Parser Evaluation across Frameworks without Format Conversion

Wai Lok Tam
Interfaculty Initiative in Information Studies
University of Tokyo
7-3-1 Hongo Bunkyo-ku
Tokyo 113-0033 Japan

Yo Sato
Dept of Computer Science
Queen Mary
University of London
Mile End Road
London E1 4NS, U.K.

Yusuke Miyao **Jun-ichi Tsujii**
Dept of Computer Science
University of Tokyo
7-3-1 Hongo Bunkyo-ku
Tokyo 113-0033 Japan

Abstract

In the area of parser evaluation, formats like GR and SD which are based on dependencies, the simplest representation of syntactic information, are proposed as framework-independent metrics for parser evaluation. The assumption behind these proposals is that the simplicity of dependencies would make conversion from syntactic structures and semantic representations used in other formalisms to GR/SD a easy job. But (Miyao et al., 2007) reports that even conversion between these two formats is not easy at all. Not to mention that the 80% success rate of conversion is not meaningful for parsers that boast 90% accuracy. In this paper, we make an attempt at evaluation across frameworks without format conversion. This is achieved by generating a list of names of phenomena with each parse. These names of phenomena are matched against the phenomena given in the gold standard. The number of matches found is used for evaluating the parser that produces the parses. The evaluation method is more effective than evaluation methods which involve format conversion because the generation of names of phenomena from the output of a parser loaded is done by a recognizer that has a 100% success rate of recognizing a phenomenon illustrated by a sentence. The success rate is made possible by the reuse of native codes: codes used for writing the parser and rules of the grammar loaded into the parser.

1 Introduction

The traditional evaluation method for a deep parser is to test it against a list of sentences, each of which is paired with a yes or no. The parser is evaluated on the number of grammatical sentences it accepts and that of ungrammatical sentences it rules out. A problem with this approach to evaluation is that it neither penalizes a parser for getting an analysis wrong for a sentence nor rewards it for getting it right. What prevents the NLP community from working out a universally applicable reward and penalty scheme is the absence of a gold standard that can be used across frameworks. The correctness of an analysis produced by a parser can only be judged by matching it to the analysis produced by linguists in syntactic structures and semantic representations created specifically for the framework on which the grammar is based. A match or a mismatch between analyses produced by different parsers based on different frameworks does not lend itself for a meaningful comparison that leads to a fair evaluation of the parsers. To evaluate two parsers across frameworks, two kinds of methods suggest themselves:

1. Converting an analysis given in a certain format native to one framework to another native to a differernt framework (e.g. converting from a CCG (Steedman, 2000) derivation tree to an HPSG (Pollard and Sag, 1994) phrase structure tree with AVM)

2. Converting analyses given in different framework-specific formats to some simpler format proposed as a framework-independent evaluation schema (e.g. converting from

HPSG phrase structure tree with AVM to GR (Briscoe et al., 2006))

However, the feasibility of either solution is questionable. Even conversion between two evaluation schemata which make use of the simplest representation of syntactic information in the form of dependencies is reported to be problematic by (Miyao et al., 2007).

In this paper, therefore, we propose a different method of parser evaluation that makes no attempt at any conversion of syntactic structures and semantic representations. We remove the need for such conversion by abstracting away from comparison of syntactic structures and semantic representations. The basic idea is to generate a list of names of phenomena with each parse. These names of phenomena are matched against the phenomena given in the gold standard for the same sentence. The number of matches found is used for evaluating the parser that produces the parse.

2 Research Problem

Grammar formalisms differ in many aspects. In syntax, they differ in POS label assignment, phrase structure (if any), syntactic head assignment (if any) and so on, while in semantics, they differ from each other in semantic head assignment, role assignment, number of arguments taken by predicates, etc. Finding a common denominator between grammar formalisms in full and complex representation of syntactic information and semantic information has been generally considered by the NLP community to be an unrealistic task, although some serious attempts have been made recently to offer simpler representation of syntactic information (Briscoe et al., 2006; de Marneffe et al., 2006).

Briscoe et al (2006)'s Grammatical Relation (GR) scheme is proposed as a framework-independent metric for parsing accuracy. The promise of GR lies actually in its dependence on a framework that makes use of simple representation of syntactic information. The assumption behind the usefulness of GR for evaluating the output of parsers is that most conflicts between grammar formalisms would be removed by discarding less useful information carried by complex syntactic or semantic representations used in grammar formalisms during conversion to GRs. But is this assumption true? The answer is not clear. A GR represents syntactic information in the form of a binary relation between a token assigned as the head of the relation and other tokens assigned as its dependents. Notice however that grammar frameworks considerably disagree in the way they assign heads and non-heads. This would raise the doubt that, no matter how much information is removed, there could still remain disagreements between grammar formalisms in what is left.

The simplicity of GR, or other dependency-based metrics, may give the impression that conversion from a more complex representation into it is easier than conversion between two complex representations. In other words, GRs or a similar dependency relation looks like a promising candidate for *lingua franca* of grammar frameworks. However the experiment results given by Miyao et al (2007) show that even conversion into GRs of predicate-argument structures, which is not much more complex than GRs, is not a trivial task. Miyao et al (2007) manage to convert 80% of the predicate-argument structures outputted by their deep parser, ENJU, to GRs correctly. However the parser, with an over 90% accuracy, is too good for the 80% conversion rate. The lesson here is that simplicity of a representation is a different thing from simplicity in converting into that representation.

3 Outline of our Solution

The problem of finding a common denominator for grammar formalisms and the problem of conversion to a common denominator may be best addressed by evaluating parsers without making any attempt to find a common denominator or conduct any conversion. Let us describe briefly in this section how such evaluation can be realised.

3.1 Creating the Gold Standard

The first step of our evaluation method is to construct or find a number of sentences and get an annotator to mark each sentence for the phenomena illustrated by each sentence. After annotating all the sentences in a test suite, we get a list of pairs, whose first element is a sentence ID and second is again a list, one of the corresponding phenomena. This list of pairs is our gold standard. To illustrate, suppose we only get sentence 1 and sentence 2 in our test suite.

(1) John gives a flower to Mary

(2) John gives Mary a flower

Sentence 1 is assigned the phenomena: proper noun, unshifted ditransitive, preposition. Sentence 2 is assigned the phenomena: proper noun, dative-shifted ditransitive. Our gold standard is thus the following list of pairs:

⟨1, ⟨proper noun, unshifted ditransitive, preposition⟩ ⟩,
⟨2, ⟨proper noun,dative-shifted ditransitive⟩ ⟩

3.2 Phenomena Recognition

The second step of our evaluation method requires a small program that recognises what phenomena are illustrated by an input sentence taken from the test suite based on the output resulted from parsing the sentence. The recogniser provides a set of conditions that assign names of phenomena to an output, based on which the output is matched with some framework-specific regular expressions. It looks for hints like the rule being applied at a node, the POS label being assigned to a node, the phrase structure and the role assigned to a reference marker. The names of phenomena assigned to a sentence are stored in a list. The list of phenomena forms a pair with the ID of the sentence, and running the recogniser on multiple outputs obtained by batch parsing (with the parser to be evaluated) will produce a list of such pairs, in exactly the same format as our gold standard. Let us illustrate this with a parser that:

1. assigns a monotransitive verb analysis to ‘give’ and an adjunct analysis to ‘to Mary’ in 1

2. assigns a ditransitive verb analysis to ‘give’ in 2

The list of pairs we obtain from running the recogniser on the results produced by batch parsing the test suite with the parser to be evaluated is the following:

⟨1,⟨proper noun,monotransitive,preposition,adjunct⟩⟩,
⟨2, ⟨proper noun,dative-shifted ditransitive⟩ ⟩

3.3 Performance Measure Calculation

Comparing the two list of pairs generated from the previous steps, we can calculate the precision and recall of a parser using the following formulae:

$$Precision = (\sum_{i=1}^{n} \frac{| R_i \cap A_i |}{| R_i |}) \div n \qquad (1)$$

$$Recall = (\sum_{i=1}^{n} \frac{| R_i \cap A_i |}{| A_i |}) \div n \qquad (2)$$

where list R_i is the list generated by the recogniser for sentence i, list A_i is the list produced by annotators for sentence i, and n the number of sentences in the test suite.

In our example, the parser that does a good job with dative-shifted ditransitives but does a poor job with unshifted ditranstives would have a precision of:

$$(\frac{2}{4} + \frac{2}{2}) \div 2 = 0.75$$

and a recall of:

$$(\frac{2}{3} + \frac{2}{2}) \div 2 = 0.83$$

4 Refining our Solution

In order for the precision and recall given above to be a fair measure, it is necessary for both the recogniser and the annotators to produce an exhaustive list of the phenomena illustrated by a sentence.

But we foresee that annotation errors are likely to be a problem of exhaustive annotation, as is reported in Miyao et al (2007) for the gold standard described in Briscoe et al (2006). Exhaustive annotation procedures require annotators to repeatedly parse a sentence in search for a number of phenomena, which is not the way language is normally processed by humans. Forcing annotators to do this, particularly for a long and complex sentence, is a probable reason for the annotation errors in the gold standard described in (Briscoe et al., 2006).

To avoid the same problem in our creation of a gold standard, we propose to allow non-exhaustive annotation. In fact, our proposal is to limit the number of phenomena assigned to a sentence to one. This decision on which phenomenon to be assigned is made, when the test suite is constructed, for each of the sentences contained in it. Following the traditional approach, we include every sentence in the test suite, along with the core phenomenon we intend to test it on (Lehmann and Oepen, 1996). Thus, Sentence 1 would be assigned the phenomenon of unshifted ditransitive. Sentence 2 would be assigned the phenomenon of dative-shifted ditransitive. This revision of annotation policy removes the need for exhaustive annotation. Instead, annotators are given a new task. They are asked to assign to each sentence *the most common error that a parser is likely to make*. Thus Sentence 1 would be assigned adjunct for such an error. Sentence 2 would be assigned the error of noun-noun compound. Note that these errors are also names of phenomena.

This change in annotation policy calls for a change in the calculation of precision and recall. We leave the recogniser as it is, i.e. to produce an exhaustive list of phenomena, since it is far beyond our remit to render it intelligent enough to select a single, intended, phenomenon. Therefore, an incorrectly low precision would result from a mismatch between the exhaustive list generated by the recogniser and the singleton list produced by annotators for a sentence. For example, suppose we only have sentence 2 in our test suite and the parser correctly analyses the sentence. Our recogniser assigns two phenomena (proper noun, dative-shifted ditransitive) to this sentence as before. This would result in a precision of 0.5.

Thus we need to revise our definition of precision, but before we give our new definition, let us define a truth function t:

$$t(A \supset B) = \begin{cases} 1 & A \quad \supset B \\ 0 & A \cap B \quad = \emptyset \end{cases}$$

$$t(A \cap B = \emptyset) = \begin{cases} 0 & A \cap B \quad \neq \emptyset \\ 1 & A \cap B \quad = \emptyset \end{cases}$$

Now, our new definition of precision and recall is as follows:

$$Precision = \frac{(\sum_{i=1}^{n} \frac{t(R_i \supset AP_i) + t(R_i \cap AN_i = \emptyset)}{2})}{n} \tag{3}$$

$$Recall = \frac{(\sum_{i=1}^{n} \frac{|R_i \cap AP_i|}{|AP_i|})}{n} \tag{4}$$

where list AP_i is the list of phenomena produced by annotators for sentence i, and list AN_i is the list of errors produced by annotators for sentence i.

While the change in the definition of recall is trivial, the new definition of precision requires some explanation. The exhaustive list of phenomena generated by our recogniser for each sentence is taken as a combination of two answers to two questions on the two lists produced by annotators for each sentence. The correct answer to the question on the one-item-list of phenomenon produced by annotators for a sentence is a superset-subset relation between the list generated by our recogniser and the one-item-list of phenomenon produced by annotators. The correct answer to the question on the one-item-list of error produced by annotators for a sentence is the non-existence of any common member between the list generated by our recogniser and the one-item-list of error produced by annotators.

To illustrate, let us try a parser that does a good job with dative-shifted ditransitives but does a poor job with unshifted ditranstives on both 2 and 1. The precision of such a parser would be:

$$(\frac{0}{2} + \frac{2}{2}) \div 2 = 0.5$$

and its recall would be:

$$(\frac{0}{1} + \frac{1}{1}) \div 2 = 0.5$$

5 Experiment

For this abstract, we evaluate ENJU (Miyao, 2006), a released deep parser based on the HPSG formalism and a parser based on the Dynamic Syntax formalism (Kempson et al., 2001) under development against the gold standard given in table 1.

The precision and recall of the two parsers (ENJU and DSPD, which stands for "Dynamic Syntax Parser under Development") are given in table 3:

The experiment that we report here is intended to be an experiment with the evaluation method described in the last section, rather than a very serious attempt to evaluate the two parsers in question. The sentences in table 1 are carefully selected to include both sentences that illustrate core phenomena and sentences that illustrate rarer but more interesting (to linguists) phenomena. But there are too few of them. In fact, the most important number that we have obtained from our experiment is the 100% success rate in recognizing the phenomena given in table 1.

ID	Phenomenon	Error
1	unshifted ditransitive	adjunct
2	dative-shifted ditransitive	noun-noun compound
3	passive	adjunct
4	nominal gerund	verb that takes verbal complement
5	verbal gerund	imperative
6	preposition	particle
7	particle	preposition
8	adjective with extrapolated sentential complement	relative clause
9	inversion	question
10	raising	control

Figure 1: Gold Standard for Parser Evaluation

ID	Sentence
1	John gives a flower to Mary
2	John give Mary a flower
3	John is dumped by Mary
4	Your walking me pleases me
5	Abandoning children increased
6	He talks to Mary
7	John makes up the story
8	It is obvious that John is a fool
9	Hardly does anyone know Mary
10	John continues to please Mary

Figure 2: Sentences Used in the Gold Standard

Measure	ENJU	DSPD
Precision	0.8	0.7
Recall	0.7	0.5

Figure 3: Performance of Two Parsers

6 Discussion

6.1 Recognition Rate

The 100% success rate is not as surprising as it may look. We made use of two recognisers, one for each parser. Each of them is written by the one of us who is somehow involved in the development of the parser whose output is being recognised and familiar with the formalism on which the output is based. This is a clear advantage to format conversion used in other evaluation methods, which is usually done by someone familiar with either the source or the target of conversion, but not both, as such a recogniser only requires knowledge of one formalism and one parser. For someone who is involved in the development of the grammar and of the parser that runs it, it is straightforward to write a recogniser that can make use of the code built into the parser or rules included in the grammar. We can imagine that the 100% recognition rate would drop a little if we needed to recognise a large number of sentences but were not allowed sufficient time to write detailed regular expressions. Even in such a situation, we are confident that the success rate of recognition would be higher than the conversion method.

Note that the effectiveness of our evaluation method depends on the success rate of recognition to the same extent that the conversion method employed in Briscoe et al. (2006) and de Marneff et al. (2006) depends on the conversion rate. Given the high success rate of recognition, we argue that our evaluation method is more effective than any evaluation method which makes use of a format claimed to be framework independent and involves conversion of output based on a different formalism to the proposed format.

6.2 Strictness of Recognition and Precision

There are some precautions regarding the use of our evaluation method. The redefined precision 4 is affected by the strictness of the recogniser. To illustrate, let us take Sentence 8 in Table 1 as an example. ENJU provides the correct phrase structure analysis using the desired rules for this sentence but makes some mistakes in assigning roles to the adjective and the copular verb. The recogniser we write for ENJU is very strict and refuses to assign the phenomenon ‘adjective with extrapolated sentential complement’ based on the output given by ENJU. So ENJU gets 0 point for its answer to the question on the singleton list of phenomenon in the gold standard. But it gets 1 point for its answer to the question on the singleton list of error in the gold standard because it does not go to the other extreme: a relative clause analysis, yielding a 0.5 precision. In this case, this value is fair for ENJU, which produces a partially correct analysis. However, a parser that does not accept the sentence at all, a parser that fails to produce any output or one that erroneously produces an unexpected phenomenon would get the same result: for Sentence 8, such a parser would still get a precision of 0.5, simply because its output does not show that it assigns a relative clause analysis.

We can however rectify this situation. For the lack of parse output, we can add an exception clause to make the parser automatically get a 0 precision (for that sentence). Parsers that make unexpected mistakes are more problematic. An obvious solution to deal with these parsers is to come up with an exhaustive list of mistakes but this is an unrealistic task. For the moment, a temporary but realistic solution would be to expand the list of errors assigned to each sentence in the gold standard and ask annotators to make more intelligent guess of the mistakes that can be made by parsers by considering factors such as similarities in phrase structures or the sharing of sub-trees.

6.3 Combining Evaluation Methods

For all measures, some distortion is unavoidable when applied to exceptional cases. This is true for the classical precision and recall, and our redefined precision and recall is no exception. In the case of the classical precision and recall, the distortion is countered by the inverse relation between them so that even if one is distorted, we can tell from the other that how well (poorly) the object of evaluation performs. Our redefined precision and recall works pretty much the same way.

What motivates us to derive measures so closely related to the classical precision and recall is the ease to combine the redefined precision and recall obtained from our evaluation method with the classical precision and recall obtained from other evaluation methods, so as to obtain a full picture of the performance of the object of evaluation. For example, our redefined precision and recall figures given in Table 3 (or figures obtained from running the same experiment on a larger test set) for ENJU can be combined with the precision and recall figures given in Miyao et al. (2006) for ENJU, which is based on a evaluation method that compares its predicate-argument structures those given in Penn Treebank. Here the precision and recall figures are calculated by assigning an equal weight to every sentence in Section 23 of Penn Treebank. This means that different weights are assigned to different phenomena depending on their frequency in the Penn Treebank. Such assignment of weights may not be desirable for linguists or developers of NLP systems who are targeting a corpus with a very different distribution of phenomena from this particular section of the Penn Treebank. For example, a linguist may wish to assign an equal weight across phenomena or more weights to ‘interesting’ phenomena. A developer of a question-answering system may wish to give more weights to question-related phenomena than other phenomena of less interest which are nevertheless attested more frequently in the Penn Treebank.

In sum, the classical precision and recall figures calculated by assigning equal weight to every sentence could be considered skewed from the perspective of phenomena, whereas our redefined precision and recall figures may be seen as skewed from the frequency perspective. Frequency is relative to domains: less common phenomena in some domains could occur more often in others. Our redefined precision and recall are not only useful for those who want a performance measure skewed the way they want, but also useful for those who want a performance measure as ‘unskewed’ as possible. This may be obtained by combining our redefined precision and recall with the classical precision and recall yielded from other evaluation methods.

7 Conclusion

We have presented a parser evaluation method that addresses the problem of conversion between frameworks by totally removing the need for that kind of conversion. We do some conversion but it is a different sort. We convert the output of a parser to a list of names of phenomena by drawing only on the framework that the parser is based on. It may be inevitable for some loss or inaccuracy to occur during this kind of intra-framework conversion if we try our method on a much larger test set with a much larger variety of longer sentences. But we are confident that the loss would still be far less than any inter-framework conversion work done in other proposals of cross-framework evaluation methods. What we believe to be a more problematic area is the annotation methods we have suggested. At the time we write this paper based on a small-scale experiment, we get slightly better result by asking our annotator to give one phenomenon and one common mistake for each sentence. This may be attributed to the fact that he is a member of the NLP community and hence he gets the knowledge to identify the core phenomena we want to test and the common error that parsers tend to make. If we expand our test set and includes longer sentences, annotators would make more mistakes whether they attempt exhaustive annotation or non-exhaustive annotation. It is difficult to tell whether exhaustive annotation or non-exhaustive annotation would be better for large scale experiments. As future work, we intend to try our evaluation method on more test data to determine which one is better and find ways to improve the one we believe to be better for large scale evaluation.

References

Briscoe, Ted, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of COLING/ACL 2006*.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*.

Kempson, Ruth, Wilfried Meyer-Viol, and Dov Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

Lehmann, Sabine and Stephan Oepen. 1996. TSNLP test suites for natural language processing. In *Proceedings of COLING 1996*.

Miyao, Yusuke, Kenji Sagae, and Junichi Tsujii. 2007. Towards framework-independent evaluation of deep linguistic parsers. In *Proceedings of GEAF 2007*.

Miyao, Yusuke. 2006. *From Linguistic Theory to Syntactic Analysis: Corpus-Oriented Grammar Development and Feature Forest Model*. Ph.D. thesis, University of Tokyo.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications.

Steedman, Mark. 2000. *Syntactic Process*. MIT Press.