Easy First Dependency Parsing of Modern Hebrew

Yoav Goldberg* and **Michael Elhadad**
Ben Gurion University of the Negev
Department of Computer Science
POB 653 Be'er Sheva, 84105, Israel
{yoavg|elhadad}@cs.bgu.ac.il

Abstract

We investigate the performance of an easy-first, non-directional dependency parser on the Hebrew Dependency treebank. We show that with a basic feature set the greedy parser's accuracy is on a par with that of a first-order globally optimized MST parser. The addition of morphological-agreement feature improves the parsing accuracy, making it on-par with a second-order globally optimized MST parser. The improvement due to the morphological agreement information is persistent both when gold-standard and automatically-induced morphological information is used.

1 Introduction

Data-driven Dependency Parsing algorithms are broadly categorized into two approaches (Kübler et al., 2009). *Transition based* parsers traverse the sentence from left to right[1] using greedy, local inference. *Graph based* parsers use global inference and seek a tree structure maximizing some scoring function defined over trees. This scoring function is usually decomposed over tree edges, or pairs of such edges. In recent work (Goldberg and Elhadad, 2010), we proposed another dependency parsing approach: Easy First, Non-Directional dependency parsing. Like transition based methods, the easy-first method adopts a local, greedy policy. However, it abandons the strict left-to-right processing order, replacing it with an alternative order, which attempts to make easier attachments decisions prior to harder ones. The model was applied to English dependency parsing. It was shown to be more accurate than MALTPARSER, a state-of-the-art transition based parser (Nivre et al., 2006), and near the performance of the first-order MSTPARSER, a graph based parser which decomposes its score over tree edges (McDonald et al., 2005), while being more efficient.

The easy-first parser works by making easier decisions before harder ones. Each decision can be conditioned by structures created by previous decisions, allowing harder decisions to be based on relatively rich syntactic structure. This is in contrast to the globally optimized parsers, which cannot utilize such rich syntactic structures. It was hypothesized in (Goldberg and Elhadad, 2010) that this rich conditioning can be especially beneficial in situations where informative structural information is available, such as in morphologically rich languages.

In this paper, we investigate the non-directional easy-first parser performance on Modern Hebrew, a semitic language with rich morphology, relatively free constituent order, and a small treebank compared to English. We are interested in two main questions: (a) *how well does the non-directional parser perform on Hebrew data?* and (b) *can the parser make effective use of morphological features, such as agreement?*

In (Goldberg and Elhadad, 2009), we describe a newly created Hebrew dependency treebank, and

*Supported by the Lynn and William Frankel Center for Computer Sciences, Ben Gurion University

[1] Strictly speaking, the traversal order is from start to end. This distinction is important when discussing Hebrew parsing, as the Hebrew language is written from right-to-left. We keep the left-to-right terminology throughout this paper, as this is the common terminology. However, "left" and "right" should be interpreted as "start" and "end" respectively. Similarly, "a token to the left" should be interpreted as "the previous token".

Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, pages 103–107,
Los Angeles, California, June 2010.

report results on parsing this corpus with both MALTPARSER and first- and second- order variants of MSTPARSER. We find that the second-order MSTPARSER outperforms the first order MSTPARSER, which in turn outperforms the transition based MALTPARSER. In addition, adding morphological information to the default configurations of these parsers does not improve parsing accuracy. Interestingly, when using automatically induced (rather than gold-standard) morphological information, the transition based MALTPARSER's accuracy improves with the addition of the morphological information, while the scores of both globally optimized parsers drop with the addition of the morphological information.

Our experiments in this paper show that the accuracy of the non-directional parser on the same dataset outperforms the first-order MSTPARSER. With the addition of morphological agreement features, the parser accuracy improves even further, and is on-par with the performance of the second-order MSTPARSER. The improvement due to the morphological information persists also when automatically induced morphological information is used.

2 Modern Hebrew

Some aspects that make Hebrew challenging from a language-processing perspective are:

Affixation Common prepositions, conjunctions and articles are prefixed to the following word, and pronominal elements often appear as suffixes. The segmentation of prefixes and suffixes is often ambiguous and must be determined in a specific context only. In term of dependency parsing, this means that the dependency relations occur not between space-delimited tokens, but instead between sub-token elements which we'll refer to as segments. Furthermore, mistakes in the underlying token segmentations are sure to be reflected in the parsing accuracy.

Relatively free constituent order The ordering of constituents inside a phrase is relatively free. This is most notably apparent in the verbal phrases and sentential levels. In particular, while most sentences follow an SVO order, OVS and VSO configurations are also possible. Verbal arguments can appear before or after the verb, and in many ordering. For example, the message "went from Israel to Thailand" can be expressed as "went to Thailand from Israel", "to Thailand went from Israel", "from Israel went to Thailand", "from Israel to Thailand went" and "to Thailand from Israel went". This results in long and flat VP and S structures and a fair amount of sparsity, which suggests that a dependency representations might be more suitable to Hebrew than a constituency one.

NP Structure and Construct State While constituents order may vary, NP internal structure is rigid. A special morphological marker (*Construct State*) is used to mark noun compounds as well as similar phenomena. This marker, while ambiguous, is essential for analyzing NP internal structure.

Case Marking definite direct objects are marked. The case marker in this case is the function word את appearing before the direct object.[2]

Rich templatic morphology Hebrew has a very productive morphological structure, which is based on a root+template system. The productive morphology results in many distinct word forms and a high out-of-vocabulary rate which makes it hard to reliably estimate lexical parameters from annotated corpora. The root+template system (combined with the unvocalized writing system) makes it hard to guess the morphological analyses of an unknown word based on its prefix and suffix, as usually done in other languages.

Unvocalized writing system Most vowels are not marked in everyday Hebrew text, which results in a very high level of lexical and morphological ambiguity. Some tokens can admit as many as 15 distinct readings, and the average number of possible morphological analyses per token in Hebrew text is 2.7, compared to 1.4 in English (Adler, 2007).

Agreement Hebrew grammar forces morphological agreement between Adjectives and Nouns (which should agree in Gender and Number and definiteness), and between Subjects and Verbs (which should agree in Gender and Number).

[2]The orthographic form את is ambiguous. It can also stand for the noun "shovel" and the pronoun "you"(2nd,fem,sing).

3 Easy First Non Directional Parsing

Easy-First Non Directional parsing is a greedy search procedure. It works with a list of partial structures, $p_i, \ldots, p_k$, which is initialized with the n words of the sentence. Each structure is a head token which is not yet assigned a parent, but may have dependants attached to it. At each stage of the parsing algorithm, two neighbouring partial structures, (p_i, p_{i+1}) are chosen, and one of them becomes the parent of the other. The new dependant is then removed from the list of partial structures. Parsing proceeds until only one partial structure, corresponding to the root of the sentence, is left. The choice of which neighbouring structures to attach is based on a scoring function. This scoring function is learned from data, and attempts to attach more confident attachments before less confident ones. The scoring function makes use of features. These features can be extracted from any pre-built structures. In practice, the features are defined on pre-built structures which are around the proposed attachment point. For complete details about training, features and implementation, refer to (Goldberg and Elhadad, 2010).

4 Experiments

We follow the setup of (Goldberg and Elhadad, 2009).

Data We use the Hebrew dependency treebank described in (Goldberg and Elhadad, 2009). We use Sections 2-12 (sentences 484-5724) as our training set, and report results on parsing the development set, Section 1 (sentences 0-483). As in (Goldberg and Elhadad, 2009), we do not evaluate on the test set in this work.

The data in the treebank is segmented and POS-tagged. Both the parsing models were trained on the gold-standard segmented and tagged data. When evaluating the parsing models, we perform two sets of evaluations. The first one is an oracle experiment, assuming gold segmentation and tagging is available. The second one is a real-world experiment, in which we segment and POS-tag the test-set sentences using the morphological disambiguator described in (Adler, 2007; Goldberg et al., 2008) prior to parsing.

Parsers and parsing models We use our freely available implementation[3] of the non-directional parser.

Evaluation Measure We evaluate the resulting parses in terms of unlabeled accuracy – the percent of correctly identified (child,parent) pairs[4]. To be precise, we calculate:

$$\frac{number\_of\_correctly\_identified\_pairs}{number\_of\_pairs\_in\_gold\_parse}$$

For the oracle case in which the gold-standard token segmentation is available for the parser, this is the same as the traditional unlabeled-accuracy evaluation metric. However, in the real-word setting in which the token segmentation is done automatically, the yields of the gold-standard and the automatic parse may differ, and one needs to decide how to handle the cases in which one or more elements in the identified (child,parent) pair are not present in the gold-standard parse. Our evaluation metric penalizes these cases by regarding them as mistakes.

5 Results

Base Feature Set On the first set of experiments, we used the English feature set which was used in (Goldberg and Elhadad, 2010). Our only modification to the feature set for Hebrew was not to lexicalize prepositions (we found it to work somewhat better due to the smaller treebank size, and Hebrew's rather productive preposition system).

Results of parsing the development set are summarized in Table 1. For comparison, we list the performance of the MALT and MST parsers on the same data, as reported in (Goldberg and Elhadad, 2009).

The case marker את, as well as the morphologically marked *construct nouns*, are covered by all feature models. את is a distinct lexical element in a predictable position, and all four parsers utilize such function word information. Construct nouns are differentiated from non-construct nouns already at the POS tagset level.

All models suffer from the absence of gold POS/morphological information. The easy-first non-directional parser with the basic feature set

[3] http://www.cs.bgu.ac.il/~yoavg/software/nondirparser/

[4] All the results are macro averaged.

(NONDIR) outperforms the transition based MALTPARSER in all cases. It also outperforms the first order MST1 model when gold POS/morphology information is available, and has nearly identical performance to MST1 when automatically induced POS/-morphology information is used.

Additional Morphology Features Error inspection reveals that most errors are semantic in nature, and involve coordination, PP-attachment or main-verb hierarchy. However, some small class of errors reflected morphological disagreement between nouns and adjectives. These errors were either inside a simple NP, or, in some cases, could affect relative clause attachments. We were thus motivated to add specific features encoding morphological agreement to try and avoid this class of errors.

Our features are targeted specifically at capturing noun-adjective morphological agreement.[5] When attempting to score the attachment of two neighbouring structures in the list, p_i and p_{i+1}, we inspect the pairs (p_i, p_{i+1}), (p_i, p_{i+2}), (p_{i-1}, p_{i+1}), (p_{i-2}, p_i), (p_{i+1}, p_{i+2}). For each such pair, in case it is made of a noun and an adjective, we add two features: a binary feature indicating presence or absence of gender agreement, and another binary feature for number agreement.

The last row in Table 1 (NONDIR+MORPH) presents the parser accuracy with the addition of these agreement features. Agreement contributes to the accuracy of the parser, making it as accurate as the second-order MST2. Interestingly, the non-directional model benefits from the agreement features also when automatically induced POS/morphology information is used (going from 75.5% to 76.2%). This is in contrast to the MST parsers, where the morphological features hurt the parser when non-gold morphology is used (75.6 to 73.9 for MST1 and 76.4 to 74.6 for MST2). This can be attributed to either the agreement specific nature of the morphological features added to the non-directional parser, or to the easy-first order of the non-directional parser, and to the fact the morphological features are defined only over structurally close heads at each stage of the parsing process.

[5]This is in contrast to the morphological features used in out-of-the-box MST and MALT parsers, which are much more general.

	Gold Morph/POS	Auto Morph/POS
MALT	80.3	72.9
MALT+MORPH	80.7	73.4
MST1	83.6	75.6
MST1+MORPH	83.6	73.9
MST2	84.3	76.4
MST2+MORPH	84.4	74.6
NONDIR	83.8	75.5
NONDIR+MORPH	84.2	76.2

Table 1: Unlabeled dependency accuracy of the various parsing models.

6 Discussion

We have verified that easy-first, non-directional dependency parsing methodology of (Goldberg and Elhadad, 2010) is successful for parsing Hebrew, a semitic language with rich morphology and a small treebank. We further verified that the model can make effective use of morphological agreement features, both when gold-standard and automatically induced morphological information is provided. With the addition of the morphological agreement features, the non-directional model is as effective as a second-order globally optimized MST model, while being much more efficient, and easier to extend with additional structural features.

While we get adequate parsing results for Hebrew when gold-standard POS/morphology/segmentation information is used, the parsing performance in the realistic setting, in which gold POS/morphology/segmentation information is not available, is still low. We strongly believe that parsing and morphological disambiguation should be done jointly, or at least interact with each other. This is the main future direction for dependency parsing of Modern Hebrew.

References

Meni Adler. 2007. *Hebrew Morphological Disambiguation: An Unsupervised Stochastic Word-based Approach*. Ph.D. thesis, Ben-Gurion University of the Negev, Beer-Sheva, Israel.

Yoav Goldberg and Michael Elhadad. 2009. Hebrew Dependency Parsing: Initial Results. In *Proc. of IWPT*.

Yoav Goldberg and Michael Elhadad. 2010. An efficient algorithm for easy-first non-directional dependency parsing. In *Proc. of NAACL*.

Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008.

EM Can find pretty good HMM POS-Taggers (when given a good start). In *Proc. of ACL*.

Sandra Kübler, Ryan T. McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proc of ACL*.

Joakim Nivre, Johan Hall, and Jens Nillson. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proc. of LREC*.