al.*, 1989]
G. Sampson, R. Haigh, and E. Atwell. Natural language analysis by stochastic optimization: a progress report on project april. *Journal of Experimental and Theoretical Artificial Intelligence*, 1:271–287, 1989.

[Sarkar and Joshi, 1996] Anoop Sarkar and Aravind Joshi. Coordination in Tree Adjoining Grammars: Formalization and Implementation. In *Proceedings of the* 18^{th} *International Conference on Computational Linguistics (COLING '94)*, Copenhagen, Denmark, August 1996.

[Srinivas *et al.*, 1995] B. Srinivas, Christine Doran, and Seth Kulick. Heuristics and parse ranking. In *Proceedings of the* 4^{th} *Annual International Workshop on Parsing Technologies*, Prague, September 1995.

[Srinivas *et al.*, 1996]
B. Srinivas, Christine Doran, Beth Ann Hockey, and Aravind Joshi. An approach to robust partial parsing and evaluation metrics. In *Proceedings of the Workshop on Robust Parsing at European Summer School in Logic, Language and Information*, Prague, August 1996.

[Srinivas, 1997] B. Srinivas. Performance Evaluation of Supertagging for Partial Parsing. In *Proceedings of Fifth International Workshop on Parsing Technology*, Boston, USA, September 1997.

[XTAG-Group, 1995]
The XTAG-Group. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 95-03, University of Pennsylvania, 1995. Updated version available at http://www.cis.upenn.edu/ xtag/tr/tech-report.html.