Joint Evaluation of Morphological Segmentation and Syntactic Parsing

Reut Tsarfaty Joakim Nivre Evelina Andersson
Box 635, 751 26, Uppsala University, Uppsala, Sweden
tsarfaty@stp.lingfil.uu.se, {*joakim.nivre, evelina.andersson*}*@lingfil.uu.se*

Abstract

We present novel metrics for parse evaluation in joint segmentation and parsing scenarios where the gold sequence of terminals is not known in advance. The protocol uses distance-based metrics defined for the space of trees over lattices. Our metrics allow us to precisely quantify the performance gap between non-realistic parsing scenarios (assuming gold segmented and tagged input) and realistic ones (not assuming gold segmentation and tags). Our evaluation of segmentation and parsing for Modern Hebrew sheds new light on the performance of the best parsing systems to date in the different scenarios.

1 Introduction

A parser takes a sentence in natural language as input and returns a syntactic parse tree representing the sentence's human-perceived interpretation. Current state-of-the-art parsers assume that the space-delimited words in the input are the basic units of syntactic analysis. Standard evaluation procedures and metrics (Black et al., 1991; Buchholz and Marsi, 2006) accordingly assume that the yield of the parse tree is known in advance. This assumption breaks down when parsing morphologically rich languages (Tsarfaty et al., 2010), where every space-delimited word may be effectively composed of multiple morphemes, each of which having a distinct role in the syntactic parse tree. In order to parse such input the text needs to undergo *morphological segmentation*, that is, identifying the morphological segments of each word and assigning the corresponding part-of-speech (PoS) tags to them.

Morphologically complex words may be highly ambiguous and in order to segment them correctly their analysis has to be disambiguated. The multiple morphological analyses of input words may be represented via a lattice that encodes the different segmentation possibilities of the entire word sequence. One can either select a segmentation path prior to parsing, or, as has been recently argued, one can let the parser pick a segmentation jointly with decoding (Tsarfaty, 2006; Cohen and Smith, 2007; Goldberg and Tsarfaty, 2008; Green and Manning, 2010). If the selected segmentation is different from the gold segmentation, the gold and parse trees are rendered incomparable and standard evaluation metrics break down. Evaluation scenarios restricted to gold input are often used to bypass this problem, but, as shall be seen shortly, they present an overly optimistic upper-bound on parser performance.

This paper presents a full treatment of evaluation in different parsing scenarios, using distance-based measures defined for trees over a shared common denominator defined in terms of a lattice structure. We demonstrate the informativeness of our metrics by evaluating joint segmentation and parsing performance for the Semitic language Modern Hebrew, using the best performing systems, both constituency-based and dependency-based (Tsarfaty, 2010; Goldberg, 2011a). Our experiments demonstrate that, for all parsers, significant performance gaps between realistic and non-realistic scenarios crucially depend on the kind of information initially provided to the parser. The tool and metrics that we provide are completely general and can straightforwardly apply to other languages, treebanks and different tasks.

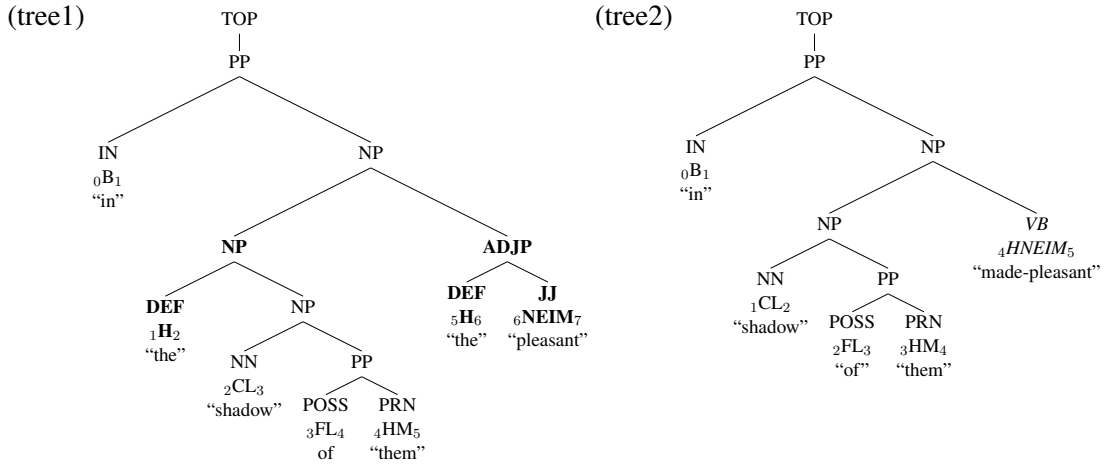

Figure 1: A correct tree (tree1) and an incorrect tree (tree2) for "BCLM HNEIM", indexed by terminal boundaries. Erroneous nodes in the parse hypothesis are marked in *italics*. Missing nodes from the hypothesis are marked in **bold**.

2 The Challenge: Evaluation for MRLs

In morphologically rich languages (MRLs) substantial information about the grammatical relations between entities is expressed at word level using inflectional affixes. In particular, in MRLs such as Hebrew, Arabic, Turkish or Maltese, elements such as determiners, definite articles and conjunction markers appear as affixes that are appended to an open-class word. Take, for example the Hebrew word-token BCLM,[1] which means "in their shadow". This word corresponds to five distinctly tagged elements: B ("in"/IN), H ("the"/DEF), CL ("shadow"/NN), FL ("of"/POSS), HM ("they"/PRN). Note that morphological segmentation is not the inverse of concatenation. For instance, the overt definite article H and the possessor FL show up only in the analysis.

The correct parse for the Hebrew phrase "BCLM HNEIM" is shown in Figure 1 (tree1), and it presupposes that these segments can be identified and assigned the correct PoS tags. However, morphological segmentation is non-trivial due to massive word-level ambiguity. The word BCLM, for instance, can be segmented into the noun BCL ("onion") and M (a genitive suffix, "of them"), or into the prefix B ("in") followed by the noun CLM ("image").[2] The multitude of morphological analyses may be encoded in a lattice structure, as illustrated in Figure 2.

[1]We use the Hebrew transliteration in Sima'an et al. (2001).

[2]The complete set of analyses for this word is provided in Goldberg and Tsarfaty (2008). Examples for similar phenomena in Arabic may be found in Green and Manning (2010).

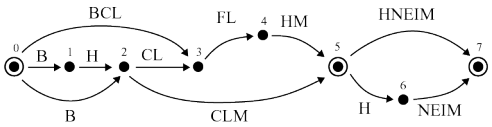

Figure 2: The morphological segmentation possibilities of BCLM HNEIM. Double-circles are word boundaries.

In practice, a statistical component is required to decide on the correct morphological segmentation, that is, to pick out the correct path through the lattice. This may be done based on linear local context (Adler and Elhadad, 2006; Shacham and Wintner, 2007; Bar-haim et al., 2008; Habash and Rambow, 2005), or jointly with parsing (Tsarfaty, 2006; Goldberg and Tsarfaty, 2008; Green and Manning, 2010). Either way, an incorrect morphological segmentation hypothesis introduces errors into the parse hypothesis, ultimately providing a parse tree which spans a different yield than the gold terminals. In such cases, existing evaluation metrics break down.

To understand why, consider the trees in Figure 1. Metrics like PARSEVAL (Black et al., 1991) calculate the harmonic means of precision and recall on labeled spans $\langle i, label, j\rangle$ where i, j are terminal boundaries. Now, the NP dominating "shadow of them" has been identified and labeled correctly in tree2, but in tree1 it spans $\langle 2, \text{NP}, 5\rangle$ and in tree2 it spans $\langle 1, \text{NP}, 4\rangle$. This node will then be counted as an error for tree2, along with its dominated and dominating structure, and PARSEVAL will score 0.

A generalized version of PARSEVAL which considers i, j character-based indices instead of terminal boundaries (Tsarfaty, 2006) will fail here too, since the missing overt definite article H will cause similar misalignments. Metrics for dependency-based evaluation such as ATTACHMENT SCORES (Buchholz and Marsi, 2006) suffer from similar problems, since they assume that both trees have the same nodes — an assumption that breaks down in the case of incorrect morphological segmentation.

Although great advances have been made in parsing MRLs in recent years, this evaluation challenge remained unsolved.[3] In this paper we present a solution to this challenge by extending TEDEVAL (Tsarfaty et al., 2011) for handling trees over lattices.

3 The Proposal: Distance-Based Metrics

Input and Output Spaces We view the joint task as a structured prediction function $h : \mathcal{X} \rightarrow \mathcal{Y}$ from input space $\mathcal{X}$ onto output space $\mathcal{Y}$. Each element $x \in \mathcal{X}$ is a sequence $x = w_1, \ldots, w_n$ of space-delimited words from a set $\mathcal{W}$. We assume a lexicon LEX, distinct from $\mathcal{W}$, containing pairs of segments drawn from a set $\mathcal{T}$ of terminals and PoS categories drawn from a set $\mathcal{N}$ of nonterminals.

$$\text{LEX} = \{\langle s, p\rangle | s \in \mathcal{T}, p \in \mathcal{N}\}$$

Each word w_i in the input may admit multiple morphological analyses, constrained by a language-specific morphological analyzer MA. The morphological analysis of an input word $\text{MA}(w_i)$ can be represented as a lattice L_i in which every arc corresponds to a lexicon entry $\langle s, p\rangle$. The morphological analysis of an input sentence x is then a lattice L obtained through the concatenation of the lattices $L_1, \ldots, L_n$ where $\text{MA}(w_1) = L_1, \ldots, \text{MA}(w_n) = L_n$. Now, let $x = w_1, \ldots, w_n$ be a sentence with a morphological analysis lattice $\text{MA}(x) = L$. We define the output space $\mathcal{Y}_{\text{MA}(x)=L}$ for h (abbreviated $\mathcal{Y}_L$), as the set of linearly-ordered labeled trees such that the yield of LEX entries $\langle s_1, p_1\rangle, \ldots, \langle s_k, p_k\rangle$ in each tree (where $s_i \in \mathcal{T}$ and $p_i \in \mathcal{N}$, and possibly $k \neq n$) corresponds to a path through the lattice L.

[3] A tool that could potentially apply here is SParseval (Roark et al., 2006). But since it does not respect word-boundaries, it fails to apply to such lattices. Cohen and Smith (2007) aimed to fix this, but in their implementation syntactic nodes internal to word boundaries may be lost without scoring.

Edit Scripts and Edit Costs We assume a set $\mathcal{A}$=$\{\text{ADD}(c, i, j), \text{DEL}(c, i, j), \text{ADD}(\langle s, p\rangle, i, j), \text{DEL}(\langle s, p\rangle, i, j)\}$ of edit operations which can add or delete a labeled node $c \in \mathcal{N}$ or an entry $\langle s, p\rangle \in$ LEX which spans the states i, j in the lattice L. The operations in $\mathcal{A}$ are properly constrained by the lattice, that is, we can only add and delete lexemes that belong to LEX, and we can only add and delete them where they can occur in the lattice. We assume a function $\text{C}(a) = 1$ assigning a unit cost to every operation $a \in \mathcal{A}$, and define the cost of a sequence $\langle a_1, \ldots, a_m\rangle$ as the sum of the costs of all operations in the sequence $\text{C}(\langle a_1, ..., a_m\rangle) = \sum_{i=1}^{m} \text{C}(a_i)$. An *edit script* $\text{ES}(y_1, y_2) = \langle a_1, \ldots, a_m\rangle$ is a sequence of operations that turns y_1 into y_2. The *tree-edit distance* is the minimum cost of any edit script that turns y_1 into y_2 (Bille, 2005).

$$\text{TED}(y_1, y_2) = \min_{\text{ES}(y_1, y_2)} \text{C}(\text{ES}(y_1, y_2))$$

Distance-Based Metrics The error of a predicted structure p with respect to a gold structure g is now taken to be the TED cost, and we can turn it into a score by normalizing it and subtracting from a unity:

$$\text{TEDEVAL}(p, g) = 1 - \frac{\text{TED}(p, g)}{|p| + |g| - 2}$$

The term $|p| + |g| - 2$ is a normalization factor defined in terms of the worst-case scenario, in which the parser has only made incorrect decisions. We would need to delete all lexemes and nodes in p and add all the lexemes and nodes of g, except for roots.

An Example Both trees in Figure 1 are contained in $\mathcal{Y}_L$ for the lattice L in Figure 2. If we replace terminal boundaries with lattice indices from Figure 2, we need 6 edit operations to turn tree2 into tree1 (deleting the nodes in *italic*, adding the nodes in **bold**) and the evaluation score will be $\text{TEDEVAL}(\text{tree2,tree1}) = 1 - \frac{6}{14+10-2} = 0.7273$.

4 Experiments

We aim to evaluate state-of-the-art parsing architectures on the morphosyntactic disambiguation of Hebrew texts in three different parsing scenarios: *(i) Gold:* assuming gold segmentation and PoS-tags, *(ii) Predicted:* assuming only gold segmentation, and *(iii) Raw:* assuming unanalyzed input text.

		SEGEVAL	PARSEVAL	TEDEVAL
Gold	PS	U: 100.00 L: 100.00	L: 88.75	U: 94.35 L: 93.39
Predicted	PS	U: 100.00 L: 90.85	L: 82.30	U: 92.92 L: 86:26
Raw	PS	U: 96.42 L: 84.54	N/A	U: 88.47 L: 80.67
Gold	RR	U: 100.00 L: 100.00	L: 83.93	U: 94.34 L: 92.45
Predicted	RR	U: 100.00 L: 91.69	L: 78.93	U: 92.82 L: 85.83
Raw	RR	U: 96.03 L: 86.10	N/A	U: 87.96 L: 79.46

Table 1: Phrase-Structure based results for the Berkeley Parser trained on bare-bone trees (PS) and relational-realizational trees (RR). We parse all sentences in the dev set. RR extra decoration is removed prior to evaluation.

		SEGEVAL	ATTSCORES	TEDEVAL
Gold	MP	100.00	U: 83.59	U: 91.76
Predicted	MP	100.00	U: 82.00	U: 91.20
Raw	MP	95.07	N/A	U: 87.03
Gold	EF	100.00	U: 84.68	U: 92.25
Predicted	EF	100.00	U: 83.97	U: 92:02
Raw	EF	95.07	N/A	U: 87.75

Table 2: Dependency parsing results by MaltParser (MP) and EasyFirst (EF), trained on the treebank converted into unlabeled dependencies, and parsing the entire dev-set.

For constituency-based parsing we use two models trained by the Berkeley parser (Petrov et al., 2006) one on phrase-structure (PS) trees and one on relational-realizational (RR) trees (Tsarfaty and Sima'an, 2008). In the *raw* scenario we let a lattice-based parser choose its own segmentation and tags (Goldberg, 2011b). For dependency parsing we use MaltParser (Nivre et al., 2007b) optimized for Hebrew by Ballesteros and Nivre (2012), and the Easy-First parser of Goldberg and Elhadad (2010) with the features therein. Since these parsers cannot choose their own tags, automatically predicted segments and tags are provided by Adler and Elhadad (2006).

We use the standard split of the Hebrew treebank (Sima'an et al., 2001) and its conversion into unlabeled dependencies (Goldberg, 2011a). We use PARSEVAL for evaluating phrase-structure trees, ATTACHSCORES for evaluating dependency trees, and TEDEVAL for evaluating all trees in all scenarios. We implement SEGEVAL for evaluating segmentation based on our TEDEVAL implementation, replacing the tree distance and size with string terms.

Table 1 shows the constituency-based parsing results for all scenarios. All of our results confirm that gold information leads to much higher scores. TEDEVAL allows us to precisely quantify the drop in accuracy from *gold* to *predicted* (as in PARSEVAL) and than from *predicted* to *raw* on a single scale. TEDEVAL further allows us to scrutinize the contribution of different sorts of information. Unlabeled TEDEVAL shows a greater drop when moving from *predicted* to *raw* than from *gold* to *predicted*, and for labeled TEDEVAL it is the other way round. This demonstrates the great importance of gold tags which provide morphologically disambiguated information for identifying phrase content.

Table 2 shows that dependency parsing results confirm the same trends, but we see a much smaller drop when moving from *gold* to *predicted*. This is due to the fact that we train the parsers for *predicted* on a treebank containing *predicted* tags. There is however a great drop when moving from *predicted* to *raw*, which confirms that evaluation benchmarks on gold input as in Nivre et al. (2007a) do not provide a realistic indication of parser performance.

For all tables, TEDEVAL results are on a similar scale. However, results are not yet comparable across parsers. RR trees are flatter than bare-bone PS trees. PS and DEP trees have different label sets. Cross-framework evaluation may be conducted by combining this metric with the cross-framework protocol of Tsarfaty et al. (2012).

5 Conclusion

We presented distance-based metrics defined for trees over lattices and applied them to evaluating parsers on joint morphological and syntactic disambiguation. Our contribution is both technical, providing an evaluation tool that can be straightforwardly applied for parsing scenarios involving trees over lattices,[4] and methodological, suggesting to evaluate parsers in all possible scenarios in order to get a realistic indication of parser performance.

Acknowledgements

We thank Shay Cohen, Yoav Goldberg and Spence Green for discussion of this challenge. This work was supported by the Swedish Science Council.

[4]The tool can be downloaded `http://stp.ling.uu.se/~tsarfaty/unipar/index.html`

References

Meni Adler and Michael Elhadad. 2006. An unsupervised morpheme-based HMM for Hebrew morphological disambiguation. In *Proceedings of COLING-ACL*.

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: A system for MaltParser optimization. Istanbul.

Roy Bar-haim, Khalil Sima'an, and Yoad Winter. 2008. Part-of-speech tagging of Modern Hebrew text. *Natural Language Engineering*, 14(2):223–251.

Philip Bille. 2005. A survey on tree-edit distance and related. problems. *Theoretical Computer Science*, 337:217–239.

Ezra Black, Steven P. Abney, D. Flickenger, Claudia Gdaniec, Ralph Grishman, P. Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith L. Klavans, Mark Liberman, Mitchell P. Marcus, Salim Roukos, Beatrice Santorini, and Tomek Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of the DARPA Workshop on Speech and Natural Language*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X*, pages 149–164.

Shay B. Cohen and Noah A. Smith. 2007. Joint morphological and syntactic disambiguation. In *Proceedings of EMNLP-CoNLL*, pages 208–217.

Yoav Goldberg and Michael Elhadad. 2010. Easy-first dependency parsing of Modern Hebrew. In *Proceedings of NAACL/HLT workshop on Statistical Parsing of Morphologically Rich Languages*.

Yoav Goldberg and Reut Tsarfaty. 2008. A single framework for joint morphological segmentation and syntactic parsing. In *Proceedings of ACL*.

Yoav Goldberg. 2011a. *Automatic Syntactic Processing of Modern Hebrew*. Ph.D. thesis, Ben-Gurion University of the Negev.

Yoav Goldberg. 2011b. Joint morphological segmentation and syntactic parsing using a PCFGLA lattice parser. In *Proceedings of ACL*.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of COLING*.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of ACL*.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007a. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Joakim Nivre, Jens Nilsson, Johan Hall, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007b. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(1):1–41.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of ACL*.

Brian Roark, Mary Harper, Eugene Charniak, Bonnie Dorr C, Mark Johnson D, Jeremy G. Kahn E, Yang Liu F, Mari Ostendorf E, John Hale H, Anna Krasnyanskaya I, Matthew Lease D, Izhak Shafran J, Matthew Snover C, Robin Stewart K, and Lisa Yung J. 2006. Sparseval: Evaluation metrics for parsing speech. In *Proceesings of LREC*.

Danny Shacham and Shuly Wintner. 2007. Morphological disambiguation of Hebrew: A case study in classifier combination. In *Proceedings of the 2007 Joint Conference of EMNLP-CoNLL*, pages pages 439–447.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a Tree-Bank for Modern Hebrew Text. In *Traitement Automatique des Langues*.

Reut Tsarfaty and Khalil Sima'an. 2008. Relational-Realizational parsing. In *Proceedings of CoLing*.

Reut Tsarfaty, Djame Seddah, Yoav Goldberg, Sandra Kuebler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing for morphologically rich language (SPMRL): What, how and whither. In *Proceedings of the first workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL) at NA-ACL*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2011. Evaluating dependency parsing: Robust and heuristics-free cross-framework evaluation. In *Proceedings of EMNLP*.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Cross-framework evaluation for statistical parsing. In *Proceedings of EACL*.

Reut Tsarfaty. 2006. Integrated morphological and syntactic disambiguation for Modern Hebrew. In *Proceeding of ACL-SRW*.

Reut Tsarfaty. 2010. *Relational-Realizational Parsing*. Ph.D. thesis, University of Amsterdam.