All Fragments Count in Parser Evaluation

Jasmijn Bastings, Khalil Sima'an

Institute for Logic, Language and Computation
University of Amsterdam
k.simaan@uva.nl

Abstract

PARSEVAL, the default paradigm for evaluating constituency parsers, calculates parsing success (Precision/Recall) as a function of the number of matching labeled brackets across the test set. Nodes in constituency trees, however, are connected together to reflect important linguistic relations such as predicate-argument and direct-dominance relations between categories. In this paper, we present FREVAL, a generalization of PARSEVAL, where the precision and recall are calculated not only for individual brackets, but also for co-occurring, connected brackets (i.e. fragments). FREVAL fragments precision (FLP) and recall (FLR) interpolate the match across the whole spectrum of fragment sizes ranging from those consisting of individual nodes (labeled brackets) to those consisting of full parse trees. We provide evidence that FREVAL is informative for inspecting relative parser performance by comparing a range of existing parsers.

Keywords: fragments, parsing, evaluation

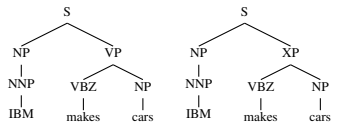
1. Motivation

Current approaches to parsing usually employ a training treebank to learn a statistical parser. The goal of learning is to obtain a parser that can reproduce the test set treebank parses as accurately as possible. The rationale behind this is that the treebank parses themselves are products of trained human annotators and, hence, should serve as the gold standard.

If indeed the ultimate goal of learning statistical parsers from treebanks is to obtain parsers that immitate human parsing capability as represented by a sample in a treebank, then parser evaluation should aim at *measuring the amount match/mismatch between a parse produced by a parser and a parse produced by human annotation*. It is crucial at this point to highlight the difference between this view of parser evaluation and a linguistically-oriented point of view: a linguistically-motivated parser evaluation focuses on linguistically-relevant aspects of parse trees that are crucial for subsequent linguistic processing, e.g., dependency relations might be very important for semantic or other linguistic processing (cf. alternative linguistically relevant proposals (Sampson and Babarczy, 2003; Carroll et al., 1998)). The contrast between linguistic relevance and the statistical view of treebank parsing as a learning problem (with some cognitive relevance) is crucial, because parser output often has other practical uses besides serving as mere input for subsequent linguistic processing, e.g., parsers may serve as target language models in machine translation systems.

Consequently, to evaluate a statistical parser learned from a treebank we need a measure of similarity between its output parse and the human annotated parse in the test set. Such a measure of similarity between two tre could measure different shared aspects between two trees. The PARSEVAL measures (Black et al., 1991) are currently the *de facto* standard for evaluating (English) parser output. To calculate the Precision and Recall, the output trees of a parser are compared to a gold standard, i.e. human-annotated trees in a treebank. A well known treebank in this respect is the Penn Wall Street Journal treebank (Marcus et al., 1993). To facilitate comparison among different parsers, it is common practice to test a parser on section 23 of that corpus and report PARSEVAL F-scores.

PARSEVAL counts how many individual brackets match between a test-tree and a gold-tree, and also whether the test-tree was a complete match or not. However, what PARSEVAL does not count, for example, is whether the matching brackets together constitute a connected unit (e.g. a subtree or paths of direct-dominance relations). Consider for example the following trees:



The trees differ in a single node labeled VP vs XP. This label change ruins the relation of VP with its VBZ verb and object NP, with its parent S and finally, this ruins the subject-verb structure (S (NP VP)). Consequently, different parsers may report very close F-scores coming from completely different parse trees, some of which might be more useful than others.

In this paper we exploit a more elaborate measure of similarity between two trees as the basis for a new parser evaluation measure called FREVAL. FREVAL is a generalization of PARSEVAL from individual nodes to arbitrary size fragments, i.e., subtrees defined as *connected non-empty subgraphs of a tree*. FREVAL computes its final precision (and recall) as a *mixture* of the individually computed precisions (and recalls) for each of the fragment granularity levels.

By employing a mixture of evaluation measures of a range of fragment sizes, FREVAL allows discriminating between parsers performing closely under PARSEVAL but otherwise having completely different kinds of output. As well as subtrees, FREVAL considers paths and parent-child relations also as fragments, thereby accommodating certain as-

pects of leaf-ancestor (Sampson and Babarczy, 2003) and dependency (Carroll et al., 1998) proposals. Interestingly, fragment mixtures has been exploited in statistical parsing, e.g., (Bod et al., 2003; Sima'an, 2000; Bansal and Klein, 2010), but never before for parser evaluation as far as we know.

2. Preliminaries

We start with an overview of PARSEVAL. We assume a test set consisting of sentences $\{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_n\}$ and their corresponding gold-standard trees $T_C = \{\tau_C^1, \tau_C^2, \ldots, \tau_C^n\}$. Now, let the parser output be a set of 'guessed' trees $T_g = \{\tau_g^1, \tau_g^2, \ldots, \tau_g^n\}$. More accurately, τ_C^i and τ_g^i denote the correct and the 'guessed' tree for sentence $\mathcal{U}_i$, respectively. PARSEVAL can be seen to represent a tree τ as a set of labeled constituents:

$$\text{Tree}(\tau) = \{\langle i, X, j\rangle \mid \langle i, X, j\rangle \in \tau\}$$

where $\langle i, X, j\rangle$ stands for a constituent in τ that covers span i to j with label X. $|\text{Tree}(\tau)|$ is the cardinality of the set, in this case the number of brackets/constituents. The PARSEVAL (Labeled Recall, Precision, and Exact Match) are as follows:

$$LR(T_C, T_g) \stackrel{def}{=} \frac{\sum_i |\text{Tree}(\tau_C^i) \cap \text{Tree}(\tau_g^i)|}{\sum_i |\text{Tree}(\tau_C^i)|}$$

$$LP(T_C, T_g) \stackrel{def}{=} \frac{\sum_i |\text{Tree}(\tau_C^i) \cap \text{Tree}(\tau_g^i)|}{\sum_i |\text{Tree}(\tau_g^i|}$$

$$EM(T_C, T_g) \stackrel{def}{=} \frac{\sum_i \delta(T_C^i, T_g^i)}{n}$$

where δ is the Kronecker delta function, returning 1 if the specified trees are equal and 0 otherwise.

3. FREVAL: Beyond Sets of Constituents

We introduce a new representation of trees in terms of their fragments. Let $\max = |\tau|$ denote the number of nodes in a tree τ. A tree τ is represented by a *sequence* of sets of situated fragments

$$\text{Tree}_1(\tau), \text{Tree}_2(\tau), \ldots, \text{Tree}_{\max}(\tau)$$

where for every $1 \leq s \leq \max$, we define $\text{Tree}_s(\tau)$ as the set of all situated fragments φ in τ of size $|\varphi| = s$. A situated fragment $\langle i, \varphi, j\rangle$ is a fragment φ together with the span $\text{span}(\varphi) = \langle i, j\rangle$ that φ covers. More formally,

$$\text{Tree}_s(\tau) \stackrel{def}{=} \{\langle i, \varphi, j\rangle \mid \text{fragment}(\varphi, \tau) \wedge |\varphi| = s \wedge \text{span}(\varphi) = \langle i, j\rangle\}$$

Where $\text{fragment}(\varphi, \tau)$ is True iff φ is a fragment of τ, i.e., a non-empty, connected subgraphs of τ. Note here that we maintain for every fragment size s a separate set $\text{Tree}_s(\tau)$ of situated fragments, i.e., we do not put together fragments of different sizes. This is crucial next because we will calculate over the whole test set a separate precision/recall for each fragments size separately. Had we not done so, the counts of larger fragments would dominate the final precision and recall figures because the number of fragements of a certain size in a tree could be exponential in the number of nodes in the tree.

With this new representation of trees in place, now we define for every fragment size s a separate Labeled Precision (LP_s) and Labeled Recall (LR_s):

$$\text{LP}_s(T_C, T_g) = \frac{\sum_i |\text{Tree}_s(\tau_C^i) \cap \text{Tree}_s(\tau_g^i)|}{\sum_i |\text{Tree}_s(\tau_g^i)|}$$

$$\text{LR}_s(T_C, T_g) = \frac{\sum_i |\text{Tree}_s(\tau_C^i) \cap \text{Tree}_s(\tau_g^i)|}{\sum_i |\text{Tree}_s(\tau_C^i)|}$$

The Fragment Labeled Recall (FLR) and Fragment Labeled Precision (FLP) are defined as a linear interpolation over the sequence of different fragment sizes:[1]

$$\text{FLR} = \sum_s \alpha_s \times \text{LR}_s \quad \text{FLP} = \sum_s \alpha_s \times \text{LP}_s$$

where α_s fulfills $\sum_s \alpha_s = 1.0$. If we set $\alpha_1 = 1$ we would obtain standard PARSEVAL LP and LR. And when $\alpha_{max} = 1.0$ this is the Exact Match for the largest trees in the treebank. Hence, FREVAL is the mean of all measures between these two extremes. In the lack of preference for certain fragments sizes over others, we choose to set α_s uniformly over all fragment sizes. Another reasonable setting for α_s could be one that takes the sparsity of the space of fragments of size s into account for smoothing the FREVAL outcomes for larger fragment sizes using results from smaller fragment sizes.

The FREVAL F1 is defined $\text{F1} = \frac{2\times(\text{FLR}\times\text{FLP})}{\text{FLR}+\text{FLP}}$, but we also define F1_s values for every s using the corresponding LR_s and LP_s values.

4. Empirical explorations

Equipped with our new evaluation metric, we ran various popular and new parsers of English on section 23 of the Penn Wall Street Journal tree-bank (Marcus et al., 1993). We cleaned up section 23 by (1) pruning traces subtrees (-NONE-), (2) removing numbers in labels (e.g., NP-2 or NP=2), (3) removing semantic tags (e.g., NP-SBJ), and finally by removing redundant rules (e.g., NP $\rightarrow$ NP).

The tested parsers are Bansal and Klein (2010) with basic refinement (B&K (basic)), Bansal and Klein (2011) all-fragments, shortest-derivation with richer annotations and state-splits (B&K (SDP)); the Berkeley Parser[2] (Petrov et al., 2006); the Charniak parser (Charniak and Johnson, 2005)[3] with and without Johnson reranking; the Collins parser (Collins, 1999) as implemented in (Bikel, 2004)[4];

[1] An alternative is to interpolate log-linearly (e.g., geometric mean) in following of BLEU in machine translation (Papineni et al., 2002). It is not yet clear whether this has added value over simple linear interpolation.

[2] http://code.google.com/p/berkeleyparser/

[3] ftp://ftp.cs.brown.edu/pub/nlparser/

[4] http://www.cis.upenn.edu/~dbikel/software.html

Double-DOP[5] (Sangati and Zuidema, 2011); and the Stanford Parser[6] (Klein and Manning, 2003) with PCFG and with Factored models.

We ran the parsers using the models trained on WSJ sections 02-21.[7] Most parsers provided such a model out-of-the-box, except for the Bikel-Collins parser, which we trained ourselves on the same sections. We evaluated the output of the parsers with PARSEVAL (using the Evalb implementation of Sekine and Collins (1997)) and FREVAL.

Table 1 shows the FREVAL evaluation results for the tested parsers. We can choose to only evaluate up to a certain fragment size, which is reflected in the various columns. In the first column, the maximum fragment size is 1 — single nodes. Therefore, here FREVAL's results are *identical* to the results of PARSEVAL.[8] In the second column, FLR and FLP were calculated on fragments of size $1, 2, \ldots, 15$, and in the third column on fragments of size $1, 2, \ldots, 25$. Finally, in the fourth column *all* fragments are taken into account (in this case, $1, 2, \ldots, 55$). Interestingly, as we take bigger fragments into account the ranking of the parsers changes. For example, when evaluating with just single nodes the Berkeley parser outperforms B&K (basic), but when we also take larger fragments into account (all other columns) B&K (basic) has the upper hand. The same is true for Double-DOP, which is outperformed by Berkeley under PARSEVAL, but finally is on par with it when evaluating using all fragments.

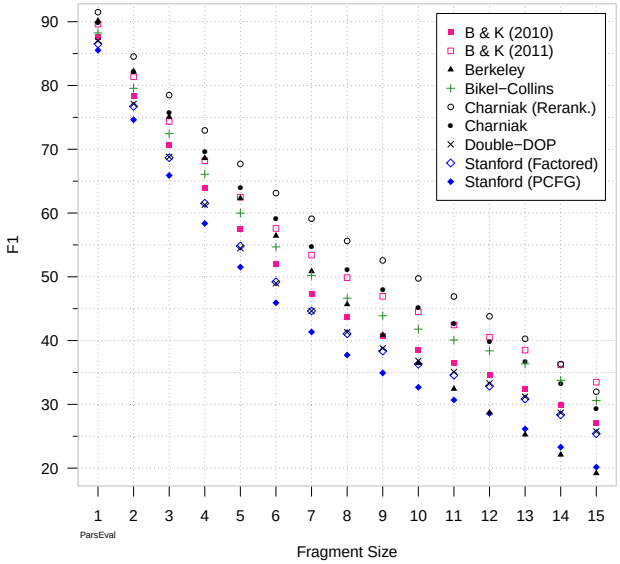


Figure 1: Absolute F_1 measure as function of fragment size up to 15.

[5]http://staff.science.uva.nl/∼fsangati/

[6]http://nlp.stanford.edu/software/lex-parser.shtml

[7]For Double-DOP and the B&K parsers we received the output directly from the respective authors.

[8]In line with the behavior of Evalb, our FREVAL implementation deletes certain nodes (e.g. 'TOP') from its input trees and tries to re-insert pre-terminals in case it deleted one holding a quote in the one tree but not in the other. On top of the default configuration, we also delete nodes with label 'ROOT'.

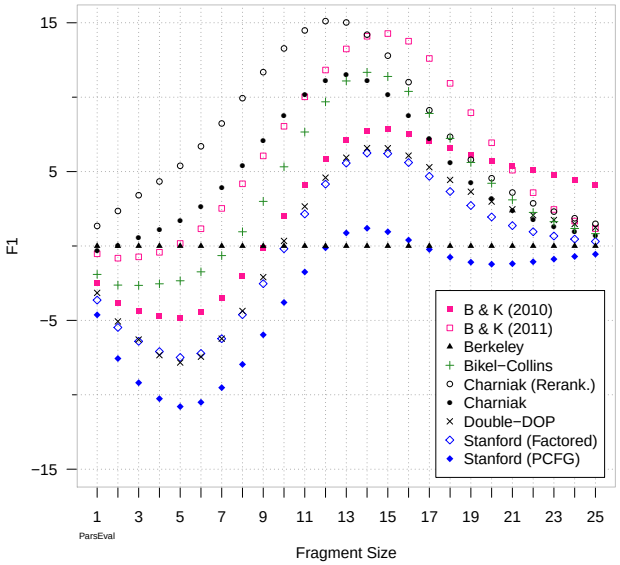


Figure 2: With Berkeley's parser as baseline, a plot of F_1 difference ($Parser - Berkeley$) for every other parser as a function of fragment size.

Figure 1 shows how these performance changes depend on the individual F_1 scores (of LR_s and LP_s) for each fragment size. Intuitively, with uniform α, the parser with the largest "area" under the curve (sum) performs best. For fragments up to size 15, Charniak's reranking parser clearly scores highest. The Berkeley parser, though, scores high on fragment size 1, but starts losing to other parsers as we take larger fragments into account. Figure 2 magnifies the differences between the parsers: it plots for each parser the difference between its F_1 score and the Berkeley parser's corresponding score as a function of fragment size. Some parsers seem to have worse performance for smaller fragment sizes but improve considerably for larger fragment sizes (e.g., Bikel-Collins, Double-DOP, Stanford factored). Both versions of Charniak's parser as well as B&K (SDP) perform well across the whole range of fragment sizes, with the plot of the latter looking almost as a horizontal shift of that of the former.

5. Discussion

The FLR and FLP measures provide an interesting new perspective on parser performance. The Parser ranking, according to the highest FREVAL F1 score, may change along the Fragment size axes. It is easy to see that one node's mismatch can cause the mismatch of a whole lot of bigger fragments. For this very reason, there are hardly any matches for fragments of size 25 and bigger. Moreover, FREVAL, like PARSEVAL, can punish a parser severely for certain mistakes, e.g. attachment errors (see e.g. Kübler and Telljohann (2002)).

The tested parsers differ from one another in various ways. We concentrate on two particular axis of differences (A) The grammar units: Context-Free productions or larger fragments, and (B) Enriched categories by manual refinements, automatic state-splits, head-lexicalization. A single parser employs discriminative reranking (Charniak (reranking)) and most parsers employ horizontal Markovization of

	Evaluated Fragment Sizes											
Fragment sizes	1 (PARSEVAL)			1-15			1-25			All Fragments		
Parser	FLR	FLP	F1	FLR	FLP	F1	FLR	FLP	F1	FLR	FLP	F1
B&K (basic) 2010	87.7	87.6	87.6	49.0	49.8	49.4	35.2	35.2	35.2	16.6	16.7	16.7
B&K (richer) 2011	89.5	89.4	89.5	52.8	58.0	55.3	35.9	44.7	39.8	16.5	20.7	18.4
Berkeley	90.0	90.3	90.2	47.4	51.1	49.2	31.1	35.0	33.0	14.2	16.4	15.3
Bikel-Collins	88.3	88.2	88.2	49.8	55.4	52.5	33.7	41.1	37.0	15.4	18.9	17.0
Charniak	89.7	89.9	89.8	52.8	57.0	54.8	35.7	40.0	37.7	16.3	18.4	17.3
Charniak (Rerank.)	91.2	91.8	91.5	56.8	60.0	58.4	38.7	42.1	40.4	17.8	19.4	18.6
Double-DOP	86.3	87.7	87.0	47.1	48.1	47.6	32.4	33.8	33.1	14.9	15.6	15.3
Stanford (Factored)	86.7	86.4	86.5	46.1	48.7	47.4	31.2	34.3	32.7	14.3	15.8	15.0
Stanford (PCFG)	85.0	86.1	85.5	43.1	44.6	43.8	29.1	29.6	29.3	13.4	13.5	13.4

Table 1: FREVAL results of the tested parsers on WSJ section 23 (all sentences), using various maximum fragment sizes. B&K stands for Bansal&Klein. FLR and FLP were computed with uniform α weights. When fragment size is exactly 1 (single nodes), FLR and FLP become identical to PARSEVAL's LR and LP scores.

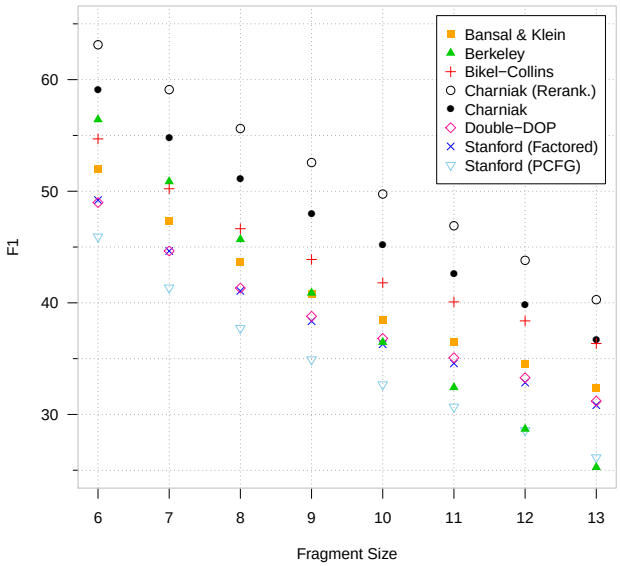


Figure 3: F1 measure (of LR_s and LP_s) for fragment sizes 6-13.

treebank productions, leading to CFG and fragment models with horizontal Markovization. Comparing the two B&K versions (basic and SDP), the increase in FREVAL F1 scores as larger fragment sizes are included confirms the importance of category refinement; the same holds for the head-lexicalization of categories, where some of the best performing parsers are found (Charniak's, Bikel-Collins); and finally, we see that parsers using fragment models are performing increasingly well along the size line, most notably B&K (basic) already outperforms Berkeley parser for fragment size 1-15, 1-25 and for all sizes, whereas it is far less accurate than Berkeley according to PARSEVAL. And surprisingly, for all fragment sizes, we find that Double-DOP (a selected-fragments parser) performs as well as Berkeley. The mix of head-lexicalization/category refinement with all-fragment modeling B&K (SDP) provides for a parser that outperforms Charniak's (without reranking) for FREVAL values (1-15), (1-25) and all fragments, despite performing slightly less accurately according to PARSEVAL. Adding a fragment-based discriminative reranker on top of Charniak's arrives at the overall best results.

6. Conclusion

Where the original PARSEVAL measure only looks at individual nodes when matching two trees, we present FREVAL, which looks at all the situated fragments in those trees. This causes a radically more fine-grained analysis of the performance of existing parsers. By looking at increasingly larger situated fragments, FREVAL indeed shows what is inside the 'evaluation gap' between the original Precision and Recall scores on the one hand and the Complete Match score on the other. Furthermore, FREVAL helps explore the impact of the different kinds of techniques (CFG rules vs. all-fragments, and refined categories) at a variety of treebank linguistic units.

7. References

Bansal, Mohit and Klein, Dan. (2010). Simple, accurate parsing with an all-fragments grammar. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1098–1107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bansal, Mohit and Klein, Dan. (2011). The surprising variance in shortest-derivation parsing. In *ACL (Short Papers)*, pages 720–725.

Bikel, Daniel M. (2004). Intricacies of collins' parsing model. *Computational Linguistics*, 30(4):479–511.

Black et al., Ezra. (1991). A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the February 1991 DARPA Speech and Natural Language Workshop*, pages 306–311. Morgan Kaufmann.

Bod, R., Scha, R., and Sima'an, K., editors. (2003). *Data Oriented Parsing*. CSLI Publications, Stanford University, Stanford, California, USA.

Carroll, John, Briscoe, Ted, and Sanfilippo, Antonio. (1998). Parser evaluation: a survey and a new proposal. In *Language Resources and Evaluation*.

Charniak, Eugene and Johnson, Mark. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Collins, Michael. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Klein, Dan and Manning, Christopher D. (2003). Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

Kübler, Sandra and Telljohann, Heike. (2002). Towards a dependency-oriented evaluation for partial parsing. In *LREC 2002 Workshop Proceedings*.

Marcus, Mitchell P., Marcinkiewicz, Mary Ann, and Santorini, Beatrice. (1993). Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19:313–330, June.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Petrov, Slav, Barrett, Leon, Thibaux, Romain, and Klein, Dan. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.

Sampson, Geoffrey and Babarczy, Anna. (2003). A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9:365–380, December.

Sangati, Federico and Zuidema, Willem. (2011). Accurate Parsing with Compact Tree-Substitution Grammars: Double-DOP . In *In proceedings of EMNLP*, pages 1–12, June.

Sekine, S. and Collins, M. J. (1997). Evalb bracket scoring program.

Sima'an, K. (2000). Tree-gram Parsing: Lexical Dependencies and Structural Relations. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, pages 53–60, Hong Kong, China.