Towards a More Careful Evaluation of Broad Coverage Parsing Systems

Wide R. Hogenhout and Yuji Matsumoto
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma
Nara 630-01, Japan
{marc-h,matsu}@is.aist-nara.ac.jp

Abstract

Since treebanks have become available to researchers a wide variety of techniques has been used to make broad coverage parsing systems. This makes quantitative evaluation very important, but the current evaluation methods have a number of drawbacks such as arbitrary choices in the treebank and the difficulty in measuring statistical significance. We suggest a more detailed method for testing a parsing system using constituent boundaries, with a number of measures that give more information than current measures, and evaluate the quality of the test. We also show that statistical significance cannot be calculated in a straightforward way, and suggest a calculation method for the case of Bracket Recall.

1 Introduction

During the last few years large treebanks have become available to many researchers, which has resulted in researches applying a range of new techniques for parsing systems. Most of the methods that are being suggested include some kind of Machine Learning, such as history based grammars and decision tree models (Black et al., 1993; Magerman, 1995), training or inducing statistical grammars (Black, Garside and Leech, 1993; Pereira and Schabes, 1992; Schabes et al., 1993), or other techniques (Bod, 1993).

Consequently, syntactical analysis has become an area with a wide variety of (a) algorithms and methods for learning and parsing, and (b) type of information used for learning and parsing (sometimes referred to as feature set). These methods only could become popular through evaluation methods for parsing systems, such as Bracket Accuracy, Bracket Recall, Sentence Accuracy and Viterbi Score. Some of them were introduced in (Black et al., 1991; Harrison et al., 1991).

These evaluation metrics have a number of problems, and in this paper we argue that they need to be reconsidered, and give a number of suggestions either to overcome those problems or to gain a better understanding of those problems. Particular problems we look at are arbitrary choices in the treebank, errors in the treebank, types of errors made by parsers, and the statistical significance of differences in test scores by parsers.

2 Problems with Evaluation Metrics

Until now a number of problems with evaluation have been pointed out. One well known problem is that measures based only on the absence of crossing errors on sentence level, such as Sentence Accuracy and Viterbi Consistency, are not usable for parsing systems that apply a partial bracketing, since a sparse bracketing improves the score. For example (Lin, 1995) discusses some other problems, but suggests an alternative that is difficult to apply. It is based on transferring constituency trees to dependency trees, but that introduces many ad hoc choices, and treebanks with dependency trees are hardly available.

Also, a treebank usually contains arbitrary choices (besides errors) made by humans, in cases where it was not clear what brackets correctly reflect the syntactical structure of the sentence.

We also mention some less discussed problems. First of all, given a test result such as Bracket Accuracy, it is necessary to know the confidance interval. In other words, if a parsing system scores 81.2% on a test, in what range should we assume the estimate to be? Basically the same problem arises with the statistical significance of the difference between the test score of two different parsers. If one scores 81.2% and the other 82.5%, should we conclude the second one is really doing better?

This is particularly important when developing a parsing system by trying various modifications, and choosing the one that performs the best on a test set. If the differences between scores become too small in relation to the test set, one will just

be making a parser for the test set and the performance will drop as soon as other data is used.

There are several problems for deciding significance for Bracket Accuracy and Bracket Recall. There is a strong variation between brackets, because some brackets are very easy and some are very hard. Also one mistake may lead other mistakes, making them not independent. As an example of the last problem, think of the indicated bracket pair in the sentence *"The dog waited for [his master on the bridge]."* This would probably produce a crossing error, since the treebank would probably contain the pair *"The dog [waited for his master] on the bridge."* The parser is now almost certain to make a second mistake, namely *"The dog waited [for his master on the bridge]."* Consequently two crossing errors are counted, whereas correcting one would imply correcting the other.

In this article we will show that this makes it impossible to calculate the significance in a straightforward way and suggest two solutions.

Another problem is that we only get a very general picture, whereas it would be interesting to know much more details. For example, how many of the bracket-pairs that constituted a crossing error when compared to the treebank would be acceptable to a human? (In other words, how often do arbitrary choices influence the result?) And, how many brackets that the parser produces are not in the treebank nor constitute a crossing error, and how many of those are not acceptable to humans?

Bracket Accuracy is often lower than it should be when the treebank does not indicate all brackets (so-called skeleton parsing). This may also make Bracket Recall seem too low.

In this paper we suggest giving more specific information about test results, and develop methods to estimate the statistical significance for test scores.

3 More Careful Measures

The data resulting from the test may be (a) general data from all bracket pairs, or (b) data on specific structures (i.e. prepositional phrases). The measures we give can be applied to either one.

We suggest performing two types of tests: regular tests and tests with a human check. The regular test should include a number of figures that we describe below, which are much more informative than the usual Bracket Recall or Bracket Precision. The more elaborate one includes a human check on certain items, which not only gives more exact information on the test result, but in particular shows the quality of the regular test. This is particularly useful if the parsing system was made independently from the treebank.

The items for the regular test are listed here. The last four items only apply to a comparison of two parsing systems (for example two modifications of the same system), here referred to as A and B.

- *TTB:* Total Treebank Brackets, number of brackets in the treebank.
- *TPB:* Total Parse Brackets, number of brackets produced by the parsing system.
- *EM:* Exact Match, the number of bracket-pairs produced by the parsing system that are equal to a pair in the treebank.
- *CE:* Crossing Error, the number of bracket-pairs produced by the parsing system that constitute a crossing error against the treebank.
- *SP:* Spurious, number of bracket pairs produced by the parsing system that were not in the treebank but also do not constitute a crossing error.
- *PINH:* Parse-error Inherited, the number of bracket-pairs produced by the parsing system that constitute a crossing error and have a direct parent bracket-pair that also constitutes a crossing error.
- *PNINH:* Parse-error Non-Inherited, the number of bracket-pairs produced by the parsing system that constitute a crossing error, but were not counted for *PINH*.
- *TINH:* Treebank Inherited, the number of bracket-pairs in the treebank that were reproduced by the parsing system and have a direct parent bracket-pair in the treebank that was also reproduced.
- *TNINH:* Treebank Non-Inherited, the number of bracket-pairs in the treebank that were reproduced by the parsing system but were not counted for *TINH*.
- *YY:* Number of brackets in the treebank that were reproduced by A and B.
- *YN:* Number of brackets in the treebank that were reproduced by A but not by B.
- *NY:* Number of brackets in the treebank that were reproduced by B but not by A.
- *NN:* Number of brackets in the treebank that were not reproduced by both A and B.

As an example, we take this 2 sentence test:

Treebank:
[He [walks to [the house]]]
[[The president] [gave [a long speech]]]

Parser:
[He [walks [to [the house]]]]
[The [[president gave] [a [long speech]]]]

The number of exactly matching brackets (*EM*) is 3+2 = 5. The number of crossing errors (*CE*) is

2, both in the second sentence. The rest, $1+1=2$ is spurious (*SP*). Further, *TTB* is 7, *TPB* is 9, *PINH* is 1 and *PNINH* is 1, *TINH* is 1 and *TNINH* is 4.

This already gives more detailed information, but we can take things a step further by having a human evaluate the most important brackets. If the test set is large, it would be undesirable or impossible to have a human evaluate every single bracket, but we can seriously reduce the workload by not considering the exact matching bracket pairs; they are simply marked as 'accepted.' The only result of evaluating these brackets would be a few errors in the treebank, which is often not really worth the trouble (unless the treebank is suspected to contain many errors). This leaves only the crossing errors and spurious brackets to be evaluated.

This leaves a much smaller amount of work, especially if there are many exact matches. Nevertheless we suggest doing a human check only on important tests, such as final evaluations.

In the human evaluation, crossing error and spurious bracket pairs are to be counted as 'acceptable' if they would fit into the correct interpretation using the style of bracketing that the parsing system aims at, ignoring the style of bracketing of the treebank.

The result of this process is that *EM*, *CE* and *SP* will be divided in accepted and rejected, giving six groups. We will refer to them as *EMA*, *EMR*, *CEA*, *CER*, *SPA* and *SPR*. If the check on *EM* is not performed, as we suggest, *EMR* will be 0.

If *YN* and *NY* are both relatively high, this shows that there are structures on which A is better than B and vice versa (the systems 'complement' each other). In that case we would recommend testing on (more) specific structures, because otherwise the general result will be misleading.

4 A Practical Example

To show the difference between the usual evaluation and our evaluation method we give the results for two parsing systems we evaluated in the course of our research. We do not intend to make any particular claims about these parsing systems, nor about the treebank we used (the test was not designed to draw conclusions about the treebank), we only use it to discuss the issues involved in evaluation.

The treebank we used was the EDR corpus (EDR, 1995), a Japanese treebank with mainly newspaper sentences. We compared two versions of a grammar based parsing system developed at our laboratory, using a stochastical grammar to select one parse for every sentence. Having two variations of the same parser, we were interested in the difference between them. We performed a test on 600 sentences from the corpus (which were not used for training).

Our evaluation was as follows:

1. Unrelevant elements such as punctuation are eliminated from both the treebank tree and the parse tree.
2. Next, all (resulting) empty bracket-pairs are removed. This was done recursively, therefore, if removing an empty bracket-pair caused its parent to become empty, the parent is also removed.
3. Double bracket-pairs are removed. For example *"The [[old man]]"* is turned into *"The [old man]"*.
4. The crossing error bracket-pairs and spurious bracket-pairs were evaluated by hand. This took about three person-hours.

In this process one step is missing, we namely wanted to remove trivial brackets before evaluating. In English there is a simple strategy for this: remove all brackets that enclose only one word. In Japanese this is not so easy. Since Japanese is an agglutinating language and words are not separated, it is difficult to say what the 'words' are in the first place. We decided on a certain level to permit brackets, and the tree from the treebank also stopped at some level so that remaining, more precise bracket-pairs were amongst those counted as spurious.

The resulting figures are in table 1 and table 2 gives the comparative items.

Table 1: Sample Test Results

Item	System A		System B	
TTB	11400		11400	
TPB	8671		8771	
EMA	6748	(77.8%)	6858	(78.2%)
EMR	0	(assumed)	0	(assumed)
CEA	204	(2.4%)	182	(2.1%)
CER	690	(8.0%)	611	(7.0%)
SPA	956	(11.0%)	1049	(12.0%)
SPR	73	(0.8%)	71	(0.8%)
PINH	523		470	
PNINH	371		323	
TINH	5212		5426	
TNINH	1536		1432	

Table 2: Comparative Measure Results

YY	6516	(57.2%)
YN	232	(2.0%)
NY	343	(3.1%)
NN	4309	(37.8%)

5 New Measures

We claim that the items listed in the previous paragraph allows a more flexible framework for evaluation. In this paragraph we will show some examples of measures that can be used. They can be calculated with these items so there is no need to discuss every one of them all the time. Table 3 gives the measures and table 4 gives the results in percentages. The measures in the lower part of this table are more directed at the test than at the parsers.

Table 3: Measures

Measure	Calculation
Generation Rate	TPB/TTB
Recall-hard	EMA/TTB
Recall-soft	$(EMA+CEA+SPA)/TTB$
Precision-hard	EMA/TPB
Precision-soft	$(EMA+CEA+SPA)/TPB$
Spuriousness	$(SPA+SPR)/TPB$
Spurious Reject	$SPR/(SPA+SPR)$
False Error	$CEA/(CEA+CER)$
Test Noise	$(EMR+CEA+SPA+SPR)/TPB$
Problem Rate	$(EMR+CEA+SPR)/TPB$
P-inheritance	$PINH/(PINH+PNINH)$
T-inheritance	$TINH/(TINH+TNINH)$

Table 4: Results for Measures

Measure	A	B
Generation Rate	71.5%	72.3%
Recall-hard	59.2%	60.2%
Recall-soft	69.4%	71.0%
Precision-hard	77.8%	78.2%
Precision-soft	91.2%	92.2%
Spuriousness	11.9%	12.8%
Spurious Reject	7.1%	6.3%
False Error	22.8%	23.0%
Test Noise	14.2%	14.8%
Problem Rate	3.2%	2.9%
P-inheritance	58.5%	59.3%
T-inheritance	77.2%	79.1%

The *generation* rate shows that both systems are rather modest in producing brackets.

We give two types of *recall*. We suggest using recall-hard, but when the treebank does not indicate all brackets recall-soft may give an indication of the proper recall.

We also present two types of *precision*. B scores better on *precision-soft*, but there is not much difference for *precision-hard*. This shows that B is better at recall but also generates more spurious brackets. The *spuriousness* also indicates this.

The other measures tell us more about the test itself. A would have been treated slightly favorable without a human check, since relatively more errors go 'undetected.' False Error shows that almost 1 out of 4 crossing errors is not really wrong, which indicates there is much difference in bracketing-style between the treebank and the parsing system. Test Noise shows how many bracket-pairs were not tested properly. Problem Rate shows the real 'myopia' of the test.

The inheritance data shows that in our test crossing errors are often related (P-inheritance). Also, reproducing a particular bracket-pair from the treebank increases the chances on reproducing its parent (T-inheritance).

6 Significance

Things would be easy if we could assume that the chance of applying a bracket is correctly modeled as a binomial experiment. We begin by mentioning two reasons why that is not possible.

- Errors that are related, such as one wrong attachment that causes a number of crossing errors, as was shown in our test by P-inheritance.
- For a binomial process we must assume that the chance on success is the same for every bracket pair. It is not, in fact there are both very easy and very hard bracket pairs, with chances varying from very small to very high.

The significance levels of all differences are worth knowing, but our main interest is the difference between A and B in recall and precision. Because of space limitations we only discuss a strategy for estimating the significance level of the measure recall-hard.

Significance for Recall-Hard First we will check whether the distribution can be modeled properly with a binomial experiment. We do this by looking at the comparative items *YY, YN, NY* and *NN*.

From these values the problem is intuitively clear: there are many easy bracket pairs that both always produce correctly, and many that both almost never produce because they are too hard, or the parsing systems simply never produce a certain type of bracket pair. Also, we have tested two rather similar parsing systems often giving the same answer, after all that is often just what one is interested in because one wants to measure improvement. We will use statistical distributions to confirm this problem occurs, and to find a solution to the significance problem.

We do not have the space to go into the details of the relations between the distributions, but if A and B would behave like a binomial variable with test size N, with P_a and P_b as respective chance on success, the distribution of YY should again be a binomial variable for test size N, with chance $P_{yy} = P_a P_b$. The expected value and variance of YY would be

$$E(YY) = NP_{yy} = NP_aP_b$$

$$V(YY) = NP_{yy}(1 - P_{yy}) = N(P_aP_b)(1 - P_aP_b)$$

For NN the distribution is the same with the opposite probabilities, a binomial variable for test size N and $P_{nn} = (1 - P_a)(1 - P_b)$. If we take $\bar{P}_a = 1 - P_a$ and $\bar{P}_b = 1 - P_b$, the expected value and variance of NN become

$$E(NN) = NP_{nn} = N\bar{P}_a\bar{P}_b$$

$$V(NN) = NP_{nn}(1 - P_{nn}) = N(\bar{P}_a\bar{P}_b)(1 - \bar{P}_a\bar{P}_b)$$

We will later put this to more use, but for now we just use it to conclude that YY is expected to be around 4063, and NN is expected to be around 1851. Using the variation we find that the observed values are both extremely rare, so we can reject the hypothesis that we are comparing two binomial variables.

Our strategy to solve this problem is assuming there are three types of brackets, namely brackets that are almost always reproduced, those that are almost never reproduced, and those that are sometimes reproduced and therefore constitute the 'real test' between the two parsing systems. Note that the first two types do not tell us anything about the difference between the parsing systems. By assuming the rest is similar to a binomial distribution, we can calculate the significance. Of course this assumption simplifies the situation, but it is closer to the truth than assuming the whole test can be modeled by a binominal distribution. And, if this assumption is not justified the whole test is not appropriate without testing on more specific phenomena.

Guessing the Real Test Size The idea behind this method is that some brackets are almost always produced, and some are never, and those should be discarded so the real test remains. Ignoring certain bracket pairs corresponds with the fact that some constituents relate to little and some to much ambiguity, making some suitable for comparison and others not. We look at the number of equal answers to estimate the number of bracket-pairs that were not too easy or too hard.

This is a theoretical operation, thus there is no need to do this in practice. We only need to estimate two parameters: $M1$ being the number of bracket-pairs that is discarded because they are always reproduced, and $M2$ being the number of bracket-pairs discarded because they are not reproduced. We reduce YY by $M1$, and NN by $M2$ (the test size is thus reduced by $M1 + M2$). This indicates an imaginary *real test*, namely the part of the test that really served to compare the parsing systems.

We calculate these quantities by assuming a binomial distribution for the *real test*, and making sure that the corrected values for YY and NN become equal to their expected value. Let

observed YY in real test = $E(YY)$ in real test =
real test size $\times$ P_a in real test $\times$ P_b in real test

then we get

$$YY{-}M1 = \frac{(YY + YN - M1)(YY + NY - M1)}{TTB - M1}$$

We do not give the derivation, but when doing the same for NN and combining the equations the following relation between $M1$ and $M2$ holds:

$$M1 = \frac{NY \times TTB + M2 \times YY - (NY + NN)(YY + NY)}{M2 - NN}$$

There are usually many values for $M1$ and $M2$ that satisfy this condition. In practice $M1$ and $M2$ have to be discrete values, so they often are not satisfying the condition *exactly*, but are close enough.

It may seem logical to find the proper values for $M1$ and $M2$ as a next step, in other words deciding how many brackets were 'too easy' and how many were 'too hard.' But our experience is that there is no need to do that, because we are only interested in the significance level of the difference between A and B, and the significance level is practically the same for all values of $M1$ and $M2$ that satisfy the condition.

As for our test, $M1$ and $M2$ can be, for example, 6234 and 4027 respectively. Whatever value we take, the significance level of the difference between A and B corresponds to being 4.7 standard variations away from the expected value. This means that we can safely conclude that B really performs better than A. The *real test* is a lot smaller, only 1139 bracket pairs, but that is still enough to be meaningful. (If the number of equal answers would be extremely high, the real test size may become too small, indicating the test is meaningless.)

7 Conclusion

We have pointed out that the measures which are currently in use have a number of weaknesses. To list the most important ones, a number of aspects of parsing systems are not measured, treebanks contain arbitrary choices, some errors are not detected and discovering statistical significance is difficult.

The test items and measures we have suggested give a better picture of the specific behaviors of parsing systems. Although not solving the problem of arbitrary choices in the treebank, we can at least find out how much influence this has on the test results by using a human check on important tests. The same goes for other problems, such as errors that are not detected by comparison with the treebank.

We suggest giving the regular items on every test, and sometimes doing a human check to discover the quality of the regular test. The amount of work in the human check can be made small when the recall of the parsing systems is high, by assuming the exact matches are correct.

We have also given a strategy for calculating the significance level of differences in scores on one particular measure, namely recall-hard. This strategy makes it possible to calculate the significance level right away from the test items, not requiring a human check.

This discussion will certainly not be the last on this subject. We have not mentioned some quantities such as the number of sentences with 1 crossing error, 2 crossing errors, and those with many crossing errors. These are of course also useful tools for evaluation. We have also not mentioned the Parse Base (calculated as the geometric mean over all sentences of $\sqrt[n]{p}$, where n is the number of words in the sentence and p the number of parses for the sentence), because that relates to a grammar rather than a parsing system. Nevertheless we feel this will help to improve the evaluation of broad coverage parsing systems.

References

E. Black, S. Abney, D. Flickenger, 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. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the Workshop on Speech and Natural Language, Defense Advanced Research Projects Agency, U.S. Govt.*, pages 306–311.

E. Black, R. Garside, and G. Leech. 1993. *Statistically Driven Computer Grammars of English: The IBM/Lancaster Approach.* Rodopi.

E. Black, F. Jelinek, J. Lafferty, and D. M. Magerman. 1993. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 31–37.

R. Bod. 1993. Using an annotated corpus as a stochastic grammar. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 37–44.

P. Harrison, S. Abney, E. Black, D. Flickenger, C. Gdaniec, R. Grishman, D. Hindle, R. Ingria, M. Marcus, B. Santorini, and T. Strzalkowski. 1991. Evaluating syntax performance of parser/grammars of English. In *Proceedings of the Workshop on Evaluating Natural Language Processing Systems, Association for Computational Linguistics.*

Japan Electronic Dictionary Research Institute, Ltd. 1995. *EDR Electronic Dictionary Technical Guide.*

D. Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1420–1425.

D. M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33d Annual Meeting of the Association for Computational Linguistics*, pages 276-283.

F. Pereira and Y. Schabes. 1992. Inside Outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135.

Y. Schabes, M. Roth, and R. Osborne. 1993. Parsing the wall street journal with the inside-outside algorithm. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 341–347.