Overview of the SPMRL 2013 Shared Task: Cross-Framework Evaluation of Parsing Morphologically Rich Languages*

Djamé Seddah[a], Reut Tsarfaty[b], Sandra Kübler[c], Marie Candito[d], Jinho D. Choi[e], Richárd Farkas[f], Jennifer Foster[g], Iakes Goenaga[h], Koldo Gojenola[i], Yoav Goldberg[j], Spence Green[k], Nizar Habash[l], Marco Kuhlmann[m], Wolfgang Maier[n], Joakim Nivre[o], Adam Przepiórkowski[p], Ryan Roth[q], Wolfgang Seeker[r], Yannick Versley[s], Veronika Vincze[t], Marcin Woliński[u], Alina Wróblewska[v], Eric Villemonte de la Clérgerie[w]

[a]U. Paris-Sorbonne/INRIA, [b]Weizman Institute, [c]Indiana U., [d]U. Paris-Diderot/INRIA, [e]IPsoft Inc., [f,t]U. of Szeged, [g]Dublin City U., [h,i]U. of the Basque Country, [j]Bar Ilan U., [k]Stanford U., [l,q]Columbia U., [m,o]Uppsala U., [n]Düsseldorf U., [p,u,v]Polish Academy of Sciences, [r]Stuttgart U., [s]Heidelberg U., [w]INRIA

Abstract

This paper reports on the first shared task on statistical parsing of morphologically rich languages (MRLs). The task features data sets from nine languages, each available both in constituency and dependency annotation. We report on the preparation of the data sets, on the proposed parsing scenarios, and on the evaluation metrics for parsing MRLs given different representation types. We present and analyze parsing results obtained by the task participants, and then provide an analysis and comparison of the parsers across languages and frameworks, reported for gold input as well as more realistic parsing scenarios.

1 Introduction

Syntactic parsing consists of automatically assigning to a natural language sentence a representation of its grammatical structure. Data-driven approaches to this problem, both for constituency-based and dependency-based parsing, have seen a surge of interest in the last two decades. These data-driven parsing approaches obtain state-of-the-art results on the *de facto* standard Wall Street Journal data set (Marcus et al., 1993) of English (Charniak, 2000; Collins, 2003; Charniak and Johnson, 2005; McDonald et al., 2005; McClosky et al., 2006; Petrov et al., 2006; Nivre et al., 2007b; Carreras et al., 2008; Finkel et al., 2008; Huang, 2008; Huang et al., 2010; Zhang and Nivre, 2011; Bohnet and Nivre, 2012; Shindo et al., 2012), and provide a foundation on which many tasks operating on semantic structure (e.g., recognizing textual entailments) or even discourse structure (coreference, summarization) crucially depend.

While progress on parsing English — the main language of focus for the ACL community — has inspired some advances on other languages, it has not, by itself, yielded high-quality parsing for other languages and domains. This holds in particular for morphologically rich languages (MRLs), where important information concerning the predicate-argument structure of sentences is expressed through word formation, rather than constituent-order patterns as is the case in English and other configurational languages. MRLs express information concerning the grammatical function of a word and its grammatical relation to other words at the word level, via phenomena such as inflectional affixes, pronominal clitics, and so on (Tsarfaty et al., 2012c).

The non-rigid tree structures and morphological ambiguity of input words contribute to the challenges of parsing MRLs. In addition, insufficient language resources were shown to also contribute to parsing difficulty (Tsarfaty et al., 2010; Tsarfaty et al., 2012c, and references therein). These challenges have initially been addressed by native-speaking experts using strong in-domain knowledge of the linguistic phenomena and annotation idiosyncrasies to improve the accuracy and efficiency of parsing models. More

*Contact authors: djame.seddah@paris-sorbonne.fr, reut.tsarfaty@weizmann.ac.il, skuebler@indiana.edu

Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages, pages 146–182, Seattle, Washington, USA, 18 October 2013.

recently, advances in PCFG-LA parsing (Petrov et al., 2006) and language-agnostic data-driven dependency parsing (McDonald et al., 2005; Nivre et al., 2007b) have made it possible to reach high accuracy with classical feature engineering techniques in addition to, or instead of, language-specific knowledge. With these recent advances, the time has come for establishing the state of the art, and assessing strengths and weaknesses of parsers across different MRLs.

This paper reports on the first shared task on statistical parsing of morphologically rich languages (the SPMRL Shared Task), organized in collaboration with the 4th SPMRL meeting and co-located with the conference on Empirical Methods in Natural Language Processing (EMNLP). In defining and executing this shared task, we pursue several goals. First, we wish to provide standard training and test sets for MRLs in different representation types and parsing scenarios, so that researchers can exploit them for testing existing parsers across different MRLs. Second, we wish to standardize the evaluation protocol and metrics on morphologically ambiguous input, an under-studied challenge, which is also present in English when parsing speech data or web-based non-standard texts. Finally, we aim to raise the awareness of the community to the challenges of parsing MRLs and to provide a set of strong baseline results for further improvement.

The task features data from nine, typologically diverse, languages. Unlike previous shared tasks on parsing, we include data in both dependency-based and constituency-based formats, and in addition to the *full data* setup (complete training data), we provide a *small* setup (a training subset of 5,000 sentences). We provide three parsing scenarios: one in which gold segmentation, POS tags, and morphological features are provided, one in which segmentation, POS tags, and features are automatically predicted by an external resource, and one in which we provide a lattice of multiple possible morphological analyses and allow for joint disambiguation of the morphological analysis and syntactic structure. These scenarios allow us to obtain the performance upper bound of the systems in lab settings using gold input, as well as the expected level of performance in realistic parsing scenarios — where the parser follows a morphological analyzer and is a part of a full-fledged NLP pipeline.

The remainder of this paper is organized as follows. We first survey previous work on parsing MRLs (§2) and provide a detailed description of the present task, parsing scenarios, and evaluation metrics (§3). We then describe the data sets for the nine languages (§4), present the different systems (§5), and empirical results (§6). Then, we compare the systems along different axes (§7) in order to analyze their strengths and weaknesses. Finally, we summarize and conclude with challenges to address in future shared tasks (§8).

2 Background

2.1 A Brief History of the SPMRL Field

Statistical parsing saw initial success upon the availability of the Penn Treebank (PTB, Marcus et al., 1994). With that large set of syntactically annotated sentences at their disposal, researchers could apply advanced statistical modeling and machine learning techniques in order to obtain high quality structure prediction. The first statistical parsing models were generative and based on treebank grammars (Charniak, 1997; Johnson, 1998; Klein and Manning, 2003; Collins, 2003; Petrov et al., 2006; McClosky et al., 2006), leading to high phrase-structure accuracy.

Encouraged by the success of phrase-structure parsers for English, treebank grammars for additional languages have been developed, starting with Czech (Hajič et al., 2000) then with treebanks of Chinese (Levy and Manning, 2003), Arabic (Maamouri et al., 2004b), German (Kübler et al., 2006), French (Abeillé et al., 2003), Hebrew (Sima'an et al., 2001), Italian (Corazza et al., 2004), Spanish (Moreno et al., 2000), and more. It quickly became apparent that applying the phrase-based treebank grammar techniques is sensitive to language and annotation properties, and that these models are not easily portable across languages and schemes. An exception to that is the approach by Petrov (2009), who trained latent-annotation treebank grammars and reported good accuracy on a range of languages.

The CoNLL shared tasks on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007a) highlighted the usefulness of an alternative linguistic formalism for the development of competitive parsing models. Dependency relations are marked between input tokens directly, and allow the annotation of

non-projective dependencies that are parseable efficiently. Dependency syntax was applied to the description of different types of languages (Tesnière, 1959; Mel'čuk, 2001), which raised the hope that in these settings, parsing MRLs will further improve.

However, the 2007 shared task organizers (Nivre et al., 2007a) concluded that: *"[Performance] classes are more easily definable via language characteristics than via characteristics of the data sets. The split goes across training set size, original data format [...], sentence length, percentage of unknown words, number of dependency labels, and ratio of (C)POSTAGS and dependency labels. The class with the highest top scores contains languages with a rather impoverished morphology."* The problems with parsing MRLs have thus not been solved by dependency parsing, but rather, the challenge has been magnified.

The first event to focus on the particular challenges of parsing MRLs was a dedicated panel discussion co-located with IWPT 2009.[1] Work presented on Hebrew, Arabic, French, and German made it clear that researchers working on non-English parsing face the same overarching challenges: poor lexical coverage (due to high level of inflection), poor syntactic coverage (due to more flexible word ordering), and, more generally, issues of data sparseness (due to the lack of large-scale resources). Additionally, new questions emerged as to the evaluation of parsers in such languages – are the word-based metrics used for English well-equipped to capture performance across frameworks, or performance in the face of morphological complexity? This event provoked active discussions and led to the establishment of a series of SPMRL events for the discussion of shared challenges and cross-fertilization among researchers working on parsing MRLs.

The body of work on MRLs that was accumulated through the SPMRL workshops[2] and hosting ACL venues contains new results for Arabic (Attia et al., 2010; Marton et al., 2013a), Basque (Bengoetxea and Gojenola, 2010), Croatian (Agic et al., 2013), French (Seddah et al., 2010; Candito and Seddah, 2010; Sigogne et al., 2011), German (Rehbein, 2011), Hebrew (Tsarfaty and Sima'an, 2010; Goldberg and Elhadad, 2010a), Hindi (Ambati et al., 2010), Korean (Chung et al., 2010; Choi and Palmer, 2011) and Spanish (Le Roux et al., 2012), Tamil (Green et al., 2012), amongst others. The awareness of the modeling challenges gave rise to new lines of work on topics such as joint morpho-syntactic processing (Goldberg and Tsarfaty, 2008), Relational-Realizational Parsing (Tsarfaty, 2010), EasyFirst Parsing (Goldberg, 2011), PLCFRS parsing (Kallmeyer and Maier, 2013), the use of factored lexica (Green et al., 2013), the use of bilingual data (Fraser et al., 2013), and more developments that are currently under way.

With new models and data, and with lingering interest in parsing non-standard English data, questions begin to emerge, such as: *What is the realistic performance of parsing MRLs using today's methods? How do the different models compare with one another? How do different representation types deal with parsing one particular language? Does the success of a parsing model on a language correlate with its representation type and learning method? How to parse effectively in the face of resource scarcity?* The first step to answering all of these questions is providing standard sets of comparable size, streamlined parsing scenarios, and evaluation metrics, which are our main goals in this SPMRL shared task.

2.2 Where We Are At: The Need for Cross-Framework, Realistic, Evaluation Procedures

The present task serves as the first attempt to standardize the data sets, parsing scenarios, and evaluation metrics for MRL parsing, for the purpose of gaining insights into parsers' performance across languages. Ours is not the first cross-linguistic task on statistical parsing. As mentioned earlier, two previous CoNLL shared tasks focused on cross-linguistic dependency parsing and covered thirteen different languages (Buchholz and Marsi, 2006; Nivre et al., 2007a). However, the settings of these tasks, e.g., in terms of data set sizes or parsing scenarios, made it difficult to draw conclusions about strengths and weaknesses of different systems on parsing MRLs.

A key aspect to consider is the relation between input tokens and tree terminals. In the standard statistical parsing setup, every input token is assumed to be a terminal node in the syntactic parse tree (after deterministic tokenization of punctuation). In MRLs,

[1] `http://alpage.inria.fr/iwpt09/panel.en.html`

[2] See `http://www.spmrl.org/` and related workshops.

morphological processes may have conjoined several words into a single token. Such tokens need to be segmented and their analyses need to be disambiguated in order to identify the nodes in the parse tree. In previous shared tasks on statistical parsing, morphological information was assumed to be known in advance in order to make the setup comparable to that of parsing English. In realistic scenarios, however, morphological analyses are initially unknown and are potentially highly ambiguous, so external resources are used to predict them. Incorrect morphological disambiguation sets a strict ceiling on the expected performance of parsers in real-world scenarios. Results reported for MRLs using gold morphological information are then, at best, optimistic.

One reason for adopting this less-than-realistic evaluation scenario in previous tasks has been the lack of sound metrics for the more realistic scenario. Standard evaluation metrics assume that the number of terminals in the parse hypothesis equals the number of terminals in the gold tree. When the predicted morphological segmentation leads to a different number of terminals in the gold and parse trees, standard metrics such as ParsEval (Black et al., 1991) or Attachment Scores (Buchholz and Marsi, 2006) fail to produce a score. In this task, we use TedEval (Tsarfaty et al., 2012b), a metric recently suggested for joint morpho-syntactic evaluation, in which normalized tree-edit distance (Bille, 2005) on morpho-syntactic trees allows us to quantify the success on the joint task in realistic parsing scenarios.

Finally, the previous tasks focused on dependency parsing. When providing both constituency-based and dependency-based tracks, it is interesting to compare results across these frameworks so as to better understand the differences in performance between parsers of different types. We are now faced with an additional question: how can we compare parsing results across different frameworks? Adopting standard metrics will not suffice as we would be comparing apples and oranges. In contrast, TedEval is defined for both phrase structures and dependency structures through the use of an intermediate representation called function trees (Tsarfaty et al., 2011; Tsarfaty et al., 2012a). Using TedEval thus allows us to explore both dependency and constituency parsing frameworks and meaningfully compare the performance of parsers of different types.

3 Defining the Shared-Task

3.1 Input and Output

We define a parser as a structure prediction function that maps sequences of space-delimited *input tokens* (henceforth, *tokens*) in a language to a set of parse trees that capture valid morpho-syntactic structures in that language. In the case of *constituency parsing*, the output structures are phrase-structure trees. In *dependency parsing*, the output consists of dependency trees. We use the term *tree terminals* to refer to the leaves of a phrase-structure tree in the former case and to the nodes of a dependency tree in the latter.

We assume that input sentences are represented as sequences of tokens. In general, there may be a many-to-many relation between input tokens and tree terminals. Tokens may be identical to the terminals, as is often the case in English. A token may be mapped to multiple terminals assigned their own POS tags (consider, e.g., the token "isn't"), as is the case in some MRLs. Several tokens may be grouped into a single (virtual) node, as is the case with multiword expressions (MWEs) (consider "pomme de terre" for "potatoe"). This task covers all these cases.

In the *standard* setup, all tokens are tree terminals. Here, the task of a parser is to predict a syntactic analysis in which the tree terminals coincide with the tokens. Disambiguating the morphological analyses that are required for parsing corresponds to selecting the correct POS tag and possibly a set of morphological features for each terminal. For the languages Basque, French, German, Hungarian, Korean, Polish, and Swedish, we assume this standard setup.

In the *morphologically complex* setup, every token may be composed of multiple terminals. In this case, the task of the parser is to predict the sequence of tree terminals, their POS tags, and a correct tree associated with this sequence of terminals. Disambiguating the morphological analysis therefore requires splitting the tokens into segments that define the terminals. For the Semitic languages Arabic and Hebrew, we assume this morphologically complex setup.

In the *multiword expression* (MWEs) setup, provided here for French only, groupings of terminals are identified as MWEs (non-terminal nodes in constituency trees, marked heads in dependency trees). Here, the parser is required to predict how terminals are grouped into MWEs on top of predicting the tree.

3.2 Data Sets

The task features nine languages from six language families, from Germanic languages (Swedish and German) and Romance (French) to Slavic (Polish), Koreanic (Korean), Semitic (Arabic, Hebrew), Uralic (Hungarian), and the language isolate Basque.

These languages cover a wide range of morphological richness, with Arabic, Basque, and Hebrew exhibiting a high degree of inflectional and derivational morphology. The Germanic languages, German and Swedish, have greater degrees of phrasal ordering freedom than English. While French is not standardly classified as an MRL, it shares MRLs characteristics which pose challenges for parsing, such as a richer inflectional system than English.

For each contributing language, we provide two sets of annotated sentences: one annotated with labeled phrase-structure trees, and one annotated with labeled dependency trees. The sentences in the two representations are aligned at token and POS levels. Both representations reflect the predicate-argument structure of the same sentence, but this information is expressed using different formal terms and thus results in different tree structures.

Since some of our native data sets are larger than others, we provide the training set in two sizes: *Full* containing all sentences in the standard training set of the language, and *5k* containing the number of sentences that is equivalent in size to our smallest training set (5k sentences). For all languages, the data has been split into sentences, and the sentences are parsed and evaluated independently of one another.

3.3 Parsing Scenarios

In the shared task, we consider three parsing scenarios, depending on how much of the morphological information is provided. The scenarios are listed below, in increasing order of difficulty.

- **Gold:** In this scenario, the parser is provided with unambiguous gold morphological segmentation, POS tags, and morphological features for each input token.
- **Predicted:** In this scenario, the parser is provided with disambiguated morphological segmentation. However, the POS tags and morphological features for each input segment are unknown.

Scenario	Segmentation	PoS+Feat.	Tree
Gold	✓	✓	–
Predicted	✓	1-best	–
Raw (1-best)	1-best	1-best	–
Raw (all)	–	–	–

Table 1: A summary of the parsing and evaluation scenarios. ✓ depicts gold information, – depicts unknown information, to be predicted by the system.

- **Raw:** In this scenario, the parser is provided with morphologically ambiguous input. The morphological segmentation, POS tags, and morphological features for each input token are unknown.

The **Predicted** and **Raw** scenarios require predicting morphological analyses. This may be done using a language-specific morphological analyzer, or it may be done jointly with parsing. We provide inputs that support these different scenarios:

- **Predicted**: Gold treebank segmentation is given to the parser. The POS tags assignment and morphological features are automatically predicted by the parser or by an external resource.
- **Raw (1-best)**: The 1st-best segmentation and POS tags assignment is predicted by an external resource and given to the parser.
- **Raw (all)**: All possible segmentations and POS tags are specified by an external resource. The parser selects jointly a segmentation and a tree.

An overview of all shown in table 1. For languages in which terminals equal tokens, only **Gold** and **Predicted** scenarios are considered. For Semitic languages we further provide input for both **Raw (1-best)** and **Raw (all)** scenarios. [3]

3.4 Evaluation Metrics

This task features nine languages, two different representation types and three different evaluation scenarios. In order to evaluate the quality of the predicted structures in the different tracks, we use a combination of evaluation metrics that allow us to compare the systems along different axes.

[3]The raw Arabic lattices were made available later than the other data. They are now included in the shared task release.

In this section, we formally define the different evaluation metrics and discuss how they support system comparison. Throughout this paper, we will be referring to different evaluation dimensions:

- **Cross-Parser Evaluation in Gold/Predicted Scenarios.** Here, we evaluate the results of different parsers on a single data set in the Gold or Predicted setting. We use standard evaluation metrics for the different types of analyses, that is, **ParsEval** (Black et al., 1991) on phrase-structure trees, and **Labeled Attachment Scores** (LAS) (Buchholz and Marsi, 2006) for dependency trees. Since ParsEval is known to be sensitive to the size and depth of trees (Rehbein and van Genabith, 2007b), we also provide the **Leaf-Ancestor** metric (Sampson and Babarczy, 2003), which is less sensitive to the depth of the phrase-structure hierarchy. In both scenarios we also provide metrics to evaluate the prediction of **MultiWord Expressions**.

- **Cross-Parser Evaluation in Raw Scenarios.** Here, we evaluate the results of different parsers on a single data set in scenarios where morphological segmentation is not known in advance. When a hypothesized segmentation is not identical to the gold segmentation, standard evaluation metrics such as ParsEval and Attachment Scores break down. Therefore, we use **TedEval** (Tsarfaty et al., 2012b), which jointly assesses the quality of the morphological and syntactic analysis in morphologically-complex scenarios.

- **Cross-Framework Evaluation.** Here, we compare the results obtained by a dependency parser and a constituency parser on the same set of sentences. In order to avoid comparing apples and oranges, we use the unlabeled **TedEval** metric, which converts all representation types internally into the same kind of structures, called function trees. Here we use TedEval's cross-framework protocol (Tsarfaty et al., 2012a), which accomodates annotation idiosyncrasies.

- **Cross-Language Evaluation.** Here, we compare parsers for the same representation type across different languages. Conducting a complete and faithful evaluation across languages would require a harmonized universal annotation scheme (possibly along the lines of (de Marneffe and Manning, 2008; McDonald et al., 2013; Tsarfaty, 2013)) or task based evaluation. As an approximation we use unlabeled **TedEval**. Since it is unlabeled, it is not sensitive to label set size. Since it internally uses function-trees, it is less sensitive to annotation idiosyncrasies (e.g., head choice) (Tsarfaty et al., 2011).

The former two dimensions are evaluated on the full sets. The latter two are evaluated on smaller, comparable, test sets. For completeness, we provide below the formal definitions and essential modifications of the evaluation software that we used.

3.4.1 Evaluation Metrics for Phrase Structures

ParsEval The ParsEval metrics (Black et al., 1991) are evaluation metrics for phrase-structure trees. Despite various shortcomings, they are the de-facto standard for system comparison on phrase-structure parsing, used in many campaigns and shared tasks (e.g., (Kübler, 2008; Petrov and McDonald, 2012)). Assume that G and H are phrase-structure gold and hypothesized trees respectively, each of which is represented by a set of tuples (i, A, j) where A is a labeled constituent spanning from i to j. Assume that g is the same as G except that it discards the root, preterminal, and terminal nodes, likewise for h and H. The ParsEval scores define the accuracy of the hypothesis in terms of the normalized size of the intersection of the constituent sets.

$$\begin{aligned} Precision(g,h) &= \frac{|g \cap h|}{|h|} \\ Recall(g,h) &= \frac{|g \cap h|}{|g|} \\ F_1(g,h) &= \frac{2 \times P \times R}{P+R} \end{aligned}$$

We evaluate accuracy on phrase-labels ignoring any further decoration, as it is in standard practices. Evalb, the standard software that implements ParsEval,[4] takes a parameter file and ignores the labels specified therein. As usual, we ignore root and POS labels. Contrary to the standard practice, we do take punctuation into account. Note that, as opposed to the *official version*, we used the SANCL'2012 version[5] modified to actually penalize non-parsed trees.

[4] http://www.spmrl.org/spmrl2013-sharedtask-metrics.html/#Evalb

[5] Modified by Petrov and McDonald (2012) to be less sensitive to punctuation errors.

Leaf-Ancestor The Leaf-Ancestor metric (Sampson and Babarczy, 2003) measures the similarity between the path from each terminal node to the root node in the output tree and the corresponding path in the gold tree. The path consists of a sequence of node labels between the terminal node and the root node, and the similarity of two paths is calculated by using the Levenshtein distance. This distance is normalized by path length, and the score of the tree is an aggregated score of the values for all terminals in the tree (x_t is the leaf-ancestor path of t in tree x).

$$LA(h,g) = \frac{\sum_{t \in yield(g)} Lv(h_t,g_t)/(len(h_t)+len(g_t))}{|yield(g)|}$$

This metric was shown to be less sensitive to differences between annotation schemes in (Kübler et al., 2008), and was shown by Rehbein and van Genabith (2007a) to evaluate trees more faithfully than ParsEval in the face of certain annotation decisions. We used the implementation of Wagner (2012).[6]

3.4.2 Evaluation Metrics for Dependency Structures

Attachment Scores Labeled and Unlabeled Attachment scores have been proposed as evaluation metrics for dependency parsing in the CoNLL shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007a) and have since assumed the role of standard metrics in multiple shared tasks and independent studies. Assume that g, h are gold and hypothesized dependency trees respectively, each of which is represented by a set of arcs (i, A, j) where A is a labeled arc from terminal i to terminal j. Recall that in the gold and predicted settings, $|g| = |h|$ (because the number of terminals determines the number of arcs and hence it is fixed). So Labeled Attachment Score equals precision and recall, and it is calculated as a normalized size of the intersection between the sets of gold and parsed arcs.[7]

$$\begin{aligned} Precision(g,h) &= \frac{|g \cap h|}{|g|} \\ Recall(g,h) &= \frac{|g \cap h|}{|h|} \\ LAS(g,h) &= \frac{|g \cap h|}{|g|} = \frac{|g \cap h|}{|h|} \end{aligned}$$

[6]The original version is available at http://www.grsampson.net/Resources.html, ours at http://www.spmrl.org/spmrl2013-sharedtask-metrics.html/#Leaf.

[7]http://ilk.uvt.nl/conll/software.html.

3.4.3 Evaluation Metrics for Morpho-Syntactic Structures

TedEval The TedEval metrics and protocols have been developed by Tsarfaty et al. (2011), Tsarfaty et al. (2012a) and Tsarfaty et al. (2012b) for coping with non-trivial evaluation scenarios, e.g., comparing parsing results across different frameworks, across representation theories, and across different morphological segmentation hypotheses.[8] Contrary to the previous metrics, which view accuracy as a normalized intersection over sets, TedEval computes the accuracy of a parse tree based on the tree-edit distance between complete trees. Assume a finite set of (possibly parameterized) edit operations $\mathcal{A} = \{a_1....a_n\}$, and a cost function $c : \mathcal{A} \rightarrow 1$. An edit script is the cost of a sequence of edit operations, and the edit distance of g, h is the minimal cost edit script that turns g into h (and vice versa). The normalized distance subtracted from 1 provides the level of accuracy on the task. Formally, the TedEval score on g, h is defined as follows, where ted is the tree-edit distance, and the $|x|$ (size in nodes) discards terminals and root nodes.

$$TedEval(g,h) = 1 - \frac{ted(g,h)}{|g| + |h|}$$

In the gold scenario, we are not allowed to manipulate terminal nodes, only non-terminals. In the raw scenarios, we can add and delete both terminals and non-terminals so as to match both the morphological and syntactic hypotheses.

3.4.4 Evaluation Metrics for Multiword-Expression Identification

As pointed out in section 3.1, the French data set is provided with tree structures encoding both syntactic information and groupings of terminals into MWEs. A given MWE is defined as a continuous sequence of terminals, plus a POS tag. In the constituency trees, the POS tag of the MWE is an internal node of the tree, dominating the sequence of pre-terminals, each dominating a terminal. In the dependency trees, there is no specific node for the MWE as such (the nodes are the terminals). So, the first token of a MWE is taken as the head of the other tokens of the same MWE, with the same label (see section 4.4).

[8]http://www.tsarfaty.com/unipar/download.html.

To evaluate performance on MWEs, we use the following metrics.

- R_MWE, P_MWE, and F_MWE are recall, precision, and F-score over full MWEs, in which a predicted MWE counts as correct if it has the correct span (same group as in the gold data).
- R_MWE +POS, R_MWE +POS, and F_MWE +POS are defined in the same fashion, except that a predicted MWE counts as correct if it has both correct span and correct POS tag.
- R_COMP, R_COMP, and F_COMP are recall, precision and F-score over non-head components of MWEs: a non-head component of MWE counts as correct if it is attached to the head of the MWE, with the specific label that indicates that it is part of an MWE.

4 The SPMRL 2013 Data Sets

4.1 The Treebanks

We provide data from nine different languages annotated with two representation types: phrase-structure trees and dependency trees.[9] Statistics about size, average length, label set size, and other characteristics of the treebanks and schemes are provided in Table 2. Phrase structures are provided in an extended bracketed style, that is, Penn Treebank bracketed style where every labeled node may be extended with morphological features expressed. Dependency structures are provided in the CoNLL-X format.[10]

For any given language, the dependency and constituency treebanks are aligned at the token and terminal levels and share the same POS tagset and morphological features. That is, any form in the CoNLL format is a terminal of the respective bracketed tree. Any CPOS label in the CoNLL format is the preterminal dominating the terminal in the bracketed tree. The FEATS in the CoNLL format are represented as dash-features decorated on the respective pre-terminal node in the bracketed tree. See Figure 1(a)–1(b) for an illustration of this alignment.

For ambiguous morphological analyses, we provide the mapping of tokens to different segmentation possibilities through lattice files. See Figure 1(c) for an illustration, where lattice indices mark the start and end positions of terminals.

For each of the treebanks, we provide a three-way *dev/train/set* split and another train set containing the first 5k sentences of *train (5k)*. This section provides the details of the original treebanks and their annotations, our data-set preparation, including preprocessing and data splits, cross-framework alignment, and the prediction of morphological information in non-gold scenarios.

4.2 The Arabic Treebanks

Arabic is a morphologically complex language which has rich inflectional and derivational morphology. It exhibits a high degree of morphological ambiguity due to the absence of the diacritics and inconsistent spelling of letters, such as Alif and Ya. As a consequence, the Buckwalter Standard Arabic Morphological Analyzer (Buckwalter, 2004; Graff et al., 2009) produces an average of 12 analyses per word.

Data Sets The Arabic data set contains two treebanks derived from the LDC Penn Arabic Treebanks (PATB) (Maamouri et al., 2004b):[11] the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009), a dependency treebank, and the Stanford version of the PATB (Green and Manning, 2010), a phrase-structure treebank. We preprocessed the treebanks to obtain strict token matching between the treebanks and the morphological analyses. This required non-trivial synchronization at the tree token level between the PATB treebank, the CATiB treebank and the morphologically predicted data, using the PATB source tokens and CATiB feature word form as a dual synchronized pivot.

The Columbia Arabic Treebank The Columbia Arabic Treebank (CATiB) uses a dependency representation that is based on traditional Arabic grammar and that emphasizes syntactic case relations (Habash and Roth, 2009; Habash et al., 2007). The CATiB treebank uses the word tokenization of the PATB

[9] Additionally, we provided the data in TigerXML format (Brants et al., 2002) for phrase structure trees containing crossing branches. This allows the use of more powerful parsing formalisms. Unfortunately, we received no submissions for this data, hence we discard them in the rest of this overview.

[10] See `http://ilk.uvt.nl/conll/`.

[11] The LDC kindly provided their latest version of the Arabic Treebanks. In particular, we used PATB 1 v4.1 (Maamouri et al., 2005), PATB 2 v3.1 (Maamouri et al., 2004a) and PATB 3 v3.3. (Maamouri et al., 2009)

	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish
train:									
#Sents	15,762	7,577	14,759	40,472		8,146	23,010	6,578	
#Tokens	589,220	96,368	443,113	719,532		170,141	351,184	68,424	
Lex. Size	36,906	25,136	27,470	77,222		40,782	11,1540	22,911	
Avg. Length	37.38	12.71	30.02	17.77		20.88	15.26	10.40	
Ratio #NT/#Tokens	0.19	0.82	0.34	0.60		0.59	0.60	0.94	
Ratio #NT/#Sents	7.40	10.50	10.33	10.70		12.38	9.27	9.84	
#Non Terminals	22	12	32	25		16	8	34	
#POS tags	35	25	29	54		16	1,975	29	
#total NTs	116,769	79,588	152,463	433,215		100,885	213,370	64,792	
Dep. Label Set Size	9	31	25	43		417	22	27	
train5k:									
#Sents	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000
#Tokens	224,907	61,905	150,984	87,841	128,046	109,987	68,336	52,123	76,357
Lex. Size	19,433	18,405	15,480	17,421	15,975	29,009	29,715	18,632	14,110
Avg. Length	44.98	12.38	30.19	17.56	25.60	21.99	13.66	10.42	15.27
Ratio #NT/#Tokens	0.15	0.83	0.34	0.60	0.42	0.57	0.68	0.94	0.58
Ratio #NT/#Sents	7.18	10.33	10.32	10.58	10.97	12.57	9.29	9.87	8.96
#Non Terminals	22	12	29	23	60	16	8	34	8
#POS Tags	35	25	29	51	50	16	972	29	25
#total NTs	35,909	5,1691	51,627	52,945	54,856	62,889	46,484	49,381	44,845
Dep. Label Set Size	9	31	25	42	43	349	20	27	61
dev:									
#Sents	1,985	948	1,235	5,000	500	1,051	2,066	821	494
#Tokens	73,932	13,851	38,820	76,704	11,301	29,989	30,480	8,600	9,341
Lex. Size	12,342	5,551	6,695	15,852	3,175	10,673	15,826	4,467	2,690
Avg. Length	37.24	14.61	31.43	15.34	22.60	28.53	14.75	10.47	18.90
Ratio #NT/#Tokens	0.19	0.74	0.33	0.63	0.47	047	0.63	0.94	0.48
Ratio #NT/#Sents	7.28	10.92	10.48	9.71	10.67	13.66	9.33	9.90	9.10
#Non Terminals	21	11	27	24	55	16	8	31	8
#POS Tags	32	23	29	50	47	16	760	29	24
#total NTs	14,452	10,356	12,951	48,560	5,338	14,366	19,283	8,132	4,496
Dep. Label Set Size	9	31	25	41	42	210	22	26	59
test:									
#Sents	1959	946	2541	5000	716	1009	2287	822	666
#Tokens	73878	11457	75216	92004	16998	19908	33766	8545	10690
Lex. Size	12254	4685	10048	20149	4305	7856	16475	4336	3112
Avg. Length	37.71	12.11	29.60	18.40	23.74	19.73	14.76	10.39	16.05
Ratio #NT/#Tokens	0.19	0.83	0.34	0.60	0.47	0.62	0.61	0.95	0.57
Ratio #NT/#Sents	7.45	10.08	10.09	11.07	11.17	12.26	9.02	9.94	9.18
#Non Terminals	22	12	30	23	54	15	8	31	8
#POS Tags	33	22	30	52	46	16	809	27	25
#total NTs	14,610	9,537	25,657	55,398	8,001	12,377	20,640	8,175	6,118
Dep. Label Set Size	9	31	26	42	41	183	22	27	56

Table 2: Overview of participating languages and treebank properties. 'Sents' = number of sentences, 'Tokens' = number of raw surface forms. 'Lex. size' and 'Avg. Length' are computed in terms of tagged terminals. 'NT' = non-terminals in constituency treebanks, 'Dep Labels' = dependency labels on the arcs of dependency treebanks. – A more comprehensive table is available at `http://www.spmrl.org/spmrl2013-sharedtask.html/#Prop`.

(a) Constituency Tree

% % every line is a single tree in a bracketed Penn Treebank format

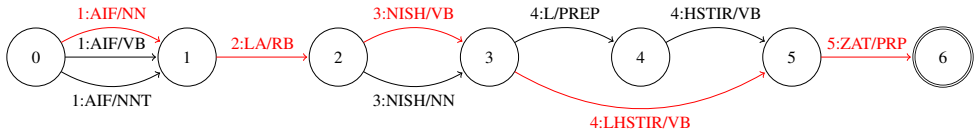
(ROOT (S (NP (NNP-#pers=3|num=sing# John))(VP (VB-#pers=3|num=sing# likes)(NP (NNP-#pers=3|num=sing# Mary)))))

(b) Dependency Tree

%% every line describes a terminal: terminal-id form lemma CPOS FPOS FEATS Head Rel PHead PRel

1	John	John	NNP	NNP	pers=3|num=sing	2	sbj	_	_
2	likes	like	VB	VB	pers=3|num=sing	0	root	_	_
3	Mary	Mary	NNP	NNP	pers=3|num=sing	2	obj	_	_

Input Lattice



%% every line describes a terminal: start-id end-id form lemma CPOS FPOS FEATS token-id

0	1	AIF	AIF	NN	NN	_	1
0	1	AIF	AIF	NNT	NNT	_	1
0	1	AIF	AIF	VB	VB	_	1
1	2	LA	LA	RB	RB	_	2
2	3	NISH	NISH	VB	VB	_	3
2	3	NISH	NISH	NN	NN	_	3
3	5	LHSTIR	HSTIR	VB	VB	_	4
3	4	L	L	PREP	PREP	_	4
4	5	HSTIR	HSTIR	VB	VB	_	4
5	6	ZAT	ZAT	PRP	PRP	_	5

Figure 1: **File formats.** Trees (a) and (b) are aligned constituency and dependency trees for a mockup English example. Boxed labels are shared across the treebanks. Figure (c) shows an ambiguous lattice. The red part represents the yield of the gold tree. For brevity, we use empty feature columns, but of course lattice arcs may carry any morphological features, in the FEATS CoNLL format.

and employs a reduced POS tagset consisting of six tags only: NOM (non-proper nominals including nouns, pronouns, adjectives and adverbs), PROP (proper nouns), VRB (active-voice verbs), VRB-PASS (passive-voice verbs), PRT (particles such as prepositions or conjunctions) and PNX (punctuation). (This stands in extreme contrast with the Buckwalter Arabic tagset (PATB official tagset) which is almost 500 tags.) To obtain these dependency trees, we used the constituent-to-dependency tool (Habash and Roth, 2009). Additional CATiB trees were annotated directly, but we only use the portions that are converted from phrase-structure representation, to ensure that the constituent and dependency yields can be aligned.

The Stanford Arabic Phrase Structure Treebank

In order to stay compatible with the state of the art, we provide the constituency data set with most of the pre-processing steps of Green and Manning (2010), as they were shown to improve baseline performance on the PATB parsing considerably.[12]

To convert the original PATB to preprocessed phrase-structure trees á la Stanford, we first discard all trees dominated by X, which indicates errors and non-linguistic text. At the phrasal level, we collapse unary chains with identical categories like NP → NP. We finally remove all traces, but, unlike Green and Manning (2010), we keep all function tags.

In the original Stanford instance, the pre-terminal morphological analyses were mapped to the shortened *Bies* tag set provided with the treebank (where Determiner markers, "DT", were added to definite noun and adjectives, resulting in 32 POS tags). Here we use the *Kulick* tagset (Kulick et al., 2006) for

[12]Both the corpus split and pre-processing code are available with the Stanford parser at `http://nlp.stanford.edu/projects/arabic.shtml`.

pre-terminal categories in the phrase-structure trees, where the *Bies* tag set is included as a morphological feature (*stanpos*) in our PATB instance.

Adapting the Data to the Shared Task We converted the CATiB representation to the CoNLL representation and added a 'split-from-previous' and 'split-from-next' markers as in LDC's tree-terminal fields.

A major difference between the CATiB treebank and the Stanford treebank lies in the way they handle paragraph annotations. The original PATB contains sequences of annotated trees that belong to a same discourse unit (e.g., paragraph). While the CATiB conversion tool considers each sequence a single parsing unit, the Stanford pre-processor treats each such tree structure rooted at S, NP or Frag as a tree spanning a single sentence. To be compatible with the predicted morphology data which was bootstrapped and trained on the CATiB interpretation, we deterministically modified the original PATB by adding pseudo XP root nodes, so that the Stanford pre-proprecessor will generate the same tree yields as the CATiB treebank.

Another important aspect of preprocessing (often-delegated as a technicality in the Arabic parsing literature) is the normalization of token forms. Most Arabic parsing work used transliterated text based on the schemes proposed by Buckwalter (2002). The transliteration schemes exhibit some small differences, but enough to increase the out-of-vocabulary rate by a significant margin (on top of strictly unknown morphemes). This phenomenon is evident in the morphological analysis lattices (in the predicted dev set there is a 6% OOV rate without normalization, and half a point reduction after normalization is applied, see (Habash et al., 2009b; Green and Manning, 2010)). This rate is much lower for gold tokenized predicted data (with an OOV rate of only 3.66%, similar to French for example). In our data set, all tokens are minimally normalized: no diacritics, no normalization.[13]

Data Splits For the Arabic treebanks, we use the data split recommended by the Columbia Arabic and Dialect Modeling (CADiM) group (Diab et al., 2013). The data of the LDC first three annotated Arabic Treebanks (ATB1, ATB2 and ATB3) were divided into roughly a 10/80/10% dev/train/test split by word volume. When dividing the corpora, document boundaries were maintained. The train5k files are simply the first 5,000 sentences of the training files.

POS Tagsets Given the richness of Arabic morphology, there are multiple POS tag sets and tokenization schemes that have been used by researchers, (see, e.g., Marton et al. (2013a)). In the shared task, we follow the standard PATB tokenization which splits off several categories of orthographic clitics, but not the definite article *Al+*. On top of that, we consider three different POS tag sets with different degrees of granularity: the *Buckwalter* tag set (Buckwalter, 2004), the *Kulick* Reduced Tag set (Kulick et al., 2006), and the *CATiB* tag set (Habash et al., 2009a), considering that granularity of the morphological analyses may affect syntactic processing. For more information see Habash (2010).

Predicted Morphology To prepare input for the Raw scenarios (§3.3), we used the MADA+TOKAN system (Habash et al., 2009b). MADA is a system for *morphological analysis and disambiguation of Arabic*. It can predict the 1-best tokenization, POS tags, lemmas and diacritization in one fell swoop. The MADA output was also used to generate the lattice files for the Raw-all scenario.

To generate input for the gold token / predicted tag input scenario, we used Morfette (Chrupała et al., 2008), a joint lemmatization and POS tagging model based on an averaged perceptron. We generated two tagging models, one trained with the *Buckwalter* tag set, and the other with the *Kulick* tag set. Both were mapped back to the CATiB POS tag set such that all predicted tags are contained in the feature field.[14]

4.3 The Basque Treebank

Basque is an agglutinative language with a high capacity to generate inflected wordforms, with free constituent order of sentence elements with respect to the main verb. Contrary to many other treebanks, the Basque treebank was originally annotated with dependency trees, which were later on converted to constituency trees.

[13] Except for the minimal normalization present in MADA's back-end tools. This script was provided to the participants.

[14] A conversion script from the rich *Buckwalter* tagset to CoNLL-like features was provided to the participants.

The Basque Dependency Treebank (BDT) is a dependency treebank in its original design, due to syntactic characteristics of Basque such as its free word order. Before the syntactic annotation, morphological analysis was performed, using the Basque morphological analyzer of Aduriz et al. (2000). In Basque each lemma can generate thousands of wordforms — differing in morphological properties such as case, number, tense, or different types of subordination for verbs. If only POS category ambiguity is resolved, the analyses remain highly ambiguous.

For the main POS category, there is an average of 1.55 interpretations per wordform, which rises to 2.65 for the full morpho-syntactic information, resulting in an overall 64% of ambiguous wordforms. The correct analysis was then manually chosen.

The syntactic trees were manually assigned. Each word contains its lemma, main POS category, POS subcategory, morphological features, and the labeled dependency relation. Each form indicates morphosyntactic features such as case, number and type of subordination, which are relevant for parsing.

The first version of the Basque Dependency Treebank, consisting of 3,700 sentences (Aduriz et al., 2003), was used in the CoNLL 2007 Shared Task on Dependency Parsing (Nivre et al., 2007a). The current shared task uses the second version of the BDT, which is the result of an extension and redesign of the original requirements, containing 11,225 sentences (150,000 tokens).

The Basque Constituency Treebank (BCT) was created as part of the CESS-ECE project, where the main aim was to obtain syntactically annotated constituency treebanks for Catalan, Spanish and Basque using a common set of syntactic categories. BCT was semi-automatically derived from the dependency version (Aldezabal et al., 2008). The conversion produced complete constituency trees for 80% of the sentences. The main bottlenecks have been sentence connectors and non-projective dependencies which could not be straightforwardly converted into projective tree structures, requiring a mechanism similar to traces in the Penn English Treebank.

Adapting the Data to the Shared Task As the BCT did not contain all of the original non-projective dependency trees, we selected the set of 8,000 matching sentences in both treebanks for the shared task.[15] This implies that around 2k trees could not be generated and therefore were discarded. Furthermore, the BCT annotation scheme does not contain attachment for most of the punctuation marks, so those were inserted into the BCT using a simple lower-left attachment heuristic. The same goes for some connectors that could not be aligned in the first phase.

Predicted Morphology In order to obtain predicted tags for the non-gold scenarios, we used the following pipeline. First, morphological analysis as described above was performed, followed by a disambiguation step. At that point, it is hard to obtain a single interpretation for each wordform, as determining the correct interpretation for each wordform may require knowledge of long-distance elements on top of the free constituency order of the main phrasal elements in Basque. The disambiguation is performed by the module by Ezeiza et al. (1998), which uses a combination of knowledge-based disambiguation, by means of Constraint Grammar (Karlsson et al., 1995; Aduriz et al., 1997), and a posterior statistical disambiguation module, using an HMM.[16]

For the shared task data, we chose a setting that disambiguates most word forms, and retains $\geq 97\%$ of the correct interpretations, leaving an ambiguity level of 1.3 interpretations. For the remaining cases of ambiguity, we chose the first interpretation, which corresponds to the most frequent option. This leaves open the investigation of more complex approaches for selecting the most appropriate reading.[17]

4.4 The French Treebank

French is not a morphologically rich language per se, though its inflectional system is richer than that of English, and it also exhibits a limited amount of word order variation occurring at different syntactic levels including the word level (e.g. pre- or post-nominal

[15] We generated a 80/10/10 split, – train/dev/test – The first 5k sentences of the train set were used as a basis for the train5k.

[16] Note that the statistical module can be parametrized according to the level of disambiguation to trade off precision and recall. For example, disambiguation based on the main categories (abstracting over morpho-syntactic features) maintains most of the correct interpretations but still gives an output with several interpretations per wordform.

[17] This is not an easy task. The ambiguity left is the hardest to solve given that the knowledge-based and statistical disambiguation processes have not been able to pick out a single reading.

adjective, pre- or post-verbal adverbs) and the phrase level (e.g. possible alternations between post verbal NPs and PPs). It also has a high degree of multi-word expressions, that are often ambiguous with a literal reading as a sequence of simple words. The syntactic and MWE analysis shows the same kind of interaction (though to a lesser extent) as morphological and syntactic interaction in Semitic languages — MWEs help parsing, and syntactic information may be required to disambiguate MWE identification.

The Data Set The French data sets were generated from the French Treebank (Abeillé et al., 2003), which consists of sentences from the newspaper Le Monde, manually annotated with phrase structures and morphological information. Part of the treebank trees are also annotated with grammatical function tags for dependents of verbs. In the SPMRL shared task release, we used only this part, consisting of 18,535 sentences,[18] split into 14,759 sentences for training, 1,235 sentences for development, and 2,541 sentences for the final evaluation.[19]

Adapting the Data to the Shared Task The constituency trees are provided in an extended PTB bracketed format, with morphological features at the pre-terminal level only. They contain slight, automatically performed, modifications with respect to the original trees of the French treebank. The syntagmatic projection of prepositions and complementizers was normalized, in order to have prepositions and complementizers as heads in the dependency trees (Candito et al., 2010).

The dependency representations are projective dependency trees, obtained through automatic conversion from the constituency trees. The conversion procedure is an enhanced version of the one described by Candito et al. (2010).

Both the constituency and the dependency representations make use of coarse- and fine-grained POS tags (CPOS and FPOS respectively). The CPOS are the categories from the original treebank. The FPOS are merged using the CPOS and specific morphological information such as verbal mood, proper/common noun distinction (Crabbé and Candito, 2008).

Multi-Word Expressions The main difference with respect to previous releases of the bracketed or dependency versions of the French treebank lies in the representation of multi-word expressions (MWEs). The MWEs appear in an extended format: each MWE bears an FPOS[20] and consists of a sequence of terminals (hereafter the "components" of the MWE), each having their proper CPOS, FPOS, lemma and morphological features. Note though that in the original treebank the only gold information provided for a MWE component is its CPOS. Since leaving this information blank for MWE components would have provided a strong cue for MWE recognition, we made sure to provide the same kind of information for every terminal, whether MWE component or not, by providing predicted morphological features, lemma, and FPOS for MWE components (even in the "gold" section of the data set). This information was predicted by the Morfette tool (Chrupała et al., 2008), adapted to French (Seddah et al., 2010).

In the constituency trees, each MWE corresponds to an internal node whose label is the MWE's FPOS suffixed by a +, and which dominates the component pre-terminal nodes.

In the dependency trees, there is no "node" for a MWE as a whole, but one node (a terminal in the CoNLL format) per MWE component. The first component of a MWE is taken as the head of the MWE. All subsequent components of the MWE depend on the first one, with the special label *dep_cpd*. Furthermore, the first MWE component bears a feature *mwehead* equal to the FPOS of the MWE. For instance, the MWE *la veille* (*the day before*) is an adverb, containing a determiner component and a common noun component. Its bracketed representation is (ADV+ (DET la) (NC veille)), and in the dependency representation, the noun *veille* depends on the determiner *la*, which bears the feature `mwehead=ADV+`.

Predicted Morphology For the predicted morphology scenario, we provide data in which the `mwehead` has been removed and with predicted

[18] The process of functional annotation is still ongoing, the objective of the FTB providers being to have all the 20000 sentences annotated with functional tags.

[19] The first 9,981 training sentences correspond to the canonical 2007 training set. The development set is the same and the last 1235 sentences of the test set are those of the canonical test set.

[20] In the current data, we did not carry along the lemma and morphological features pertaining to the MWE itself, though this information is present in the original trees.

FPOS, CPOS, lemma, and morphological features, obtained by training Morfette on the whole train set.

4.5 The German Treebank

German is a fusional language with moderately free word order, in which verbal elements are fixed in place and non-verbal elements can be ordered freely as long as they fulfill the ordering requirements of the clause (Höhle, 1986).

The Data Set The German constituency data set is based on the TiGer treebank release 2.2.[21] The original annotation scheme represents discontinuous constituents such that all arguments of a predicate are always grouped under a single node regardless of whether there is intervening material between them or not (Brants et al., 2002). Furthermore, punctuation and several other elements, such as parentheses, are not attached to the tree. In order to make the constituency treebank usable for PCFG parsing, we adapted this treebank as described shortly.

The conversion of TiGer into dependencies is a variant of the one by Seeker and Kuhn (2012), which does not contain empty nodes. It is based on the same TiGer release as the one used for the constituency data. Punctuation was attached as high as possible, without creating any new non-projective edges.

Adapting the Data to the Shared Task For the constituency version, punctuation and other unattached elements were first attached to the tree. As attachment target, we used roughly the respective least common ancestor node of the right and left terminal neighbor of the unattached element (see Maier et al. (2012) for details), and subsequently, the crossing branches were resolved.

This was done in three steps. In the first step, the head daughters of all nodes were marked using a simple heuristic. In case there was a daughter with the edge label `HD`, this daughter was marked, i.e., existing head markings were honored. Otherwise, if existing, the rightmost daughter with edge label `NK` (noun kernel) was marked. Otherwise, as default, the leftmost daughter was marked. In a second step, for each continuous part of a discontinuous constituent, a separate node was introduced. This corresponds to the "raising" algorithm described by Boyd (2007). In a third steps, all those newly introduced nodes that did not cover the head daughter of the original discontinuous node were deleted. For the second and the third step, we used the same script as for the Swedish constituency data.

Predicted Morphology For the predicted scenario, a single sequence of POS tags and morphological features has been assigned using the MATE toolchain via a model trained on the train set via cross-validation on the training set. The MATE toolchain was used to provide predicted annotation for lemmas, POS tags, morphology, and syntax. In order to achieve the best results for each annotation level, a 10-fold jackknifing was performed to provide realistic features for the higher annotation levels. The predicted annotation of the 5k training set were copied from the full data set.[22]

4.6 The Hebrew Treebank

Modern Hebrew is a Semitic language, characterized by inflectional and derivational (templatic) morphology and relatively free word order. The function words for *from/to/like/and/when/that/the* are prefixed to the next token, causing severe segmentation ambiguity for many tokens. In addition, Hebrew orthography does not indicate vowels in modern texts, leading to a very high level of word-form ambiguity.

The Data Set Both the constituency and the dependency data sets are derived from the Hebrew Treebank V2 (Sima'an et al., 2001; Guthmann et al., 2009). The treebank is based on just over 6000 sentences from the daily newspaper 'Ha'aretz', manually annotated with morphological information and phrase-structure trees and extended with head information as described in Tsarfaty (2010, ch. 5). The unlabeled dependency version was produced by conversion from the constituency treebank as described in Goldberg (2011). Both the constituency and dependency trees were annotated with a set grammatical function labels conforming to Unified Stanford Dependencies by Tsarfaty (2013).

[21] This version is available from `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html`

[22] We also provided a predicted-all scenario, in which we provided morphological analysis lattices with POS and morphological information derived from the analyses of the SMOR derivational morphology (Schmid et al., 2004). These lattices were not used by any of the participants.

Adapting the Data to the Shared Task While based on the same trees, the dependency and constituency treebanks differ in their POS tag sets, as well as in some of the morphological segmentation decisions. The main effort towards the shared task was unifying the two resources such that the two treebanks share the same lexical yields, and the same pre-terminal labels. To this end, we took the layering approach of Goldberg et al. (2009), and included two levels of POS tags in the constituency trees. The lower level is lexical, conforming to the lexical resource used to build the lattices, and is shared by the two treebanks. The higher level is syntactic, and follows the tag set and annotation decisions of the original constituency treebank.[23] In addition, we unified the representation of morphological features, and fixed inconsistencies and mistakes in the treebanks.

Data Split The Hebrew treebank is one of the smallest in our language set, and hence it is provided in only the small (5k) setting. For the sake of comparability with the 5k set of the other treebanks, we created a comparable size of dev/test sets containing the first and last 500 sentences respectively, where the rest serve as the 5k training.[24]

Predicted Morphology The lattices encoding the morphological ambiguity for the Raw (all) scenario were produced by looking up the possible analyses of each input token in the wide-coverage morphological analyzer (lexicon) of the Knowledge Center for Processing Hebrew (Itai and Wintner, 2008; MILA, 2008), with a simple heuristic for dealing with unknown tokens. A small lattice encoding the possible analyses of each token was produced separately, and these token-lattices were concatenated to produce the sentence lattice. The lattice for a given sentence may not include the gold analysis in cases of incomplete lexicon coverage.

The morphologically disambiguated input files for the Raw (1-best) scenario were produced by running the raw text through the morphological disambiguator (tagger) described in Adler and Elhadad (2006; Goldberg et al. (2008),Adler (2007). The disambiguator is based on the same lexicon that is used to produce the lattice files, but utilizes an extra module for dealing with unknown tokens Adler et al. (2008). The core of the disambiguator is an HMM tagger trained on about 70M unannotated tokens using EM, and being supervised by the lexicon.

As in the case of Arabic, we also provided data for the Predicted (gold token / predicted morphology) scenario. We used the same sequence labeler, Morfette (Chrupała et al., 2008), trained on the concatenation of POS and morphological gold features, leading to a model with respectable accuracy.[25]

4.7 The Hungarian Treebank

Hungarian is an agglutinative language, thus a lemma can have hundreds of word forms due to derivational or inflectional affixation (nominal declination and verbal conjugation). Grammatical information is typically indicated by suffixes: case suffixes mark the syntactic relationship between the head and its arguments (subject, object, dative, etc.) whereas verbs are inflected for tense, mood, person, number, and the definiteness of the object. Hungarian is also characterized by vowel harmony.[26] In addition, there are several other linguistic phenomena such as causation and modality that are syntactically expressed in English but encoded morphologically in Hungarian.

The Data Set The Hungarian data set used in the shared task is based on the Szeged Treebank, the largest morpho-syntactic and syntactic corpus manually annotated for Hungarian. This treebank is based on newspaper texts and is available in both constituent-based (Csendes et al., 2005) and dependency-based (Vincze et al., 2010) versions.

Around 10k sentences of news domain texts were made available to the shared task.[27] Each word is manually assigned all its possible morpho-syntactic

[23] Note that this additional layer in the constituency treebank adds a relatively easy set of nodes to the trees, thus "inflating" the evaluation scores compared to previously reported results. To compensate, a stricter protocol than is used in this task would strip one of the two POS layers prior to evaluation.

[24] This split is slightly different than the split in previous studies.

[25] POS+morphology prediction accuracy is 91.95% overall (59.54% for unseen tokens). POS only prediction accuracy is 93.20% overall (71.38% for unseen tokens).

[26] When vowel harmony applies, most suffixes exist in two versions – one with a front vowel and another one with a back vowel – and it is the vowels within the stem that determine which form of the suffix is selected.

[27] The original treebank contains 82,000 sentences, 1.2 million words and 250,000 punctuation marks from six domains.

tags and lemmas and the appropriate one is selected according to the context. Sentences were manually assigned a constituency-based syntactic structure, which includes information on phrase structure, grammatical functions (such as subject, object, etc.), and subcategorization information (i.e., a given NP is subcategorized by a verb or an infinitive). The constituency trees were later automatically converted into dependency structures, and all sentences were then manually corrected. Note that there exist some differences in the grammatical functions applied to the constituency and dependency versions of the treebank, since some morpho-syntactic information was coded both as a morphological feature and as decoration on top of the grammatical function in the constituency trees.

Adapting the Data to the Shared Task Originally, the Szeged Dependency Treebank contained virtual nodes for elided material *(ELL)* and phonologically covert copulas *(VAN)*. In the current version, they have been deleted, their daughters have been attached to the parent of the virtual node, and have been given complex labels, e.g. *COORD-VAN-SUBJ*, where *VAN* is the type of the virtual node deleted, *COORD* is the label of the virtual node and *SUBJ* is the label of the daughter itself. When the virtual node was originally the root of the sentence, its daughter with a predicative *(PRED)* label has been selected as the new root of the sentence (with the label *ROOT-VAN-PRED*) and all the other daughters of the deleted virtual node have been attached to it.

Predicted Morphology In order to provide the same POS tag set for the constituent and dependency treebanks, we used the dependency POS tagset for both treebank instances. Both versions of the treebank are available with gold standard and automatic morphological annotation. The automatic POS tagging was carried out by a 10-fold cross-validation on the shared task data set by `magyarlanc`, a natural language toolkit for processing Hungarian texts (segmentation, morphological analysis, POS tagging, and dependency parsing). The annotation provides POS tags and deep morphological features for each input token (Zsibrita et al., 2013).[28]

[28]The full data sets of both the constituency and dependency versions of the Szeged Treebank are available at the following website: `www.inf.u-szeged.hu/rgai/SzegedTreebank`, and `magyarlanc` is downloadable from: `www.inf.u-szeged.hu/rgai/magyarlanc`.

4.8 The Korean Treebank

The Treebank The Korean corpus is generated by collecting constituent trees from the KAIST Treebank (Choi et al., 1994), then converting the constituent trees to dependency trees using head-finding rules and heuristics. The KAIST Treebank consists of about 31K manually annotated constituent trees from 97 different sources (e.g., newspapers, novels, textbooks). After filtering out trees containing annotation errors, a total of 27,363 trees with 350,090 tokens are collected.

The constituent trees in the KAIST Treebank[29] also come with manually inspected morphological analysis based on 'eojeol'. An eojeol contains root-forms of word tokens agglutinated with grammatical affixes (e.g., case particles, ending markers). An eojeol can consist of more than one word token; for instance, a compound noun "*bus stop*" is often represented as one eojeol in Korean, 버스정류장, which can be broken into two word tokens, 버스 (bus) and 정류장 (stop). Each eojeol in the KAIST Treebank is separated by white spaces regardless of punctuation. Following the Penn Korean Treebank guidelines (Han et al., 2002), punctuation is separated as individual tokens, and parenthetical notations surrounded by round brackets are grouped into individual phrases with a function tag (`PRN` in our corpus).

All dependency trees are automatically converted from the constituent trees. Unlike English, which requires complicated head-finding rules to find the head of each phrase (Choi and Palmer, 2012), Korean is a head final language such that the rightmost constituent in each phrase becomes the head of that phrase. Moreover, the rightmost conjunct becomes the head of all other conjuncts and conjunctions in a coordination phrase, which aligns well with our head-final strategy.

The constituent trees in the KAIST Treebank do not consist of function tags indicating syntactic or semantic roles, which makes it difficult to generate dependency labels. However, it is possible to generate meaningful labels by using the rich morphology in Korean. For instance, case particles give good

[29]See Lee et al. (1997) for more details about the bracketing guidelines of the KAIST Treebank.

indications of what syntactic roles eojeols with such particles should take. Given this information, 21 dependency labels were generated according to the annotation scheme proposed by Choi (2013).

Adapting the Data to the Shared Task All details concerning the adaptation of the KAIST treebank to the shared task specifications are found in Choi (2013). Importantly, the rich KAIST treebank tag set of 1975 POS tag types has been converted to a list of CoNLL-like feature-attribute values refining coarse grained POS categories.

Predicted Morphology Two sets of automatic morphological analyses are provided for this task. One is generated by the HanNanum morphological analyzer.[30] The HanNanum morphological analyzer gives the same morphemes and POS tags as the KAIST Treebank. The other is generated by the Sejong morphological analyzer.[31] The Sejong morphological analyzer gives a different set of morphemes and POS tags as described in Choi and Palmer (2011).

4.9 The Polish Treebank

The Data Set *Składnica* is a constituency treebank of Polish (Woliński et al., 2011; Świdziński and Woliński, 2010). The trees were generated with a non-probabilistic DCG parser *Świgra* and then disambiguated and validated manually. The analyzed texts come from the one-million-token subcorpus of the National Corpus of Polish (NKJP, (Przepiórkowski et al., 2012)) manually annotated with morpho-syntactic tags.

The dependency version of *Składnica* is a result of an automatic conversion of manually disambiguated constituent trees into dependency structures (Wróblewska, 2012). The conversion was an entirely automatic process. Conversion rules were based on morpho-syntactic information, phrasal categories, and types of phrase-structure rules encoded within constituent trees. It was possible to extract dependencies because the constituent trees contain information about the head of the majority of constituents. For other constituents, heuristics were defined in order to select their heads.

The version of *Składnica* used in the shared task comprises parse trees for 8,227 sentences.[32]

Predicted Morphology For the shared task Predicted scenario, an automatic morphological annotation was generated by the *PANTERA* tagger (Acedański, 2010).

4.10 The Swedish Treebank

Swedish is moderately rich in inflections, including a case system. Word order obeys the verb second constraint in main clauses but is SVO in subordinate clauses. Main clause order is freer than in English but not as free as in some other Germanic languages, such as German. Also, subject agreement with respect to person and number has been dropped in modern Swedish.

The Data Set The Swedish data sets are taken from the Talbanken section of the Swedish Treebank (Nivre and Megyesi, 2007). Talbanken is a syntactically annotated corpus developed in the 1970s, originally annotated according to the MAMBA scheme (Teleman, 1974) with a syntactic layer consisting of flat phrase structure and grammatical functions. The syntactic annotation was later automatically converted to full phrase structure with grammatical functions and from that to dependency structure, as described by Nivre et al. (2006).

Both the phrase structure and the dependency version use the functional labels from the original MAMBA scheme, which provides a fine-grained classification of syntactic functions with 65 different labels, while the phrase structure annotation (which had to be inferred automatically) uses a coarse set of only 8 labels. For the release of the Swedish treebank, the POS level was re-annotated to conform to the current de facto standard for Swedish, which is the Stockholm-Umeå tagset (Ejerhed et al., 1992) with 25 base tags and 25 morpho-syntactic features, which together produce over 150 complex tags.

For the shared task, we used version 1.2 of the treebank, where a number of conversion errors in the dependency version have been corrected. The phrase structure version was enriched by propagating morpho-syntactic features from preterminals (POS

[30] `http://kldp.net/projects/hannanum`
[31] `http://www.sejong.or.kr`
[32] *Składnica* is available from `http://zil.ipipan.waw.pl/Sklicense`.

tags) to higher non-terminal nodes using a standard head percolation table, and a version without crossing branches was derived using the lifting strategy (Boyd, 2007).

Adapting the Data to the Shared Task Explicit attribute names were added to the feature field and the split was changed to match the shared task minimal training set size.

Predicted Morphology POS tags and morpho-syntactic features were produced using the Hun-PoS tagger (Halácsy et al., 2007) trained on the Stockholm-Umeå Corpus (Ejerhed and Källgren, 1997).

5 Overview of the Participating Systems

With 7 teams participating, more than 14 systems for French and 10 for Arabic and German, this shared task is on par with the latest large-scale parsing evaluation campaign SANCL 2012 (Petrov and McDonald, 2012). The present shared task was extremely demanding on our participants. From 30 individuals or teams who registered and obtained the data sets, we present results for the seven teams that accomplished successful executions on these data in the relevant scenarios in the given the time frame.

5.1 Dependency Track

Seven teams participated in the dependency track. Two participating systems are based on MaltParser: MALTOPTIMIZER (Ballesteros, 2013) and AI:KU (Cirik and Şensoy, 2013). MALTOPTIMIZER uses a variant of MaltOptimizer (Ballesteros and Nivre, 2012) to explore features relevant for the processing of morphological information. AI:KU uses a combination of MaltParser and the original MaltOptimizer. Their system development has focused on the integration of an unsupervised word clustering method using contextual and morphological properties of the words, to help combat sparseness.

Similarly to MaltParser ALPAGE:DYALOG (De La Clergerie, 2013) also uses a shift-reduce transition-based parser but its training and decoding algorithms are based on beam search. This parser is implemented on top of the tabular logic programming system DyALog. To the best of our knowledge, this is the first dependency parser capable of handling word lattice input.

Three participating teams use the MATE parser (Bohnet, 2010) in their systems: the BASQUETEAM (Goenaga et al., 2013), IGM:ALPAGE (Constant et al., 2013) and IMS:SZEGED:CIS (Björkelund et al., 2013). The BASQUETEAM uses the MATE parser in combination with MaltParser (Nivre et al., 2007b). The system combines the parser outputs via MaltBlender (Hall et al., 2007). IGM:ALPAGE also uses MATE and MaltParser, once in a pipeline architecture and once in a joint model. The models are combined via a re-parsing strategy based on (Sagae and Lavie, 2006). This system mainly focuses on MWEs in French and uses a CRF tagger in combination with several large-scale dictionaries to handle MWEs, which then serve as input for the two parsers.

The IMS:SZEGED:CIS team participated in both tracks, with an ensemble system. For the dependency track, the ensemble includes the MATE parser (Bohnet, 2010), a best-first variant of the easy-first parser by Goldberg and Elhadad (2010b), and turbo parser (Martins et al., 2010), in combination with a ranker that has the particularity of using features from the constituent parsed trees. CADIM (Marton et al., 2013b) uses their variant of the easy-first parser combined with a feature-rich ensemble of lexical and syntactic resources.

Four of the participating teams use external resources in addition to the parser. The IMS:SZEGED:CIS team uses external morphological analyzers. CADIM uses SAMA (Graff et al., 2009) for Arabic morphology. ALPAGE:DYALOG and IGM:ALPAGE use external lexicons for French. IGM:ALPAGE additionally uses Morfette (Chrupała et al., 2008) for morphological analysis and POS tagging. Finally, as already mentioned, AI:KU clusters words and POS tags in an unsupervised fashion exploiting additional, un-annotated data.

5.2 Constituency Track

A single team participated in the constituency parsing task, the IMS:SZEGED:CIS team (Björkelund et al., 2013). Their phrase-structure parsing system uses a combination of 8 PCFG-LA parsers, trained using a product-of-grammars procedure (Petrov, 2010). The 50-best parses of this combination are then reranked by a model based on the reranker by Charniak and

Johnson (2005).[33]

5.3 Baselines

We additionally provide the results of two baseline systems for the nine languages, one for constituency parsing and one for dependency parsing.

For the dependency track, our baseline system is MaltParser in its default configuration (the arc-eager algorithm and liblinear for training). Results marked as BASE:MALT in the next two sections report the results of this baseline system in different scenarios.

The constituency parsing baseline is based on the most recent version of the PCFG-LA model of Petrov et al. (2006), used with its default settings and five split/merge cycles, for all languages.[34] We use this parser in two configurations: a '1-best' configuration where all POS tags are provided to the parser (predicted or gold, depending on the scenario), and another configuration in which the parser performs its own POS tagging. These baselines are referred to as BASE:BKY+POS and BASE:BKY+RAW respectively in the following results sections. Note that even when BASE:BKY+POS is given gold POS tags, the Berkeley parser sometimes fails to reach a perfect POS accuracy. In cases when the parser cannot find a parse with the provided POS, it falls back on its own POS tagging for all tokens.

6 Results

The high number of submitted system variants and evaluation scenarios in the task resulted in a large number of evaluation scores. In the following evaluation, we focus on the best run for each participant, and we aim to provide key points on the different dimensions of analysis resulting from our evaluation protocol. We invite our interested readers to browse the comprehensive representation of our results on the official shared-task results webpages.[35]

[33]Note that a slight but necessary change in the configuration of one of our metrics, which occurred after the system submission deadline, resulted in the IMS:SZEGED:CIS team to submit suboptimal systems for 4 languages. Their final scores are actually slightly higher and can be found in (Björkelund et al., 2013).

[34]For Semitic languages, we used the lattice based PCFG-LA extension by Goldberg (2011).

[35]http://www.spmrl.org/spmrl2013-sharedtask-results.html.

6.1 Gold Scenarios

This section presents the parsing results in gold scenarios, where the systems are evaluated on gold segmented and tagged input. This means that the sequence of terminals, POS tags, and morphological features are provided based on the treebank annotations. This scenario was used in most previous shared tasks on data-driven parsing (Buchholz and Marsi, 2006; Nivre et al., 2007a; Kübler, 2008). Note that this scenario was not mandatory. We thank our participants for providing their results nonetheless.

We start by reviewing dependency-based parsing results, both on the trees and on multi-word expression, and continue with the different metrics for constituency-based parsing.

6.1.1 Dependency Parsing

Full Training Set The results for the gold parsing scenario of dependency parsing are shown in the top block of table 3.

Among the six systems, IMS:SZEGED:CIS reaches the highest LAS scores, not only on average, but for every single language. This shows that their approach of combining parsers with (re)ranking provides robust parsing results across languages with different morphological characteristics. The second best system is ALPAGE:DYALOG, the third best system is MALTOPTIMIZER. The fact that AI:KU is ranked below the Malt baseline is due to their submission of results for 6 out of the 9 languages. Similarly, CADIM only submitted results for Arabic and ranked in the third place for this language, after the two IMS:SZEGED:CIS runs. IGM:ALPAGE and BASQUETEAM did not submit results for this setting.

Comparing LAS results across languages is problematic due to the differences between languages, treebank size and annotation schemes (see section 3), so the following discussion is necessarily tentative. If we consider results across languages, we see that the lowest results (around 83% for the best performing system) are reached for Hebrew and Swedish, the languages with the smallest data sets. The next lowest result, around 86%, is reached for Basque. Other languages reach similar LAS scores, around 88-92%. German, with the largest training set, reaches the highest LAS, 91.83%.

Interstingly, all systems have high LAS scores on the Korean Treebank given a training set size

team	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	avg.
1) gold setting / full training set										
IMS:SZEGED:CIS	**89.83**	**86.68**	**90.29**	**91.83**	**83.87**	**88.06**	**89.59**	**89.58**	**83.97**	**88.19**
ALPAGE:DYALOG	85.87	80.39	87.69	88.25	80.70	79.60	88.23	86.00	79.80	84.06
MALTOPTIMIZER	87.03	82.07	85.71	86.96	80.03	83.14	89.39	80.49	77.67	83.61
BASE:MALT	82.28	69.19	79.86	79.98	76.61	72.34	88.43	77.70	75.73	78.01
AI:KU			86.39	86.98	79.42	83.67		85.16	78.87	55.61
CADIM	85.56									9.51
2) gold setting / 5k training set										
IMS:SZEGED:CIS	**87.35**	**85.69**	**88.73**	**87.70**	**83.87**	**87.21**	83.38	**89.16**	**83.97**	**86.34**
ALPAGE:DYALOG	83.25	79.11	85.66	83.88	80.70	78.42	81.91	85.67	79.80	82.04
MALTOPTIMIZER	85.30	81.40	84.93	83.59	80.03	82.37	**83.74**	79.79	77.67	82.09
BASE:MALT	80.36	67.13	78.16	76.64	76.61	71.27	81.93	76.64	75.73	76.05
AI:KU			84.98	83.47	79.42	82.84		84.37	78.87	54.88
CADIM	82.67									9.19
3) predicted setting / full training set										
IMS:SZEGED:CIS	**86.21**	**85.14**	85.24	**89.65**	**80.89**	**86.13**	**86.62**	**87.07**	**82.13**	**85.45**
ALPAGE:DYALOG	81.20	77.55	82.06	84.80	73.63	75.58	81.02	82.56	77.54	79.55
MALTOPTIMIZER	81.90	78.58	79.00	82.75	73.01	79.63	82.65	79.89	75.82	79.25
BASE:MALT	80.36	70.11	77.98	77.81	69.97	70.15	82.06	75.63	73.21	75.25
AI:KU			72.57	82.32	69.01	78.92		81.86	76.35	51.23
BASQUETEAM		84.25	84.51	88.66		84.97			80.88	47.03
IGM:ALPAGE			**85.86**							9.54
CADIM	83.20									9.24
4) predicted setting / 5k training set										
IMS:SZEGED:CIS	**83.66**	**83.84**	83.45	**85.08**	**80.89**	**85.24**	**80.80**	**86.69**	**82.13**	**83.53**
MALTOPTIMIZER	79.64	77.59	77.56	79.22	73.01	79.00	75.90	79.50	75.82	77.47
ALPAGE:DYALOG	78.65	76.06	80.11	73.07	73.63	74.48	73.79	82.04	77.54	76.60
BASE:MALT	78.48	68.12	76.54	74.81	69.97	69.08	74.87	75.29	73.21	73.37
AI:KU			71.23	79.16	69.01	78.04		81.30	76.35	50.57
BASQUETEAM		83.19	82.65	84.70		84.01			80.88	46.16
IGM:ALPAGE			**83.60**							9.29
CADIM	80.51									8.95

Table 3: Dependency parsing: LAS scores for full and 5k training sets and for gold and predicted input. Results in bold show the best results per language and setting.

of approximately 23,000 sentences, which is a little over half of the German treebank. For German, on the other hand, only the IMS:SZEGED:CIS system reaches higher LAS scores than for Korean. This final observation indicates that more than treebank size is important for comparing system performance across treebanks. This is the reason for introducing the reduced set scenario, in which we can see how the participating system perform on a common ground, albeit small.

5k Training Set The results for the gold setting on the 5k train set are shown in the second block of Table 3. Compared with the full training, we see that there is a drop of around 2 points in this setting. Some parser/language pairs are more sensitive to data sparseness than others. CADIM, for instance, exhibit a larger drop than MALTOPTIMIZER on Arabic, and MALTOPTIMIZER shows a smaller drop than IMS:SZEGED:CIS on French. On average, among all systems that covered all languages, MALTOPTIMIZER has the smallest drop when moving to 5k training, possibly since the automatic feature optimization may differ for different data set sizes.

Since all languages have the same number of sentences in the train set, these results can give us limited insight into the parsing complexity of the different treebanks. Here, French, Arabic, Polish, and Korean reach the highest LAS scores while Swedish reaches

Team	F_MWE	F_COMP	F_MWE+POS
1) gold setting / full training set			
AI:KU	99.39	99.53	99.34
IMS:SZEGED:CIS	99.26	99.39	99.21
MALTOPTIMIZER	98.95	98.99	0
ALPAGE:DYALOG	98.32	98.81	0
BASE:MALT	68.7	72.55	68.7
2) predicted setting / full training set			
IGM:ALPAGE	80.81	81.18	77.37
IMS:SZEGED:CIS	79.45	80.79	70.48
ALPAGE:DYALOG	77.91	79.25	0
BASQUE-TEAM	77.19	79.81	0
MALTOPTIMIZER	70.29	74.25	0
BASE:MALT	67.49	71.01	0
AI:KU	0	0	0
3) predicted setting / 5k training set			
IGM:ALPAGE	77.66	78.68	74.04
IMS:SZEGED:CIS	77.28	78.92	70.42
ALPAGE:DYALOG	75.17	76.82	0
BASQUETEAM	73.07	76.58	0
MALTOPTIMIZER	65.76	70.42	0
BASE:MALT	62.05	66.8	0
AI:KU	0	0	0

Table 4: Dependency Parsing: MWE results

the lowest one. Treebank variance depends not only on the language but also on annotation decisions, such as label set (Swedish, interestingly, has a relatively rich one). A more careful comparison would then take into account the correlation of data size, label set size and parsing accuracy. We investigate these correlations further in section 7.1.

6.1.2 Multiword Expressions

MWE results on the gold setting are found at the top of Table 4. All systems, with the exception of BASE:MALT, perform exceedingly well in identifying the spans and non-head components of MWEs given gold morphology.[36] These almost perfect scores are the consequence of the presence of two gold MWE features, namely MWEHEAD and PRED=Y, which respectively indicate the node span of the whole MWE and its dependents, which do not have a gold feature field. The interesting scenario is, of course, the predicted one, where these features are not provided to the parser, as in any realistic application.

[36]Note that for the labeled measure F_MWE+POS, both MALTOPTIMIZER and ALPAGE:DYALOG have an F-score of zero, since they do not attempt to predict the MWE label at all.

6.1.3 Constituency Parsing

In this part, we provide accuracy results for phrase-structure trees in terms of ParsEval F-scores. Since ParsEval is sensitive to the non-terminals-per-word ratio in the data set (Rehbein and van Genabith, 2007a; Rehbein and van Genabith, 2007b), and given the fact that this ratio varies greatly within our data set (as shown in Table 2), it must be kept in mind that ParsEval should only be used for comparing parsing performance over treebank instances sharing the exact same properties in term of annotation schemes, sentence length and so on. When comparing F-Scores across different treebanks and languages, it can only provide a rough estimate of the relative difficulty or ease of parsing these kinds of data.

Full Training Set The F-score results for the gold scenario are provided in the first block of Table 5. Among the two baselines, BASE:BKY+POS fares better than BASE:BKY+RAW since the latter selects its own POS tags and thus cannot benefit from the gold information. The IMS:SZEGED:CIS system clearly outperforms both baselines, with Hebrew as an outlier.[37]

As in the dependency case, the results are not strictly comparable across languages, yet we can draw some insights from them. We see considerable differences between the languages, with Basque, Hebrew, and Hungarian reaching F-scores in the low 90s for the IMS:SZEGED:CIS system, Korean and Polish reaching above-average F-scores, and Arabic, French, German, and Swedish reaching F-scores below the average, but still in the low 80s. The performance is, again, not correlated with data set sizes. Parsing Hebrew, with one of the smallest training sets, obtains higher accuracy many other languages, including Swedish, which has the same training set size as Hebrew. It may well be that gold morphological information is more useful for combatting sparseness in languages with richer morphology (though Arabic here would be an outlier for this conjecture), or it may be that certain treebanks and schemes are inherently harder to parser than others, as we investigate in section 7.

For German, the language with the largest training

[37]It might be that the easy layer of syntactic tags benefits from the gold POS tags provided. See section 4 for further discussion of this layer.

team	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	avg.
1) gold setting / full training set										
IMS:SZEGED:CIS	**82.20**	**90.04**	**83.98**	**82.07**	91.64	**92.60**	**86.50**	**88.57**	**85.09**	**86.97**
BASE:BKY+POS	80.76	76.24	81.76	80.34	**92.20**	87.64	82.95	88.13	82.89	83.66
BASE:BKY+RAW	79.14	69.78	80.38	78.99	87.32	81.44	73.28	79.51	78.94	78.75
2) gold setting / 5k training set										
IMS:SZEGED:CIS	**79.47**	**88.45**	**82.25**	**74.78**	91.64	**91.87**	**80.10**	**88.18**	**85.09**	**84.65**
BASE:BKY+POS	77.54	74.06	78.07	71.37	**92.20**	86.74	72.85	87.91	82.89	80.40
BASE:BKY+RAW	75.22	67.16	75.91	68.94	87.32	79.34	60.40	78.30	78.94	74.61
3) predicted setting / full training set										
IMS:SZEGED:CIS	**81.32**	**87.86**	**81.83**	**81.27**	**89.46**	**91.85**	**84.27**	**87.55**	**83.99**	**85.49**
BASE:BKY+POS	78.66	74.74	79.76	78.28	85.42	85.22	78.56	86.75	80.64	80.89
BASE:BKY+RAW	79.19	70.50	80.38	78.30	86.96	81.62	71.42	79.23	79.18	78.53
4) predicted setting / 5k training set										
IMS:SZEGED:CIS	**78.85**	**86.65**	**79.83**	**73.61**	**89.46**	**90.53**	**78.47**	**87.46**	**83.99**	**83.21**
BASE:BKY+POS	74.84	72.35	76.19	69.40	85.42	83.82	67.97	87.17	80.64	77.53
BASE:BKY+RAW	74.57	66.75	75.76	68.68	86.96	79.35	58.49	78.38	79.18	74.24

Table 5: Constituent Parsing: ParsEval F-scores for full and 5k training sets and for gold and predicted input. Results in bold show the best results per language and setting.

set and the highest scores in dependency parsing, the F-scores are at the lower end. These low scores, which are obtained despite the larger treebank and only moderately free word-order, are surprising. This may be due to case syncretism; gold morphological information exhibits its own ambiguity and thus may not be fully utilized.

5k Training Set Parsing results on smaller comparable test sets are presented in the second block of Table 5. On average, IMS:SZEGED:CIS is less sensitive than BASE:BKY+POS to the reduced size. Systems are not equally sensitive to reduced training sets, and the gaps range from 0.4% to 3%, with German and Korean as outliers (Korean suffering a 6.4% drop in F-score and German 7.3%). These languages have the largest treebanks in the full setting, so it is not surprising that they suffer the most. But this in itself does not fully explain the cross-treebank trends. Since ParsEval scores are known to be sensitive to the label set sizes and the depth of trees, we provide LeafAncestor scores in the following section.

6.1.4 Leaf-Ancestor Results

The variation across results in the previous subsection may have been due to differences across annotation schemes. One way to neutralize this difference (to some extent) is to use a different metric. We evaluated the constituency parsing results using the Leaf-Ancestor (LA) metric, which is less sensitive to the number of nodes in a tree (Rehbein and van Genabith, 2007b; Kübler et al., 2008). As shown in Table 6, these results are on a different (higher) scale than ParsEval, and the average gap between the full and 5k setting is lower.

Full Training Set The LA results in gold setting for full training sets are shown in the first block of Table 6. The trends are similar to the ParsEval F-scores. German and Arabic present the lowest LA scores (in contrast to the corresponding F-scores, Arabic is a full point below German for IMS:SZEGED:CIS). Basque and Hungarian have the highest LA scores. Hebrew, which had a higher F-score than Basque, has a lower LA than Basque and is closer to French. Korean also ranks worse in the LA analysis. The choice of evaluation metrics thus clearly impacts system rankings – F-scores rank some languages suspiciously high (e.g., Hebrew) due to deeper trees, and another metric may alleviate that.

5k Training Set The results for the leaf-ancestor (LA) scores in the gold setting for the 5k training set are shown in the second block of Table 6. Across

team	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	avg.
1) gold setting / full training set										
IMS:SZEGED:CIS	**88.61**	**94.90**	**92.51**	**89.63**	**92.84**	**95.01**	**91.30**	**94.52**	**91.46**	**92.31**
BASE:BKY+POS	87.85	91.55	91.74	88.47	92.69	92.52	90.82	92.81	90.76	91.02
BASE:BKY+RAW	87.05	89.71	91.22	87.77	91.29	90.62	87.11	90.58	88.97	89.37
2) gold setting / 5k training set										
IMS:SZEGED:CIS	**86.68**	**94.21**	**91.56**	**85.74**	**92.84**	**94.79**	**88.87**	**94.17**	**91.46**	**91.15**
BASE:BKY+POS	86.26	90.72	89.71	84.11	92.69	92.11	86.75	92.91	90.76	89.56
BASE:BKY+RAW	84.97	88.68	88.74	83.08	91.29	89.94	81.82	90.31	88.97	87.53
3) predicted setting / full training set										
IMS:SZEGED:CIS	**88.45**	**94.50**	**91.79**	**89.32**	**91.95**	**94.90**	**90.13**	**94.11**	**91.05**	**91.80**
BASE:BKY+POS	86.60	90.90	90.96	87.46	89.66	91.72	89.10	92.56	89.51	89.83
BASE:BKY+RAW	86.97	89.91	91.11	87.46	90.77	90.50	86.68	90.48	89.16	89.23
4) predicted setting / 5k training set										
IMS:SZEGED:CIS	**86.69**	**93.85**	**90.76**	**85.20**	**91.95**	**94.05**	**87.99**	**93.99**	**91.05**	**90.61**
BASE:BKY+POS	84.76	89.83	89.18	83.05	89.66	91.24	84.87	92.74	89.51	88.32
BASE:BKY+RAW	84.63	88.50	89.00	82.69	90.77	89.93	81.50	90.08	89.16	87.36

Table 6: Constituent Parsing: Leaf-Ancestor scores for full and 5k training sets and for gold and predicted input.

parsers, IMS:SZEGED:CIS again has a smaller drop than BASE:BKY+POS on the reduced size. German suffers the most from the reduction of the training set, with a loss of approximately 4 points. Korean, however, which was also severely affected in terms of F-scores, only loses 1.17 points in the LA score. On average, the LA seem to reflect a smaller drop when reducing the training set — this underscores again the impact of the choice of metrics on system evaluation.

6.2 Predicted Scenarios

Gold scenarios are relatively easy since syntactically relevant morphological information is disambiguated in advance and is provided as input. Predicted scenarios are more difficult: POS tags and morphological features have to be automatically predicted, by the parser or by external resources.

6.2.1 Dependency Parsing

Eight participating teams submitted dependency results for this scenario. Two teams submitted for a single language. Four teams covered all languages.

Full Training Set The results for the predicted scenario in full settings are shown in the third block of Table 3. Across the board, the results are considerably lower than the gold scenario. Again, IMS:SZEGED:CIS is the best performing system, followed by ALPAGE:DYALOG and MALTOPTIMIZER. The only language for which IMS:SZEGED:CIS is outperformed is French, for which IGM:ALPAGE reaches higher results (85.86% vs. 85.24%). This is due to the specialized treatment of French MWEs in the IGM:ALPAGE system, which is thereby shown to be beneficial for parsing in the predicted setting.

If we compare the results for the predicted setting and the gold one, given the full training set, the IMS:SZEGED:CIS system shows small differences between 1.5 and 2 percent. The only exception is French, for which the LAS drops from 90.29% to 85.24% in the predicted setting. The other systems show somewhat larger differences than IMS:SZEGED:CIS, with the highest drops for Arabic and Korean. The AI:KU system shows a similar problem as IMS:SZEGED:CIS for French.

5k Training Set When we consider the predicted setting for the 5k training set, in the last block of Table 3, we see the same trends as comparing with the full training set or when comparing to the gold setting. Systems suffer from not having gold standard data, and they suffer from the small training set. Interestingly, the loss between the different training set sizes in the predicted setting is larger than in the

gold setting, but only marginally so, with a difference < 0.5. In other words, the predicted setting adds a challenge to parsing, but it only minimally compounds data sparsity.

6.2.2 Multiword Expressions Evaluation

In the predicted setting, shown in the second block of table 4 for the full training set and in the third block of the same table for the 5k training set, we see that only two systems, IGM:ALPAGE and IMS:SZEGED:CIS can predict the MWE label when it is not present in the training set. IGM:ALPAGE's approach of using a separate classifier in combination with external dictionaries is very successful, reaching an F_MWE+POS score of 77.37. This is compared to the score of 70.48 by IMS:SZEGED:CIS, which predicts this node label as a side effect of their constituent feature enriched dependency model (Björkelund et al., 2013). AI:KU has a zero score for all predicted settings, which results from an erroneous training on the gold data rather than the predicted data.[38]

6.2.3 Constituency Parsing

Full Training Set The results for the predicted setting with the full training set are shown in the third block of table 5. A comparison with the gold setting shows that all systems have a lower performance in the predicted scenario, and the differences are in the range of 0.88 for Arabic and 2.54 for Basque. It is interesting to see that the losses are generally smaller than in the dependency framework: on average, the loss across languages is 2.74 for dependencies and 1.48 for constituents. A possible explanation can be found in the two-dimensional structure of the constituent trees, where only a subset of all nodes is affected by the quality of morphology and POS tags. The exception to this trend is Basque, for which the loss in constituents is a full point higher than for dependencies. Another possible explanation is that all of our constituent parsers select their own POS tags in one way or another. Most dependency parsers accept predicted tags from an external resource, which puts an upper-bound on their potential performance.

5k Training Set The results for the predicted setting given the 5k training set are shown in the bottom block of table 5. They show the same trends as the dependency ones: The results are slightly lower than the results obtained in gold setting and the ones utilizing the full training set.

6.2.4 Leaf Ancestor Metrics

Full Training Set The results for the predicted scenario with a full training set are shown in the third block of table 6. In the LA evaluation, the loss in moving from gold morphology are considerably smaller than in F-scores. For most languages, the loss is less than 0.5 points. Exceptions are French with a loss of 0.72, Hebrew with 0.89, and Korean with 1.17. Basque, which had the highest loss in F-scores, only shows a minor loss of 0.4 points. Also, the average loss of 0.41 points is much smaller than the one in the ParsEval score, 1.48.

5k Training Set The results for the predicted setting given the 5k training set are shown in the last block of table 6. These results, though considerably lower (around 3 points), exhibit the exact same trends as observed in the gold setting.

6.3 Realistic Raw Scenarios

The previous scenarios assume that input surface tokens are identical to tree terminals. For languages such as Arabic and Hebrew, this is not always the case. In this scenario, we evaluate the capacity of a system to predict both morphological segmentation and syntactic parse trees given raw, unsegmented input tokens. This may be done via a pipeline assuming a 1-st best morphological analysis, or jointly with parsing, assuming an ambiguous morphological analysis lattice as input. In this task, both of these scenarios are possible (see section 3). Thus, this section presents a realistic evaluation of the participating systems, using TedEval, which takes into account complete morpho-syntactic parses.

Tables 7 and 8 present labeled and unlabeled TedEval results for both constituency and dependency parsers, calculated only for sentence of length $<= 70$.[39] We firstly observe that labeled TedEval scores are considerably lower than unlabeled TedEval scores, as expected, since unlabeled scores evaluate only structural differences. In the labeled setup,

[38] Unofficial updated results are to to be found in (Cirik and Şensoy, 2013)

[39] TedEval builds on algorithms for calculating edit distance on complete trees (Bille, 2005). In these algorithms, longer sentences take considerably longer to evaluate.

	Arabic full training set		Arabic 5k training set		Hebrew 5k training set		All	
	Acc (x100)	Ex (%)	Acc (x100)	Ex (%)	Acc (x100)	Ex (%)	Avg.	Soft Avg.
IMS:SZEGED:CIS (Bky)	83.34	1.63	82.54	0.67	56.47	0.67	69.51	69.51
IMS:SZEGED:CIS	**89.12**	**8.37**	**87.82**	**5.56**	**86.08**	**8.27**	**86.95**	**86.95**
CADIM	87.81	6.63	86.43	4.21	-	-	43.22	86.43
MALTOPTIMIZER	86.74	5.39	85.63	3.03	83.05	5.33	84.34	84.34
ALPAGE:DYALOG	86.60	5.34	85.71	3.54	82.96	6.17	41.48	82.96
ALPAGE:DYALOG (RAW)	-	-	-	-	82.82	4.35	41.41	82.82
AI:KU	-	-	-	-	78.57	3.37	39.29	78.57

Table 7: Realistic Scenario: Tedeval Labeled Accuracy and Exact Match for the Raw scenario.
The upper part refers to constituency results, the lower part refers to dependency results

	Arabic full training set		Arabic 5k training set		Hebrew 5k training set		All	
	Acc (x100)	Ex (%)	Acc (x100)	Ex (%)	Acc (x100)	Ex (%)	Avg.	Soft Avg.
IMS:SZEGED:CIS (Bky)	**92.06**	9.49	**91.29**	7.13	89.30	13.60	90.30	90.30
IMS:SZEGED:CIS	91.74	**9.83**	90.85	**7.30**	**89.47**	**16.97**	90.16	90.16
ALPAGE:DYALOG	89.99	7.98	89.46	5.67	88.33	12.20	88.90	88.90
MALTOPTIMIZER	90.09	7.08	89.47	5.56	87.99	11.64	88.73	88.73
CADIM	90.75	8.48	89.89	5.67	-	-	44.95	89.89
ALPAGE:DYALOG (RAW)	-	-	-	-	87.61	10.24	43.81	87.61
AI:KU	-	-	-	-	86.70	8.98	43.35	86.70

Table 8: Realistic Scenario: Tedeval Unlabeled Accuracy and Exact Match for the Raw scenario.
Top upper part refers to constituency results, the lower part refers to dependency results.

the IMS:SZEGED:CIS dependency parser are the best for both languages and data set sizes. Table 8 shows that their unlabeled constituency results reach a higher accuracy than the next best system, their own dependency results. However, a quick look at the exact match metric reveals lower scores than for its dependency counterparts.

For the dependency-based joint scenarios, there is obviously an upper bound on parser performance given inaccurate segmentation. The transition-based systems, ALPAGE:DYALOG & MALTOPTIMIZER, perform comparably on Arabic and Hebrew, with ALPAGE:DYALOG being slightly better on both languages. Note that ALPAGE:DYALOG reaches close results on the 1-best and the lattice-based input settings, with a slight advantage for the former. This is partly due to the insufficient coverage of the lexical resource we use: many lattices do not contain the gold path, so the joint prediction can only as be high as the lattice predicted path allows.

7 Towards In-Depth Cross-Treebank Evaluation

Section 6 reported evaluation scores across systems for different scenarios. However, as noted, these results are not comparable across languages, representation types and parsing scenarios due to differences in the data size, label set size, length of sentences and also differences in evaluation metrics.

Our following discussion in the first part of this section highlights the kind of impact that data set properties have on the standard metrics (label set size on LAS, non-terminal nodes per sentence on F-score). Then, in the second part of this section we use the TedEval cross-experiment protocols for comparative evaluation that is less sensitive to representation types and annotation idiosyncrasies.

7.1 Parsing Across Languages and Treebanks

To quantify the impact of treebank characteristics on parsing parsing accuracy we looked at correlations of treebank properties with parsing results. The most highly correlated combinations we have found are shown in Figures 2, 3, and 4 for the dependency track and the constituency track (F-score and LeafAnces-

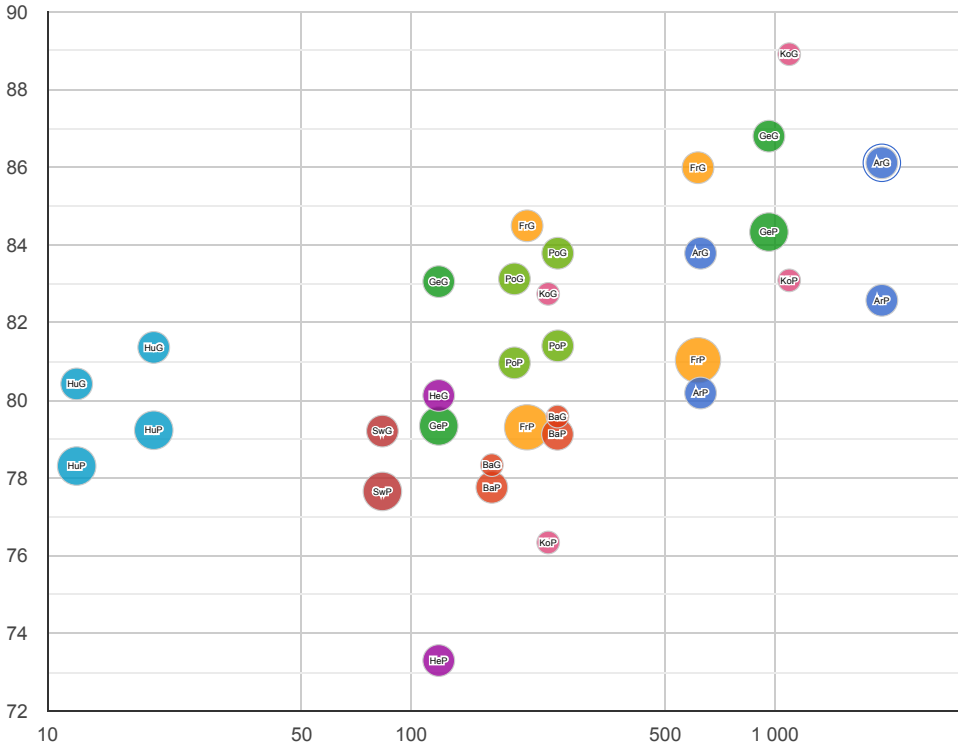


Figure 2: The correlation between treebank size, label set size, and LAS scores. *x: treebank size / #labels ; y: LAS (%)*

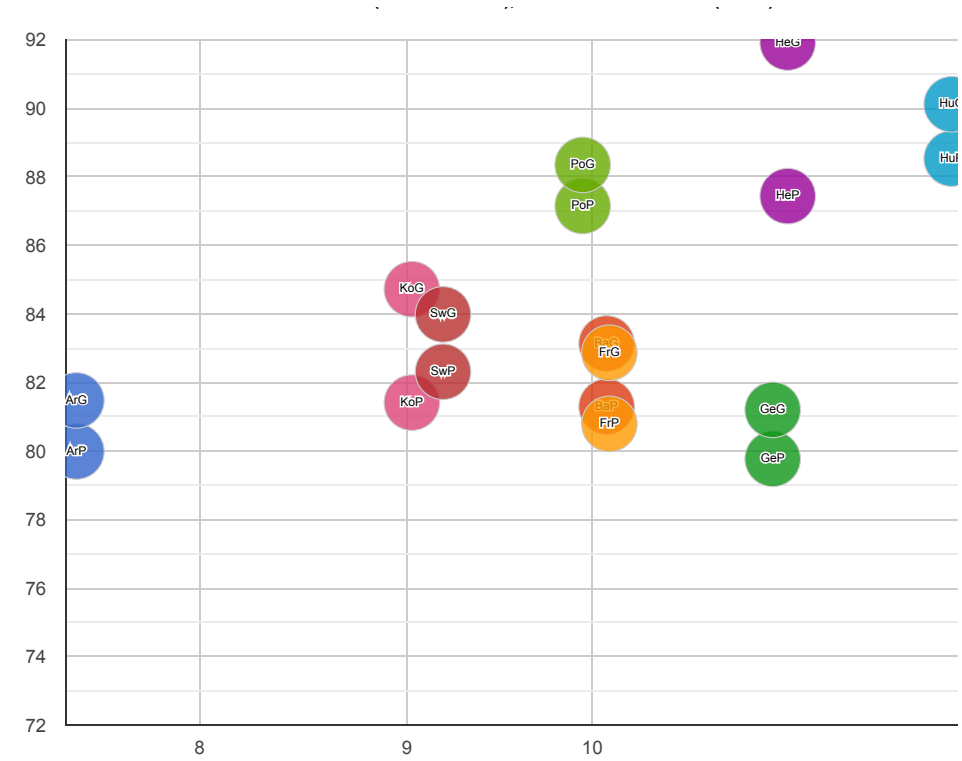


Figure 3: The correlation between the non terminals per sentence ratio and F-scores. *x: #non terminal/ #sentence ; y: F1 (%)*

tor) respectively.

Figure 2 presents the LAS against the average number of tokens relative to the number of labels. The numbers are averaged per language over all participating systems, and the size of the "bubbles" is proportional to the number of participants for a given language setting. We provide "bubbles" for all languages in the predicted (-P) and gold (-G) setting, for both training set sizes. The lower dot in terms of parsing scores always corresponds to the reduced training set size.

Figure 2 shows a clear correlation between dataset complexity and parsing accuracy. The simpler the data set is (where "simple" here translates into large data size with a small set of labels), the higher the results of the participating systems. The bubbles reflects a diagonal that indicates correlation between these dimensions. Beyond that, we see two interesting points off of the diagonal. The Korean treebank (pink) in the gold setting and full training set can be parsed with a high LAS relative to its size and label set. It is also clear that the Hebrew treebank (purple) in the predicted version is the most difficult one to parse, relative to our expectation about its complexity. Since the Hebrew gold scenario is a lot closer to the diagonal again, it may be that this outlier is due to the coverage and quality of the predicted morphology.

Figure 3[40] shows the correlation of data complexity in terms of the average number of non-terminals per sentence, and parsing accuracy (ParsEval F-score). Parsing accuracy is again averaged over all participating systems for a given language. In this figure, we see a diagonal similar to the one in figure 2, where Arabic (dark blue) has high complexity of the data (here interpreted as flat trees, low number of non terminals per sentence) and low F-scores accordingly. Korean (pink), Swedish (burgundy), Polish (light green), and Hungarian (light blue) follow, and then Hebrew (purple) is a positive outlier, possibly due to an additional layer of "easy" syntactic POS nodes which increases tree size and inflates F-scores. French (orange), Basque (red), and German (dark green) are negative outliers, falling off the diagonal. German has the lowest F-score with respect to what would be expected for the non-terminals per sentence ratio, which is in contrast to the LAS figure where German occurs among the less complex data set to parse. A possible explanation may be the crossing branches in the original treebank which were re-attached. This creates flat and variable edges which might be hard predict accurately.

Figure 4[41] presents the correlation between parsing accuracy in terms the LeafAncestor metrics (macro averaged) and treebank complexity in terms of the average number of non-terminals per sentence. As in the correlation figures, the parsing accuracy is averaged over the participanting systems for any language. The LeafAncestor accuracy is calculated over phrase structure trees, and we see a similar diagonal to the one in Figure 3 showing that flatter treebanks are harder (that is, are correlated with lower averaged scores) But, its slope is less steep than for the F-score, which confirms the observation that the LeafAncestor metric is less sensitive than F-score to the non-terminals-per-sentence ratio.

Similarly to Figure 3, German is a negative outlier, which means that this treebank is harder to parse – it obtains lower scores on average than we would expect. As for Hebrew, it is much closer to the diagonal. As it turns out, the "easy" POS layer that inflates the scores does not affect the LA ratings as much.

7.2 Evaluation Across Scenarios, Languages and Treebanks

In this section we analyze the results in cross-scenario, cross-annotation, and cross-framework settings using the evaluation protocols discussed in (Tsarfaty et al., 2012b; Tsarfaty et al., 2011; Tsarfaty et al., 2012a).

As a starting point, we select comparable sections of the parsed data, based on system runs trained on the small train set (train5k). For those, we selected subsets containing the first 5,000 tree terminals (respecting sentence boundaries) of the test set. We only used TedEval on sentences up to 70 terminals long, and projectivized non-projective sentences in all sets. We use the TedEval metrics to calculate scores on both constituency and dependency structures in all languages and all scenarios. Since the metric defines one scale for all of these different cases, we can

[40] This figure was created from the IMS:SZEGED:CIS (Const.) and our own PCFG-LA baseline in POS Tagged mode (BASE:BKY+POS) so as to avoid the noise introduced by the parser's own tagging step (BASE:BKY+RAW).

[41] This figure was created under the same condition as the F-score correlation in figure (Figure 3).

Figure 4: The correlation between the non terminals per sentence ratio and Leaf Accuracy (macro) scores. *x: #non terminal/ #sentence ; y: Acc.(%)*

compare the performance across annotation schemes, assuming that those subsets are representative of their original source.[42]

Ideally, we would be using labeled TedEval scores, as the labeled parsing task is more difficult, and labeled parses are far more informative than unlabeled ones. However, most constituency-based parsers do not provide function labels as part of the output, to be compared with the dependency arcs. Furthermore, as mentioned earlier, we observed a huge difference between label set sizes for the dependency runs. Consequently, labeled scores will not be as informative across treebanks and representation types. We will therefore only use labels across scenarios for the same language and representation type.

[42]We choose this sample scheme for replicability. We first tried sampling sentences, aiming at the same average sentence length (20), but that seemed to create artificially difficult test sets for languages as Polish and overly simplistic ones for French or Arabic.

7.2.1 Cross-Scenario Evaluation: raw vs. gold

One novel aspect of this shared task is the evaluation on non-gold segmentation in addition to gold morphology. One drawback is that the scenarios are currently not using the same metrics — the metrics generally applied for gold and predicted scenrios cannot apply for raw. To assess how well state of the art parsers perform in raw scenarios compared to gold scenarios, we present here TedEval results comparing raw and gold systems using the evaluation protocol of Tsarfaty et al. (2012b).

Table 9 presents the labeled and unlabeled results for Arabic and Hebrew (in Full and 5k training settings), and Table 10 presents unlabeled TedEval results (for all languages) in the gold settings. The unlabeled TedEval results for the raw settings are substantially lower then TedEval results on the gold settings for both languages.

When comparing the unlabeled TedEval results for Arabic and Hebrew on the participating systems, we see a loss of 3-4 points between Table 9 (raw) and Table 10 (gold). In particular we see that for the best per-

forming systems on Arabic (IMS:SZEGED:CIS for both constituency and dependency), the gap between gold and realistic scenarios is 3.4 and 4.3 points, for the constituency and the dependency parser respectively. These results are on a par with results by Tsarfaty et al. (2012b), who showed for different settings, constituency and dependency based, that raw scenarios are considerably more difficult to parse than gold ones on the standard split of the Modern Hebrew treebank.

For Hebrew, the performance gap between unlabeled TedEval in raw (Table 9) and gold (Table 10) is even more salient, with around 7 and 8 points of difference between the scenarios. We can only speculate that such a difference may be due to the difficulty of resolving Hebrew morpho-syntactic ambiguities without sufficient syntactic information. Since Hebrew and Arabic now have standardized morphologically and syntactically analyzed data sets available through this task, it will be possible to investigate further how cross-linguistic differences in morphological ambiguity affect full-parsing accuracy in raw scenarios.

This section compared the raw and gold parsing results only on unlabeled TedEval metrics. According to what we have seen so far is expected that for labeled TedEval metrics using the same protocol, the gap between gold and raw scenario will be even greater.

7.2.2 Cross-Framework Evaluation: Dependency vs. Constituency

In this section, our focus is on comparing parsing results across constituency and dependency parsers based on the protocol of Tsarfaty et al. (2012a) We have only one submission from IMS:SZEGED:CIS in the constituency track, and. from the same group, a submission on the dependency track. We only compare the IMS:SZEGED:CIS results on constituency and dependency parsing with the two baselines we provided. The results of the cross-framework evaluation protocol are shown in Table 11.

The results comparing the two variants of the IMS:SZEGED:CIS systems show that they are very close for all languages, with differences ranging from 0.03 for German to 0.8 for Polish in the gold setting.

It has often been argued that dependency parsers perform better than a constituency parser, but we notice that when using a cross framework protocol, such as TedEval, and assuming that our test set sample is representative, the difference between the interpretation of both representation's performance is alleviated. Of course, here the metric is unlabeled, so it simply tells us that both kind of parsing models are equally able to provide similar tree structures. Said differently, the gaps in the quality of predicting the same underlying structure across representations for MRLs is not as large as is sometimes assumed.

For most languages, the baseline constituency parser performs better than the dependency baseline one, with Basque and Korean as an exception, and at the same time, the dependency version of IMS:SZEGED:CIS performs slightly better than their constituent parser for most languages, with the exception of Hebrew and Hungarian. It goes to show that, as far as these present MRL results go, there is no clear preference for a dependency over a constituency parsing representation, just preferences among particular models.

More generally, we can say that even if the linguistic coverage of one theory is shown to be better than another one, it does not necessarily mean that the statistical version of the formal theory will perform better for structure prediction. System performance is more tightly related to the efficacy of the learning and search algorithms, and feature engineering on top of the selected formalism.

7.2.3 Cross-Language Evaluation: All Languages

We conclude with an overall outlook of the TedEval scores across all languages. The results on the gold scenario, for the small training set and the 5k test set are presented in Table 10. We concentrate on gold scenarios (to avoid the variation in coverage of external morphological analyzers) and choose unlabeled metrics as they are not sensitive to label set sizes. We emphasize in bold, for each parsing system (row in the table), the top two languages that most accurately parsed by it (boldface) and the two languages it performed the worse on (italics).

We see that the European languages German and Hungarian are parsed most accurately in the constituency-based setup, with Polish and Swedish having an advantage in dependency parsing. Across all systems, Korean is the hardest to parse, with Ara-

	Arabic	Hebrew	AVG1	SOFT AVG	Arabic	Hebrew	AVG2	SOFT AVG2
				1) Constituency Evaluation				
			Labeled TedEval				*Unabeled TedEval*	
IMS:SZEGED:CIS (Bky)	83.59	56.43	70.01	70.01	**92.18**	88.02	**90.1**	90.1
				2) Dependency Evaluation				
			Labeled TedEval				*Unabeled TedEval*	
IMS:SZEGED:CIS	**88.61**	**84.74**	**86.68**	86.68	91.41	**88.58**	90	90
ALPAGE:DYALOG	87.20	81.65	40.83	81.65	90.74	87.44	89.09	89.09
CADIM	87.99	-	44	**87.99**	91.22	-	45.61	**91.22**
MALTOPTIMIZER	86.62	81.74	43.31	86.62	90.26	87.00	45.13	90.26
ALPAGE:DYALOG (RAW)	-	82.82	41.41	82.82	-	87.43	43.72	87.43
AI:KU	-	77.8	38.9	77.8	-	85.87	42.94	85.87

Table 9: Labeled and Unlabeled TedEval Results for raw Scenarios, Trained on 5k sentences and tested on 5k terminals. *The upper part refers to constituency parsing and the lower part refers to dependency parsing*.

	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish
				1) Constituency Evaluation					
IMS:SZEGED:CIS (Bky)	*95.35*	96.91	95.98	**97.12**	96.22	**97.92**	*92.91*	97.19	96.65
BASE:BKY+POS	*95.11*	94.69	95.08	**97.01**	95.85	**97.08**	*90.55*	96.99	96.38
BASE:BKY+RAW	94.58	*94.32*	94.72	**96.74**	95.64	**96.15**	*87.08*	95.93	95.90
				2) Dependency Evaluation					
IMS:SZEGED:CIS	*95.76*	**97.63**	96.59	96.88	96.29	97.56	*94.62*	**98.01**	97.22
ALPAGE:DYALOG	*93.76*	95.72	95.75	96.4	95.34	95.63	*94.56*	**96.80**	**96.55**
BASE:MALT	*94.16*	95.08	*94.21*	94.55	94.98	95.25	94.27	**95.83**	**95.33**
AI:KU	-	-	95.46	96.34	*95.07*	96.53	-	**96.88**	**95.87**
MALTOPTIMIZER	*94.91*	**96.82**	95.23	**96.32**	95.46	96.30	*94.69*	96.06	95.90
CADIM	94.66	-	-	-	-	-	-	-	-

Table 10: Cross-Language Evaluation: Unlabeled TedEval Results in gold input scenario, On a 5k-sentences set set and a 5k-terminals test set. *The upper part refers to constituency parsing and the lower part refers to dependency parsing. For each system we mark the two top scoring languages in* **bold** *and the two lowest scoring languages in* italics.

team	Arabic	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish
				1) gold setting					
IMS:SZEGED:CIS (Bky)	95.82	97.30	96.15	97.43	**96.37**	**98.25**	94.07	97.22	96.89
IMS:SZEGED:CIS	**95.87**	**98.06**	**96.61**	**97.46**	96.31	97.93	**94.62**	**98.04**	**97.24**
BASE:BKY+POS	95.61	95.25	95.48	97.31	96.03	97.53	92.15	96.97	96.66
BASE:MALT	94.26	95.76	94.23	95.53	95.00	96.09	94.27	95.90	95.35
				2) predicted setting					
IMS:SZEGED:CIS (Bky)	**95.74**	97.07	**96.21**	**97.31**	96.10	**98.03**	94.05	96.92	96.90
IMS:SZEGED:CIS	95.18	**97.67**	96.15	97.09	**96.22**	97.63	**94.43**	**97.50**	**97.02**
BASE:BKY+POS		95.03	95.35	97.12	95.36	97.20	91.34	96.92	96.25
BASE:MALT		95.49	93.84	95.39	94.41	95.72	93.74	96.04	95.09

Table 11: Cross Framework Evaluation: Unlabeled TedEval on generalized gold trees in gold scenario, trained on 5k sentences and tested on 5k terminals.

bic, Hebrew and to some extent French following. It appears that on a typological scale, Semitic and Asian languages are still harder to parse than a range of European languages in terms of structural difficulty and complex morpho-syntactic interaction. That said, note that we cannot tell why certain treebanks appear more challenging to parse then others, and it is still unclear whether the difficulty is inherent on the language, in the currently available models, or because of the annotation scheme and treebank consistency.[43]

[43]The latter was shown to be an important factor orthogonal to the morphologically-rich nature of the treebank's language

8 Conclusion

This paper presents an overview of the first shared task on parsing morphologically rich languages. The task features nine languages, exhibiting different linguistic phenomena and varied morphological complexity. The shared task saw submissions from seven teams, and results produced by more than 14 different systems. The parsing results were obtained in different input scenarios (gold, predicted, and raw) and evaluated using different protocols (cross-framework, cross-scenario, and cross-language). In particular, this is the first time an evaluation campaign reports on the execution of parsers in realistic, morphologically ambiguous, setting.

The best performing systems were mostly ensemble systems combining multiple parser outputs from different frameworks or training runs, or integrating a state-of-the-art morphological analyzer on top of a carefully designed feature set. This is consistent with previous shared tasks such as ConLL 2007 or SANCL'2012. However, dealing with ambiguous morphology is still difficult for all systems, and a promising approach, as demonstrated by ALPAGE:DYALOG, is to deal with parsing and morphology jointly by allowing lattice input to the parser. A promising generalization of this approach would be the full integration of all levels of analysis that are mutually informative into a joint model.

The information to be gathered from the results of this shared task is vast, and we only scratched the surface with our preliminary analyses. We uncovered and documented insights of strategies that make parsing systems successful: parser combination is empirically proven to reach a robust performance across languages, though language-specific strategies are still a sound avenue for obtaining high quality parsers for that individual language. The integration of morphological analysis into the parsing needs to be investigated thoroughly, and new approaches that are morphologically aware need to be developed.

Our cross-parser, cross-scenario, and cross-framework evaluation protocols have shown that, as expected, more data is better, and that performance on gold morphological input is significantly higher than that in more realistic scenarios. We have shown that gold morphological information is more helpful to some languages and parsers than others, and that it may also interact with successful identification of multiword expressions. We have shown that differences between dependency and constituency are smaller than previously assumed and that properties of the learning model and granularity of the output labels are more influential. Finally, we observed that languages which are typologically farthest from English, such as Semitic and Asian languages, are still amongst the hardest to parse, regardless of the parsing method used.

Our cross-treebank, in-depth analysis is still preliminary, owing to the limited time between the end of the shared task and the deadline for publication of this overview. but we nonetheless feel that our findings may benefit researchers who aim to develop parsers for diverse treebanks.[44]

A shared task is an inspection of the state of the art, but it may also accelerate research in an area by providing a stable data basis as well as a set of strong baselines. The results produced in this task give a rich picture of the issues associated with parsing MRLs and initial cues towards their resolution. This set of results needs to be further analyzed to be fully understood, which will in turn contribute to new insights. We hope that this shared task will provide inspiration for the design and evaluation of future parsing systems for these languages.

Acknowledgments

We heartily thank Miguel Ballesteros and Corentin Ribeire for running the dependency and constituency baselines. We warmly thank the Linguistic Data Consortium: Ilya Ahtaridis, Ann Bies, Denise DiPersio, Seth Kulick and Mohamed Maamouri for releasing the Arabic Penn Treebank for this shared task and for their support all along the process. We thank Alon Itai and MILA, the knowledge center for processing Hebrew, for kindly making the Hebrew treebank and morphological analyzer available for us, Anne Abeillé for allowing us to use the French treebank, and Key-Sun Choi for the Kaist Korean Treebank. We thank Grzegorz Chrupała for providing the morphological analyzer *Morfette*, and Joachim

(Schluter and van Genabith, 2007)

[44] The data set will be made available as soon as possible under the license distribution of the shared-task, with the exception of the Arabic data, which will continue to be distributed by the LDC.

Wagner for his *LeafAncestor* implementation. We finally thank Özlem Çetinoğlu, Yuval Marton, Benoit Crabbé and Benoit Sagot who have been nothing but supportive during all that time.

At the end of this shared task (though watch out for further updates and analyses), what remains to be mentioned is our deep gratitude to all people involved, either data providers or participants. Without all of you, this shared task would not have been possible.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a treebank for French. In Anne Abeillé, editor, *Treebanks*. Kluwer, Dordrecht.

Szymon Acedański. 2010. A Morphosyntactic Brill Tagger for Inflectional Languages. In *Advances in Natural Language Processing*, volume 6233 of *Lecture Notes in Computer Science*, pages 3–14. Springer-Verlag.

Meni Adler and Michael Elhadad. 2006. An unsupervised morpheme-based HMM for Hebrew morphological disambiguation. In *Proceedings COLING-ACL*, pages 665–672, Sydney, Australia.

Meni Adler, Yoav Goldberg, David Gabay, and Michael Elhadad. 2008. Unsupervised lexicon-based resolution of unknown words for full morphological analysis. In *Proceedings of ACL-08: HLT*, pages 728–736, Columbus, OH.

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

Itziar Aduriz, José María Arriola, Xabier Artola, A Díaz de Ilarraza, et al. 1997. Morphosyntactic disambiguation for Basque based on the constraint grammar formalism. In *Proceedings of RANLP*, Tzigov Chark, Bulgaria.

Itziar Aduriz, Eneko Agirre, Izaskun Aldezabal, Iñaki Alegria, Xabier Arregi, Jose Maria Arriola, Xabier Artola, Koldo Gojenola, Aitor Maritxalar, Kepa Sarasola, et al. 2000. A word-grammar based morphological analyzer for agglutinative languages. In *Proceedings of COLING*, pages 1–7, Saarbrücken, Germany.

Itziar Aduriz, Maria Jesus Aranzabe, Jose Maria Arriola, Aitziber Atutxa, A Diaz de Ilarraza, Aitzpea Garmendia, and Maite Oronoz. 2003. Construction of a Basque dependency treebank. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 201–204, Växjö, Sweden.

Zeljko Agic, Danijela Merkler, and Dasa Berovic. 2013. Parsing Croatian and Serbian by using Croatian dependency treebanks. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL)*, Seattle, WA.

I. Aldezabal, M.J. Aranzabe, A. Diaz de Ilarraza, and K. Fernández. 2008. From dependencies to constituents in the reference corpus for the processing of Basque. In *Procesamiento del Lenguaje Natural, n^o 41 (2008)*, pages 147–154. XXIV edición del Congreso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN).

Bharat Ram Ambati, Samar Husain, Joakim Nivre, and Rajeev Sangal. 2010. On the role of morphosyntactic features in Hindi dependency parsing. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef van Genabith. 2010. Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL)*, Los Angeles, CA.

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: An optimization tool for MaltParser. In *Proceedings of EACL*, pages 58–62, Avignon, France.

Miguel Ballesteros. 2013. Effective morphological feature selection with MaltOptimizer at the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 53–60, Seattle, WA.

Kepa Bengoetxea and Koldo Gojenola. 2010. Application of different techniques to dependency parsing of Basque. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Philip Bille. 2005. A survey on tree edit distance and related problems. *Theoretical Computer Science*, 337(1–3):217–239, 6.

Anders Björkelund, Ozlem Cetinoglu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (Re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 134–144, Seattle, WA.

Ezra Black, Steven Abney, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith Klavans, Mark Liberman, Mitchell 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 Speech and Natural Language Workshop 1991*, pages 306–311, Pacific Grove, CA.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the EMNLP-CoNLL*, pages 1455–1465, Jeju, Korea.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of COLING*, pages 89–97, Beijing, China.

Adriane Boyd. 2007. Discontinuity revisited: An improved conversion to context-free representations. In *Proceedings of the Linguistic Annotation Workshop*, Prague, Czech Republic.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT)*, pages 24–41, Sozopol, Bulgaria.

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

Tim Buckwalter. 2002. Arabic morphological analyzer version 1.0. Linguistic Data Consortium.

Tim Buckwalter. 2004. Arabic morphological analyzer version 2.0. Linguistic Data Consortium.

Marie Candito and Djamé Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Marie Candito, Benoit Crabbé, and Pascal Denis. 2010. Statistical French dependency parsing: Treebank conversion and first results. In *Proceedings of LREC*, Valletta, Malta.

Xavier Carreras, Michael Collins, and Terry Koo. 2008. TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing. In *Proceedings of CoNLL*, pages 9–16, Manchester, UK.

Eugene Charniak and Mark Johnson. 2005. Course-to-fine n-best-parsing and maxent discriminative reranking. In *Proceedings of ACL*, pages 173–180, Barcelona, Spain.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *AAAI/IAAI*, pages 598–603.

Eugene Charniak. 2000. A maximum entropy inspired parser. In *Proceedings of NAACL*, pages 132–139, Seattle, WA.

Jinho D. Choi and Martha Palmer. 2011. Statistical dependency parsing in Korean: From corpus generation to automatic parsing. In *Proceedings of Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 1–11, Dublin, Ireland.

Jinho D. Choi and Martha Palmer. 2012. Guidelines for the Clear Style Constituent to Dependency Conversion. Technical Report 01-12, University of Colorado at Boulder.

Key-sun Choi, Young S. Han, Young G. Han, and Oh W. Kwon. 1994. KAIST Tree Bank Project for Korean: Present and Future Development. In *In Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 7–14, Nara, Japan.

Jinho D. Choi. 2013. Preparing Korean data for the shared task on parsing morphologically rich languages. arXiv:1309.1649.

Grzegorz Chrupała, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morfette. In *Proceedings of LREC*, Marrakech, Morocco.

Tagyoung Chung, Matt Post, and Daniel Gildea. 2010. Factors affecting the accuracy of Korean parsing. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Volkan Cirik and Hüsnü Şensoy. 2013. The AI-KU system at the SPMRL 2013 shared task: Unsupervised features for dependency parsing. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 68–75, Seattle, WA.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Matthieu Constant, Marie Candito, and Djamé Seddah. 2013. The LIGM-Alpage architecture for the SPMRL 2013 shared task: Multiword expression analysis and dependency parsing. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 46–52, Seattle, WA.

Anna Corazza, Alberto Lavelli, Giogio Satta, and Roberto Zanoli. 2004. Analyzing an Italian treebank with state-of-the-art statistical parsers. In *Proceedings of the Third Workshop on Treebanks and Linguistic Theories (TLT)*, Tübingen, Germany.

Benoit Crabbé and Marie Candito. 2008. Expériences d'analyse syntaxique statistique du français. In *Actes de la 15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN'08)*, pages 45–54, Avignon, France.

Dóra Csendes, János Csirik, Tibor Gyimóthy, and András Kocsor. 2005. The Szeged treebank. In *Proceedings of the 8th International Conference on Text, Speech and Dialogue (TSD)*, Lecture Notes in Computer Science, pages 123–132, Berlin / Heidelberg. Springer.

Eric De La Clergerie. 2013. Exploring beam-based shift-reduce dependency parsing with DyALog: Results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of*

Morphologically-Rich Languages, pages 81–89, Seattle, WA.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. Technical Report CCLS-13-02, Center for Computational Learning Systems, Columbia University.

Eva Ejerhed and Gunnel Källgren. 1997. Stockholm Umeå Corpus. Version 1.0. Department of Linguistics, Umeå University and Department of Linguistics, Stockholm University.

Eva Ejerhed, Gunnel Källgren, Ola Wennstedt, and Magnus Åström. 1992. The linguistic annotation system of the Stockholm–Umeå Corpus project. Technical Report 33, University of Umeå: Department of Linguistics.

Nerea Ezeiza, Iñaki Alegria, José María Arriola, Rubén Urizar, and Itziar Aduriz. 1998. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of COLING*, pages 380–384, Montréal, Canada.

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *Proceedings of ACL*, pages 959–967, Columbus, OH.

Alexander Fraser, Helmut Schmid, Richárd Farkas, Renjing Wang, and Hinrich Schütze. 2013. Knowledge sources for constituent parsing of German, a morphologically rich and less-configurational language. *Computational Linguistics*, 39(1):57–85.

Iakes Goenaga, Koldo Gojenola, and Nerea Ezeiza. 2013. Exploiting the contribution of morphological information to parsing: the BASQUE TEAM system in the SPRML'2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 61–67, Seattle, WA.

Yoav Goldberg and Michael Elhadad. 2010a. Easy-first dependency parsing of Modern Hebrew. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Yoav Goldberg and Michael Elhadad. 2010b. An efficient algorithm for easy-first non-directional dependency parsing. In *Proceedings of HLT: NAACL*, pages 742–750, Los Angeles, CA.

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

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*, Columbus, OH.

Yoav Goldberg, Reut Tsarfaty, Meni Adler, and Michael Elhadad. 2009. Enhancing unlexicalized parsing performance using a wide coverage lexicon, fuzzy tag-set mapping, and EM-HMM-based lexical probabilities. In *Proceedings of EALC*, pages 327–335, Athens, Greece.

Yoav Goldberg. 2011. *Automatic syntactic processing of Modern Hebrew*. Ph.D. thesis, Ben Gurion University of the Negev.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) version 3.1. Linguistic Data Consortium LDC2009E73.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of COLING*, pages 394–402, Beijing, China.

Nathan Green, Loganathan Ramasamy, and Zdenék Žabokrtský. 2012. Using an SVM ensemble system for improved Tamil dependency parsing. In *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, pages 72–77, Jeju, Korea.

Spence Green, Marie-Catherine de Marneffe, and Christopher D. Manning. 2013. Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1):195–227.

Noemie Guthmann, Yuval Krymolowski, Adi Milea, and Yoad Winter. 2009. Automatic annotation of morpho-syntactic dependencies in a Modern Hebrew Treebank. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT)*, Groningen, The Netherlands.

Nizar Habash and Ryan Roth. 2009. CATiB: The Columbia Arabic Treebank. In *Proceedings of ACL-IJCNLP*, pages 221–224, Suntec, Singapore.

Nizar Habash, Ryan Gabbard, Owen Rambow, Seth Kulick, and Mitch Marcus. 2007. Determining case in Arabic: Learning complex linguistic behavior requires complex linguistic features. In *Proceedings of EMNLP-CoNLL*, pages 1084–1092, Prague, Czech Republic.

Nizar Habash, Reem Faraj, and Ryan Roth. 2009a. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009b. MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*. Cairo, Egypt.

Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.

Jan Hajič, Alena Böhmová, Eva Hajičová, and Barbora Vidová-Hladká. 2000. The Prague Dependency Treebank: A three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*. Kluwer Academic Publishers.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. HunPos – an open source trigram tagger. In *Proceedings of ACL*, pages 209–212, Prague, Czech Republic.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülşen Eryiğit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? A study in multilingual parser optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 933–939, Prague, Czech Republic.

Chung-hye Han, Na-Rae Han, Eon-Suk Ko, Martha Palmer, and Heejong Yi. 2002. Penn Korean Treebank: Development and evaluation. In *Proceedings of the 16th Pacific Asia Conference on Language, Information and Computation*, Jeju, Korea.

Tilman Höhle. 1986. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340, Göttingen, Germany.

Zhongqiang Huang, Mary Harper, and Slav Petrov. 2010. Self-training with products of latent variable grammars. In *Proceedings of EMNLP*, pages 12–22, Cambridge, MA.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL*, pages 586–594, Columbus, OH.

Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March.

Mark Johnson. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

Laura Kallmeyer and Wolfgang Maier. 2013. Data-driven parsing using probabilistic linear context-free rewriting systems. *Computational Linguistics*, 39(1).

Fred Karlsson, Atro Voutilainen, Juha Heikkilae, and Arto Anttila. 1995. *Constraint Grammar: a language-independent system for parsing unrestricted text*. Walter de Gruyter.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*, pages 423–430, Sapporo, Japan.

Sandra Kübler, Erhard W. Hinrichs, and Wolfgang Maier. 2006. Is it really that difficult to parse German? In *Proceedings of EMNLP*, pages 111–119, Sydney, Australia, July.

Sandra Kübler, Wolfgang Maier, Ines Rehbein, and Yannick Versley. 2008. How to compare treebanks. In *Proceedings of LREC*, pages 2322–2329, Marrakech, Morocco.

Sandra Kübler. 2008. The PaGe 2008 shared task on parsing German. In *Proceedings of the Workshop on Parsing German*, pages 55–63, Columbus, OH.

Seth Kulick, Ryan Gabbard, and Mitch Marcus. 2006. Parsing the Arabic Treebank: Analysis and Improvements. In *Proceedings of the Treebanks and Linguistic Theories Conference*, pages 31–42, Prague, Czech Republic.

Joseph Le Roux, Benoit Sagot, and Djamé Seddah. 2012. Statistical parsing of Spanish and data driven lemmatization. In *Proceedings of the Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, pages 55–61, Jeju, Korea.

Kong Joo Lee, Byung-Gyu Chang, and Gil Chang Kim. 1997. Bracketing Guidelines for Korean Syntactic Tree Tagged Corpus. Technical Report CS/TR-97-112, Department of Computer Science, KAIST.

Roger Levy and Christopher D. Manning. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of ACL*, Sapporo, Japan.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Hubert Jin. 2004a. Arabic Treebank: Part 2 v 2.0. LDC catalog number LDC2004T02.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004b. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Hubert Jin. 2005. Arabic Treebank: Part 1 v 3.0. LDC catalog number LDC2005T02.

Mohamed Maamouri, Ann Bies, Seth Kulick, Fatma Gaddeche, Wigdan Mekki, Sondos Krouna, and Basma Bouziri. 2009. The Penn Arabic Treebank part 3 version 3.1. Linguistic Data Consortium LDC2008E22.

Wolfgang Maier, Miriam Kaeshammer, and Laura Kallmeyer. 2012. Data-driven PLCFRS parsing revisited: Restricting the fan-out to two. In *Proceedings of the Eleventh International Conference on Tree Adjoining Grammars and Related Formalisms (TAG+11)*, Paris, France.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics*, 19(2):313–330.

Andre Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mario Figueiredo. 2010. Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings of EMNLP*, pages 34–44, Cambridge, MA.

Yuval Marton, Nizar Habash, and Owen Rambow. 2013a. Dependency parsing of Modern Standard Arabic with lexical and inflectional features. *Computational Linguistics*, 39(1):161–194.

Yuval Marton, Nizar Habash, Owen Rambow, and Sarah Alkhulani. 2013b. SPMRL'13 shared task system: The CADIM Arabic dependency parser. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 76–80, Seattle, WA.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of HLT:NAACL*, pages 152–159, New York, NY.

Ryan T. McDonald, Koby Crammer, and Fernando C. N. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*, pages 91–98, Ann Arbor, MI.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Tackstrom, Claudia Bedini, Nuria Bertomeu Castello, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL*, Sofia, Bulgaria.

Igor Mel'čuk. 2001. *Communicative Organization in Natural Language: The Semantic-Communicative Structure of Sentences*. J. Benjamins.

Knowledge Center for Processing Hebrew MILA. 2008. Hebrew morphological analyzer. http://mila.cs.technion.ac.il.

Antonio Moreno, Ralph Grishman, Susana Lopez, Fernando Sanchez, and Satoshi Sekine. 2000. A treebank of Spanish and its application to parsing. In *Proceedings of LREC*, Athens, Greece.

Joakim Nivre and Beáta Megyesi. 2007. Bootstrapping a Swedish treeebank using cross-corpus harmonization and annotation projection. In *Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories*, pages 97–102, Bergen, Norway.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of LREC*, pages 1392–1395, Genoa, Italy.

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, Prague, Czech Republic.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007b. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 Shared Task on Parsing the Web. In *Proceedings of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL), a NAACL-HLT 2012 workshop*, Montreal, Canada.

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

Slav Petrov. 2009. *Coarse-to-Fine Natural Language Processing*. Ph.D. thesis, University of California at Bekeley, Berkeley, CA.

Slav Petrov. 2010. Products of random latent variable grammars. In *Proceedings of HLT: NAACL*, pages 19–27, Los Angeles, CA.

Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. 2012. *Narodowy Korpus Jkezyka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Ines Rehbein and Josef van Genabith. 2007a. Evaluating Evaluation Measures. In *Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007*, Tartu, Estonia.

Ines Rehbein and Josef van Genabith. 2007b. Treebank annotation schemes and parser evaluation for German. In *Proceedings of EMNLP-CoNLL*, Prague, Czech Republic.

Ines Rehbein. 2011. Data point selection for self-training. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 62–67, Dublin, Ireland.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of HLT-NAACL*, pages 129–132, New York, NY.

Geoffrey Sampson and Anna Babarczy. 2003. A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9(04):365–380.

Natalie Schluter and Josef van Genabith. 2007. Preparing, restructuring, and augmenting a French Treebank: Lexicalised parsers or coherent treebanks? In *Proc. of PACLING 07*, Melbourne, Australia.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of LREC*, Lisbon, Portugal.

Djamé Seddah, Grzegorz Chrupała, Ozlem Cetinoglu, Josef van Genabith, and Marie Candito. 2010. Lemmatization and statistical lexicalized parsing of morphologically-rich languages. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL)*, Los Angeles, CA.

Wolfgang Seeker and Jonas Kuhn. 2012. Making ellipses explicit in dependency conversion for a German

treebank. In *Proceedings of LREC*, pages 3132–3139, Istanbul, Turkey.

Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino, and Masaaki Nagata. 2012. Bayesian symbol-refined tree substitution grammars for syntactic parsing. In *Proceedings of ACL*, pages 440–448, Jeju, Korea.

Anthony Sigogne, Matthieu Constant, and Eric Laporte. 2011. French parsing enhanced with a word clustering method based on a syntactic lexicon. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 22–27, Dublin, Ireland.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altmann, and Noa Nativ. 2001. Building a tree-bank of Modern Hebrew text. *Traitement Automatique des Langues*, 42:347–380.

Marek Świdziński and Marcin Woliński. 2010. Towards a bank of constituent parse trees for Polish. In *Proceedings of Text, Speech and Dialogue*, pages 197–204, Brno, Czech Republic.

Ulf Teleman. 1974. *Manual för grammatisk beskrivning av talad och skriven svenska*. Studentlitteratur.

Lucien Tesnière. 1959. *Éléments De Syntaxe Structurale*. Klincksieck, Paris.

Reut Tsarfaty and Khalil Sima'an. 2010. Modeling morphosyntactic agreement in constituency-based parsing of Modern Hebrew. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL)*, Los Angeles, CA.

Reut Tsarfaty, Djame Seddah, Yoav Goldberg, Sandra Kübler, 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)*, Los Angeles, CA.

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

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

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012b. Joint evaluation for segmentation and parsing. In *Proceedings of ACL*, Jeju, Korea.

Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2012c. Parsing morphologically rich languages: Introduction to the special issue. *Computational Linguistics*, 39(1):15–22.

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

Reut Tsarfaty. 2013. A unified morpho-syntactic scheme of Stanford dependencies. In *Proceedings of ACL*, Sofia, Bulgaria.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian Dependency Treebank. In *Proceedings of LREC*, Valletta, Malta.

Joachim Wagner. 2012. *Detecting Grammatical Errors with Treebank-Induced Probabilistic Parsers*. Ph.D. thesis, Dublin City University.

Marcin Woliński, Katarzyna Głowińska, and Marek Świdziński. 2011. A preliminary version of Składnica—a treebank of Polish. In *Proceedings of the 5th Language & Technology Conference*, pages 299–303, Poznań, Poland.

Alina Wróblewska. 2012. Polish Dependency Bank. *Linguistic Issues in Language Technology*, 7(1):1–15.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL:HLT*, pages 188–193, Portland, OR.

János Zsibrita, Veronika Vincze, and Richárd Farkas. 2013. magyarlanc: A toolkit for morphological and dependency parsing of Hungarian. In *Proceedings of RANLP*, pages 763–771, Hissar, Bulgaria.