Coverage-based Evaluation of Parser Generalizability

Tuomo Kakkonen and **Erkki Sutinen**
Department of Computer Science and Statistics
University of Joensuu
P.O. Box 111, FI-80101 Joensuu, Finland
{tuomo.kakkonen,erkki.sutinen}@cs.joensuu.fi

Abstract

We have carried out a series of coverage evaluations of diverse types of parsers using texts from several genres such as newspaper, religious, legal and biomedical texts. We compared the overall coverage of the evaluated parsers and analyzed the differences by text genre. The results indicate that the coverage typically drops several percentage points when parsers are faced with texts on genres other than newspapers.

1 Introduction

The fact that most of the parser evaluation resources employed consist of texts from a single genre constitutes a deficiency in most of the parser evaluations. Evaluations are typically carried out on newspaper texts, *i.e.* on section 23 of the *Penn Treebank* (PTB) (Marcus et al., 1993). A further complication is that many parsing models are trained on the same treebank. Parsers therefore come to be applied to texts from numerous other genres untested. The obvious question that confronts us in these circumstances is: How well will a parser that performs well on financial texts from the Wall Street Journal generalize to other text types?

This present paper addresses parser evaluation from the perspective of coverage. It is a part of a set of evaluations in which selected parsers are evaluated using five criteria: *preciseness*, coverage, *robustness*, *efficiency* and *subtlety*. *Parsing coverage* refers to the ability of a parser to produce an analysis of sentences of naturally occurring free-text. We used parsing coverage to assess the generalizability of the grammars and parsing models and we looked for answers to the following research questions:

- What is the parsing coverage of the evaluated parsers?
- How does the text genre affect the parsing coverage?

Previous work on evaluation methods and resources is discussed in Section 2. Section 3 describes the evaluation method and test settings. In Section 4, we give the results of the experiments. Section 5 concludes with a description of remaining problems and directions for future research.

2 Preliminaries

2.1 Coverage Evaluation

Prasad and Sarkar (2000) observe that the notion of coverage has the following two meanings in the context of parsing. *Grammatical coverage* is the parser's ability to handle different linguistic phenomena, and *parsing coverage* is a measure of the percentage of naturally occurring free text in which a parser can produce a full parse. We divide parsing coverage further into *genre coverage* on different types of texts such as newspapers, religious, biomedicine and fiction.[1]

[1] The classification of texts in terms of domain, genre, register and style is a rather controversial issue (see, for example, discussion by Lee (2001)). A detailed analysis of these issues falls outside of the scope of this paper. We have therefore adopted a simplified approach by indicating differences between texts by using the word genres. One may think of genres (in this sense) as indicating fundamental categorical differences between texts that are revealed in sets of attributes such as domain (e.g. art, science, religion, government), medium

Parsing coverage can be measured as the percentage of input sentences to which a parser is able to assign a parse. No annotated text is needed for performing parsing coverage evaluations. On one hand, it can be argued that coverage alone constitutes a rather weak measure of a parser's performance, and thus of its *generalizability* to diverse text genres. An obvious problem with measuring coverage alone is that a parser returning undetailed and flat analyses will easily get high coverage, whereas a parser that outputs detailed analyses will suffer in covering all the input sentences. Moreover, preciseness and coverage can be seen as conflicting requirements for a parser. Increasing preciseness of the grammar often causes its coverage to decrease; adding more constraints to the grammar causes some of the sentences to be rejected even they are acceptable to users of the language. Loosening the constraints allows more sentences to be parsed, thus increasing the coverage, but at the same time easily leads into overgeneration, problems with disambiguation and decreased preciseness.

On the other hand, the points that we raised above indicate that there is a strong relationship between coverage and preciseness. The aim of syntactic parsers is to analyze whole sentences, not just fragments (constituents/D links) precisely. The connection between coverage and preciseness is clear in the case of sentence level evaluations measures[2]: A sentence that cannot be fully analyzed cannot have a complete match with the correct structure in the evaluation resource. Consequently, we argue that coverage can be used a measure of generalizability; It sets the upper bound for the performance on the sentence-level evaluation measures. However, the evaluation should always be accompanied with data on the preciseness of the parser and the level of detail in its output.

2.2 Previous Coverage and Cross-genre Evaluations

Relatively little work has been done on the empirical evaluation of parsers for text types other than newspaper texts. A key issue in available evaluation materials is the genre homogeneity. Almost all the available resources are based on a single genre (nearly always newspaper texts). This makes it impossible to extrapolate anything useful about the generalizability of the developed grammars and parsing models.

To our knowledge, this experiment is the only one reported in the literature that compares the coverage of a set of parsers for English. The studies that critically examine the genre dependency have come to the same unsurprising conclusion that the text genre has an effect on the parser's performance. The genre dependency of parsers is an accepted fact and has been described by, among others, Sekine (1997) and Gildea (2001). For example, Clegg and Shepherd (2005) have undertaken experiments on biomedical data using the *GENIA treebank*. Laakso (2005) reports experiments on the *CHILDES* corpus of transcribed speech between parents and the children. Mazzei and Lombardo (2004) report cross-training experiments in Italian on newspaper and civil law texts. They observed a dramatic drop of, most commonly, around 10-30 percentage points in the parsing coverage.

2.3 Reasons for the Coverage Drop

Genre dependency is caused by several factors. One is that each text genre is characterized by genre-specific words (Biber, 1993). Another feature of genre dependency is syntactic structure distributions. Baldwin et al. (2004) have conducted one of the rare studies that offer an analysis of the main reasons for the diminished coverage. They experimented with an HPSG grammar that was a created manually based on a corpus of data extracted from informal genres such as conversations about schedules and e-mails about e-commerce. The grammar was used for parsing a random sample of texts from several genres. A diagnosis of failures to parse sentences with full lexical span[3] revealed the following causes for the errors: missing lexical entries (40%), missing constructions (39%), preprocessor errors (4%), fragments (4%), parser failures (4%), and garbage strings (11%). They came to the conclusion that lexical expansion should be the first step in the process of parser enhancement.

(e.g. spoken, written), content (topic, theme) and type (narrative, argumentation, etc.).

[2] For example Yamada & Matsumoto (2003) uses *complete match* metric (the percentage of sentences whose unlabeled D structure is completely correct) to evaluate the sentence-level preciseness of D parsers.

[3] Sentences that contained only words included in the lexicon.

3 Experiments

3.1 Research Approach

In order to investigate the effect of the text genre on the parsing results, we constructed a test corpus of more than 800,000 sentences and divided them into six genres. We parsed these texts by using five parsing systems.

The design of our test settings and materials was guided by our research questions (above). We answered the first question by parsing vast document collections with several state-of-the-art parsing systems and then measuring their parsing coverage on the data. Because we had divided our purpose-built test set into genre-specific subsets, this allowed us to measure the effects of genre variance and so provide an answer to the second research question. We also included two parsers that had been developed in the 1990s to evaluate the extent to which progress has been made in parsing technology in genre dependency and parsing coverage.

3.2 Evaluation Metric and Measures

The most important decision in parsing coverage evaluation is how the distinction between a covered and uncovered sentence is made. This has to be defined separately for each parser and the definition depends on the type of output. We implemented a set of Java tools to record the statistics from the parsers' outputs. In addition to completely failed parses, we recorded information about incomplete analyses and the number of times the parsers crashed or terminated during parsing.

3.3 Materials

The test set consisted of 826,485 sentences divided into six sub-corpora. In order to cover several genres and to guarantee the diversity of the text types, we sourced a diversity of materials from several collections. There are six sub-corpora in the material and each covers one of the following genres: newspaper, legislation, fiction, non-fiction, religion and biomedicine.

Table 1 shows the sub-corpora and the figures associated with each corpus. In total there were 15,385,855 tokens. The style of the newspaper texts led us to make an initial hypothesis that a similar performance would probably be achievable with non-fiction texts, and we suspected that the legislative and fiction texts might be more difficult to parse because of the stylistic idiosyncrasies involved. Biomedical texts also contained a considerable number of words that are probably not found in the lexicons. These two difficulties were compounded in the religious texts, and the average length of the religion sub-corpus was far higher than the average.

Table 1. The test sets**.**

Genre	Description	No. of sentences	Avg. length
Legislation	Discussions of the Canadian Parliament	390,042	17.2
Newspaper	Texts from several newspapers	217,262	19.5
Fiction	Novels from the 20th and 21st century	97,156	15.9
Non-fiction	Non-fiction books from the 20th and 21st century	61,911	21.9
Religion	The Bible, the Koran, the Book of Mormon	45,459	27.1
Biomedicine	Abstracts from biomedical journals	14,655	21.6
TOTAL		826,485	18.6

3.4 The Parsers

We included both dependency (D)- and phrase structure (PS)-based systems in the experiment. The parsers use a *Probabilistic Context-free* Grammar (PCFG), *Combinatory Categorial Grammar* (CCG), a semi-context sensitive grammar and a D-based grammar.

Apple Pie Parser (APP) (v. 5.9, 4 April 1997) is a bottom-up probabilistic chart parser which finds the analysis with the best score by means of best-first search algorithm (Sekine, 1998). It uses a semi-context sensitive grammar obtained automatically from the PTB. The parser outputs a PS analysis consisting of 20 syntactic tags. No word-level analysis is assigned. We regard a sentence as having been covered if APP finds a single S non-terminal which dominates the whole sentence and if it does not contain any X tags which would indicate constituents of unrecognized category.

C&C Parser (v. 0.96, 23 November 2006) is based on a CCG. It applies log-linear probabilistic tagging and parsing models (Clark and Curran, 2004). Because the parser marks every output as

either parsed or failed, evaluation of failed parses is straightforward. Fragmented parses were detected from the *grammatical relations* (GR) output. Because GR representations can form cycles, an analysis was not required to have a unique root. Instead, a parse was regarded as being incomplete if, after projecting each GR to a graph allowing cycles, more than one connected set (indicating a fragmented analysis) was found.

MINIPAR (unknown version, 1998) is a principle-based parser applying a distributed chart algorithm and a D-style grammar (Lin, 1998). The syntactic tagset comprises 27 grammatical relation types and word and phrase types are marked with 20 tags. A sentence is regarded as having been covered by MINIPAR if a single root is found for it that is connected to all the words in the sentence through a path. The root should in addition be assigned with a phrase/sentence type marker.

Stanford Parser (referred in the remainder of this text as SP) (v. 1.5.1, 30 May 2006) can use both an unlexicalized and lexicalized PCFGs (Klein and Manning, 2003). This parser uses a CYK search algorithm and can output both D and PS analyses (de Marneffe et al., 2006). We ran the experiment on the unlexicalized grammar and carried out the evaluation on the D output consisting of 48 D types. We regard a sentence as having been covered by SP in a way similar to that in MINIPAR: the sentence is covered if the D tree returned by the parser has a single root node in which there is a path to all the other nodes in the tree.

StatCCG (Preliminary public release, 14 January 2004) is a statistical parser for CCG that was developed by Julia Hockenmaier (2003). In contrast to C&C, this parser is based on a generative probabilistic model. The lexical category set has around 1,200 types, and there are four atomic types in the syntactic description. StatCCG marks every relevant sentence as 'failed' or 'too long' in its output. We were therefore able to calculate the failed parses directly from the system output. We regarded parses as being partially covered when no sentence level non-terminal was found.

3.5 Test Settings

We wanted to create similar and equal conditions for all parsers throughout the evaluation. Moreover, language processing applications that involve parsing must incorporate practical limits on resource consumption.[4] Hence, we limited the use of memory to the same value for all the parsers and experiments.[5] We selected 650 MB as the upper limit. It is a realistic setting for free working memory in a typical personal computer with 1 GB memory.

4 Results

Table 2 summarizes the results of the experiments. The parsing coverage of the parsers for each of the sub-corpora is reported separately. Total figures are given for both parser and sub-corpus level. In Table 3, the coverage figures are further broken down to indicate the percentage of the analyses that failed or were incomplete or those occasions on which the parser crashed or terminated during the process.

The five parsers were able to cover, on average, 88.8% of the sentences. The coverage was, unsurprisingly, highest on the newspaper genre. The lowest average coverage was achieved on the religion genre. The difficulties in parsing the religious texts are attributable at least in part to the length of the sentences in the sub-corpus (on average 27.1 words per sentence), which was the highest over all the genres. Contrary to our expectation, the biomedical genre, with its specialist terminology, was not the most difficult genre for the parsers.

If one excludes the one-word sentences from the legislation dataset, SP had the best coverage and best generalizability over the text genres. APP was the second best performer in this experiment, both in coverage and generalizability. While APP produces shallow parses, this helps it to obtain a high coverage. Moreover, comparing the F-scores reported in the literature for the five parsers revealed that the F-score (70.1) of this parser was more than 10 percentage points lower than the score of the second worst parser MINIPAR. Thus, it is obvious that the high coverage in APP is achieved at the cost of preciseness and lack of detail in the output.

[4] In addition, parsing in the order of hundreds of thousands of sentences with five parsers takes thousands of hours of processor time. It was therefore necessary for us to limit the memory consumption in order to be able to run the experiments in parallel.

[5] Several methods were used for limiting the memory usage depending on the parser. For example, in the Java-based parsers, the limit was set on the size of the Java heap.

Table 2. Comparison of the parsing results for each sub-corpus and parser. "Average" column gives the average of the coverage figures for the six genres weighted according to the number of sentences in each genre. The column labeled "Generalizability" shows the drop of the coverage in the lowest-scoring genre compared to the coverage in the newspaper genre.

Parser	*Newspaper*	*Legislation*	*Fiction*	*Non-fiction*	*Religion*	*Biomedi-cine*	*Average*	*Gener-alizability*
APP	99.8	98.9	97.5	96.4	93.1	98.9	98.5	6.7
C&C	87.8	84.9	86.0	81.2	75.5	84.8	85.0	14.0
MINIPAR	88.0	68.8	68.0	71.5	34.4	70.1	72.1	60.9
SP*	99.8	99.5	98.0	98.3	98.9	98.5	99.2	1.8
StatCCG	96.7	85.2	87.7	86.7	94.0	83.3	89.1	13.9
Average	94.4	87.5	87.4	86.8	79.2	87.1	88.8	19.5

*SP experienced a coverage drop of tens of percentage points in comparison to other genres on the Hansard dataset. This was caused mainly by a single issue: the dataset contained a number of sentences that contained only a single word – sentences such as "Nay.", "Agreed.", "No." and so on. Because no root node is assigned to D analysis by SP, the parser did not return any analysis for such sentences. These sentences were omitted from the evaluation. When the sentences were included, the coverage on legislation data was 59.5% and the average was 73.4%.

Table 3**.** Breakdown of the failures. All the results are reported as a percentage of the total number of sentences. Column 'Incomplete' reports the proportion of sentences that were parsed, but the analysis was not full. Column 'Failed' shows those cases in which the parser was not able to return a parse. Column 'Terminated' shows the proportion of the cases in which the parser crashed or terminated during the process of parsing a sentence.

Parser	*Incomplete*	*Failed*	*Terminated*
APP	1.5	0.0	0.001
C&C	12.8	2.2	0.006
MINIPAR	27.9	0.0	0.009
SP	0.5	0.4	0.002
StatCCG	9.6	1.4	0.000
Average	10.5	0.8	0.004

While StatCCG outperformed C&C parser by 4.1 percentage points in average coverage, the two CCG-based parsers achieved a similar generalizability. StatCCG was the most stable parser in the experiment. It did not crash or terminate once on the test data.

The only parser based on a manually-constructed grammar, MINIPAR, had the lowest coverage and generalizability. MINIPAR also proved to have stability problems. While this parser achieved an 88.0% coverage with the newspaper corpus, its performance dropped over 10 percentage points with other corpora. Its coverage was only 34.4% with the religion genre. The most commonly occurring type of problem with this data was a fragmented analysis occasioned by sentences beginning with an 'And' or 'Or' that was not connected to any other words in the parse tree.

5 Conclusion

This paper describes our experiments in parsing diverse text types with five parsers operating with four different grammar formalisms. To our knowledge, this experiment is the only large-scale comparison of the coverage of a set of parsers for English reported in the literature. On average, the parsing coverage of the five parsers on newspaper texts was 94.4%. The average dropped from 5.6 to 15.2 percentage points on the other five text genres. The lowest average scores were achieved on the religion test set.

In comparison to MINIPAR, the results indicate that the coverage of the newer parsers has improved. The good performance of the APP may partly be explained by a rather poor preciseness: the rate of just over 70% is much lower than that of other parsers. APP also produces a shallow analysis that enables it to achieve a high coverage.

One observation that should be made relates to the user friendliness and documentation of the parsing systems. The parsing of a vast collection of texts using several parsing systems was neither simple nor straightforward. To begin with, most of the parsers crashed at least once during the course of the experiments. The C&C parser, for example, terminates when it encounters a sentence with two spaces between words. It would be far more con-

venient for users if such sentences were automatically skipped or normalized.

While another feature is that all the parsers have a set of parameters that can be adjusted, the accompanying documentation about their effects is in many cases insufficiently detailed. From the NLP practitioner's point of view, the process of selecting an appropriate parser for a given task is complicated by the fact that the output format of a parser is frequently described in insufficient detail. It would also be useful in many NLP applications if the parser were able to indicate whether or not it could parse a sentence completely. It would also be optimal if a confidence score indicating the reliability of the returned analysis could be provided.

The most obvious directions for work of this kind would include other text genres, larger collections of texts and more parsers. One could also pinpoint the most problematic types of sentence structures by applying error-mining techniques to the results of the experiments.

References

Timothy Baldwin, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen. 2004. Road-testing the English Resource Grammar over the British National Corpus. In *Proceedings of the 4th Language Resources and Evaluation Conference (LREC)*, Lisbon, Portugal.

Douglas Biber. 1993. Using Register-diversified Corpora for General Language Studies. *Computational Linguistics*, 19(2):219–241.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and Log-Linear Models. In *Proceedings of the 42nd ACL*, Barcelona, Spain.

Andrew B. Clegg and Adrian J. Shepherd. 2005. Evaluating and Integrating Treebank Parsers on a Biomedical Corpus. In *Proceedings of the Workshop on Software at the 43rd ACL*, Ann Arbor, Michigan, USA.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the 5th LREC*, Genoa, Italy.

Daniel Gildea. 2001. Corpus Variation and Parser Performance. In Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, Pittsburgh, Pennsylvania, USA.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. PhD Dissertation, University of Edinburgh, UK.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st ACL*, Sapporo, Japan.

Aarre Laakso. On Parsing CHILDES. 2005. In Proceedings of the Second Midwest Computational Linguistics Colloquium. Columbus, Ohio, USA,

David Lee. 2001. Genres, Registers, Text Types, Domains, and Styles: Clarifying the Concepts and Navigating a Path through the BNC Jungle. *Language Learning & Technology*, 5(3):37–72.

Dekang Lin. 1998. Dependency-Based Evaluation of MINIPAR. In *Proceedings of the 1st LREC*, Granada, Spain.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Alessandro Mazzei and Vincenzo Lombardo. 2004. A Comparative Analysis of Extracted Grammars. In *Proceedings of the 16th European Conference on Artificial Intelligence*, Valencia, Spain.

Rashmi Prasad and Anoop Sarkar. 2000. Comparing Test-suite Based Evaluation and Corpus-based Evaluation of a Wide Coverage Grammar for English. In *Proceedings of the Using Evaluation within HLT Programs: Results and Trends Workshop at the 2nd LREC*, Athens, Greece.

Geoffrey Sampson, editor. 1995. *English for the Computer: The Susanne Corpus and Analytic Scheme*. Oxford University Press, Oxford, UK.

Satoshi Sekine. 1997. The Domain Dependence of Parsing. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington, DC, USA.

Satoshi Sekine. 1998. *Corpus-based Parsing and Sublanguage Studies*. PhD Thesis. New York University, New York, USA.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *Proceedings of the 8th International Workshop on Parsing Technologies*, Nancy, France.