Automatic Parser Evaluation

Taavet Kikas, Margus Treumuth
April 23, 2007

1 The Task

The task was to implement an automatic parser evaluation program, which compares automatically parsed trees with the Gold Standard trees and computes PARSEVAL measures [Black et al. 1991]. The trees under evaluation were generated using Dan Bikel's parser [Bikel, 2004]. The Gold standard trees were taken from the Penn Treebank.

There were altogether five sets of parsed trees, which had to be evaluated. Each set contained parses for the same sentences. The difference was that each set was obtained using Dan Bikel's parser, which was trained on different training set.

2 PARSEVAL Measures

PARSEVAL measures are the standard way of evaluating parser quality. There are three PARSEVAL measures: Precision, Recall and Cross Brackets. Precision and recall are essentially the same measures that are used in information retrieval. Precision is the relative amount of correct constituents in a parse. A constituent is considered to be correct if it matches a constituent in the Gold Standard. Recall is the relative amount of correct constituents compared to the Gold Standard parse.

The number of Cross Brackets is a special measure used for parse trees. Cross Brackets occur if a parse contains bracketing (A (B C) but the Gold Standard provides bracketing (A B (C)). In other words, combining these two trees would result in a "tree" where each constituent can have more than one parent.

Precision

$$P = \frac{\text{\# Correct Constituents}}{\text{\# Constituents in parser output}}$$

Recall

$$R = \frac{\text{\# Correct Constituents}}{\text{\# Constituents in gold standard}}$$

(Relative) Cross Brackets

$$C = \frac{\text{\# Constituents in parser output that cross gold standard constituents}}{\text{\# Constituents in parser output}}$$

3 Preprocessing

The Gold Standard trees and the trees obtained using Bikel's parser follow essentially the same notation rules. Still, there are some important differences, which require some corpus preprocessing before the evaluation. We have divided this preprocessing into three steps.

3.1 Removing Unknown Words and Additional Labels

This preprocessing step eliminates the additional labels that are used in the Gold Standard and also eliminates words that are labeled with –NONE- tag. The extra brackets around the Gold Standard trees, which are absent in Bikel's trees, are left intact in this and all the following preprocessing steps. This reduces slightly recall as these brackets are accounted as a missing constituent.

The Gold Standard contains additional labels. In general, the basic label indicates the form of the constituent (NP, PP, ADJP, etc.), while additional labels (separated by a hyphen) indicate function (NP-SBJ = subject, ADVP-TMP = temporal adverb, CP-REL = relative clause, etc.). These additional labels exist only in Gold Tree parses and are absent in Bikel's parses. Therefore all additional labels are removed and only basic labels are kept.

Not all constituents in the Gold Standard are marked for function; in most cases there is at most one additional label, but there may be more (IP-INF-PRP = purpose infinitive, IP-IMP-SPE = direct speech imperative, etc.).

3.2 Removing Empty Constituents

The first preprocessing step eliminates unknown constituents but does not remove the phrases they are in. This can sometimes result in an empty phrase if all words in the phrase are removed. The purpose of the second preprocessing step is to remove recursively all empty constituents.

3.3 Black's Preprocessing

The third preprocessing step involves preprocessing operations proposed by Black et al. (1991). These operations include:

- Removing all auxiliaries
- Removing "not"
- Removing pre-infinitival "to"
- Removing punctuation

Other operations, like removing empty constituents, are performed already in previous phases or are not relevant for the current parses. We also note that we do not erase parenthesis enclosing a single constituent or a word as proposed by Black et al.

4 Evaluation Algorithm

The evaluation takes place as follows. First, the evaluation processes finds all constituents in a Gold Standard parse and then in Bikel's parse. For every constituent its label and span (the starting point and the ending point) are found (Table 1).

The next step is to compare the constituents between two parses. If constituent's label and its span in Bikel's parse match with some constituent's label and span in Gold Standard parse, then the constituent is said to be correct. The process is repeated for every constituent in Bikel's parse until all correct and incorrect constituents have been found. Finally the Precision, Recall and Cross-Brackets are calculated.

Tabel 1. And example of constituents and their spans.

Label	Text	Start	End
S	An appeal is expected .	1	6
NP	An appeal	1	3
DT	An	1	2
NN	appeal	2	3
VP	is expected	3	5
VBZ	is	3	4
VP	expected	4	5
VBN	expected	4	5
.	.	5	6

5 Implementation

The program for evaluating parses was implemented in Python and was approximately 400 lines long. To use the program, one must install the Python interpreter, which can be downloaded from http://www.python.org/download/. This step might be unnecessary on UNIX/Linux machines where the interpreter is often already installed. After installing Python, the evaluation program can be executed from the command line as follows.

```
python parseval.py
```

```
or just
```

```
parseval.py
```

Further instructions will be printed on the screen. The most common way to run the program is to enter something like this:

```
python parseval.py parse.txt gold.txt –c –x
```

This will run the evaluation process for the given text files (the first file contains parses to be evaluated, the second file contains Gold Standard parses). The command line arguments that were used tell the program to delete empty words, additional labels, and empty phrases.

The program accepts several more command-line options. The syntax for running the program from command line has the general form

```
parseval.py [file 1] [file 2] [options]
```

The program accepts the following arguments:

- -c remove unidentified words and additional labels eg. NP-SBJ --> NP
- -s display tags and their span
- -t display trees
- -i show evaluation results for every single tree
- -b use Black (Black et al., 1991) preprocessing (first step only)
- -x remove empty phrases

6 Evaluation

The evaluation consisted of four different experiments.

1) Evaluation without any preprocessing
2) Evaluation with removing unknown words and additional labels
3) Evaluation with removing unknown words, additional labels and empty constituents
4) Evaluation with removing unknown words, additional labels, empty constituents and Black's preprocessing

The results of these experiments are given in the following section.

7 Results

Table 3 contains the results of our experiments. The Precision and Recall vary from approximately 40% to 94% and are in all cases rather close. The Cross Brackets vary from 20% to 2%. The experiments clearly demonstrate the importance of preprocessing. The most important improvement is achieved by deleting empty phrases after having deleted unknown words. We see that applying Black's preprocessing steps in addition to other preprocessing steps, has no positive effect, rather a little negative one.

Most of the errors in the first two experiments were caused by unidentified words, e.g. "*-68" in the following sentence.

```
( (S
    (NP-SBJ-68 (DT An) (NN appeal) )
    (VP (VBZ is)
      (VP (VBN expected)
        (NP (-NONE- *-68) )))
    (. .) ))
```

Although unknown constituents were removed, phrases containing them were left intact, sometimes resulting in an empty phrase. As the length on every phrase was still assumed to be at least one, it caused a certain inconsistency between parses. To be more precise, empty phrases in Gold Standard caused a relative shift between the Gold Standard constituents and Bikel's constituents. This happened because empty phrases were already removed from Bikel's parses before. An example of what one extra constituent can cause, is given below (Table 2).

Table 2. An example of an empty noun-phrase shifting constituent spans.

Label	Text	Gold		Bikel	
		Start	End	Start	End
S	An appeal is expected .	1	7	1	6
NP	An appeal	1	3	1	3
DT	An	1	2	1	2
NN	appeal	2	3	2	3
VP	is expected	3	6	3	5
VBZ	is	3	4	3	4
VP	expected	4	6	4	5
VBN	expected	4	5	4	5
.	.	6	7	5	6
NP		**5**	**6**		

In this example NP is an empty noun-phrase, which can be found in the gold standard parse but not in the Bikel's parse. The small shift caused by this constituent results in several false constituents.

Table 3. The evaluation results.

	Precision	Recall	Cross Brackets
1) without preprocessing			
Gold Standard and Sections 02-09 PoS	42,73%	39,82%	21,64%
Gold Standard and Sections 02-09	41,58%	38,69%	21,67%
Gold Standard and Sections 02-15 PoS	48,30%	44,87%	20,59%
Gold Standard and Sections 02-15	41,88%	38,98%	21,55%
Gold Standard and Sections 02-21	50,92%	46,98%	18,81%
2) with preprocessing			
Gold Standard and Sections 02-09 PoS	52,20%	49,62%	19,94%
Gold Standard and Sections 02-09	50,96%	48,38%	19,98%
Gold Standard and Sections 02-15 PoS	54,54%	51,47%	20,40%
Gold Standard and Sections 02-15	51,36%	48,75%	19,84%
Gold Standard and Sections 02-21	57,47%	53,87%	18,51%
3) with preprocessing; with empty phrase elimination			
Gold Standard and Sections 02-09 PoS	92,54%	89,87%	2,54%
Gold Standard and Sections 02-09	90,05%	87,34%	2,88%
Gold Standard and Sections 02-15 PoS	93,80%	90,75%	2,92%
Gold Standard and Sections 02-15	90,77%	87,99%	2,64%
Gold Standard and Sections 02-21	94,49%	91,00%	3,27%
4) with preprocessing; with empty phrase elimination; with Black preprocessing			
Gold Standard and Sections 02-09 PoS	92,46%	89,59%	2,64%
Gold Standard and Sections 02-09	89,94%	87,02%	3,00%
Gold Standard and Sections 02-15 PoS	93,74%	90,47%	3,03%
Gold Standard and Sections 02-15	90,71%	87,71%	2,75%
Gold Standard and Sections 02-21	94,25%	90,53%	3,44%

8 Common parse errors

Table 4 summarizes most common parse errors that occurred during the evaluation. The largest contribution of mistakes is made by the first two experiments.

Tale 4. Most common parse errors in the order of frequency.

	Error
1.	Invalid NP
2.	Invalid VP
3.	Invalid S
4.	Invalid PP
5.	Invalid SBAR
6.	Invalid NN (most commonly used instead of DT according to constituent spans)
7.	Inavalid ADJP

Most of the parsing errors were span errors, which means that the constituents had an additional element or missed an element and therefore had different spans.

9 Description of our observations

We compared Dan Bikel's parser's output with a standardized (Gold Standard) output. Our automatic evaluation focused on comparing the respective constituent spans and labels.

We can conclude that this automatic evaluation technique is rather good. The availability of treebanks is the main issue for this comparison. A parser should also use the same kind of information as in the reference treebank. The experiments clearly demonstrate that failing to do this plummets the results.

We also note that although unification of parse formats is necessary before measuring Recall, Precision and Cross Brackets, one must be extra careful not to incline the result in a wrong direction by “scratching off” important characteristics of parse during preprocessing.

References

E. Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of english grammars. In Proceedings of the DARPA Speech and Natural Language Workshop, pages 306–311, 1991.

Dan Bikel. A distributional analysis of a lexicalized statistical parsing model. In Proceedings of the 9th Conference on Empirical Methods in Natural Language Processing (EMNLP), Barcelona, Spain, 2004.