Joint Morphological and Syntactic Analysis for Richly Inflected Languages

Bernd Bohnet[*] **Joakim Nivre**[⋆] **Igor Boguslavsky**[•◦] **Richárd Farkas**[◇] **Filip Ginter**[†] **Jan Hajič**[‡]

[*]University of Birmingham, School of Computer Science
[⋆]Uppsala University, Department of Linguistics and Philology
[•]Universidad Politécnica de Madrid, Departamento de Inteligencia Artificial
[◦]Russian Academy of Sciences, Institute for Information Transmission Problems
[◇]University of Szeged, Institute of Informatics
[†]University of Turku, Department of Information Technology
[‡]Charles University in Prague, Institute of Formal and Applied Linguistics

Abstract

Joint morphological and syntactic analysis has been proposed as a way of improving parsing accuracy for richly inflected languages. Starting from a transition-based model for joint part-of-speech tagging and dependency parsing, we explore different ways of integrating morphological features into the model. We also investigate the use of rule-based morphological analyzers to provide hard or soft lexical constraints and the use of word clusters to tackle the sparsity of lexical features. Evaluation on five morphologically rich languages (Czech, Finnish, German, Hungarian, and Russian) shows consistent improvements in both morphological and syntactic accuracy for joint prediction over a pipeline model, with further improvements thanks to lexical constraints and word clusters. The final results improve the state of the art in dependency parsing for all languages.

1 Introduction

Syntactic parsing of natural language has witnessed a tremendous development during the last twenty years, especially through the use of statistical models for robust and accurate broad-coverage parsing. However, as statistical parsing techniques have been applied to more and more languages, it has also been observed that typological differences between languages lead to new challenges. In particular, it has been found over and over again that languages exhibiting rich morphological structure, often together with a relatively free word order, usually obtain lower parsing accuracy, especially in comparison to English. One striking demonstration of this tendency can be found in the CoNLL shared tasks on multilingual dependency parsing, organized in 2006 and 2007, where richly inflected languages clustered at the lower end of the scale with respect to parsing accuracy (Buchholz and Marsi, 2006; Nivre et al., 2007). These and similar observations have led to an increased interest in the special challenges posed by parsing morphologically rich languages, as evidenced most clearly by a new series of workshops devoted to this topic (Tsarfaty et al., 2010), as well as a special issue in *Computational Linguistics* (Tsarfaty et al., 2013) and a shared task on parsing morphologically rich languages.[1]

One hypothesized explanation for the lower parsing accuracy observed for richly inflected languages is the strict separation of morphological and syntactic analysis assumed in many parsing frameworks (Tsarfaty et al., 2010; Tsarfaty et al., 2013). This is true in particular for data-driven dependency parsers, which tend to assume that all morphological disambiguation has been performed before syntactic analysis begins. However, as argued by Lee et al. (2011), in morphologically rich languages there is often considerable interaction between morphology and syntax, such that neither can be disambiguated without the other. Lee et al. (2011) go on to show that a discriminative model for joint morphological disambiguation and dependency parsing gives consistent improvements in morphological and syntactic accuracy, compared to a pipeline model, for Ancient Greek, Czech, Hungarian and Latin. Similarly, Bohnet and Nivre (2012) propose a model for

[1]See https://sites.google.com/site/spmrl2013/home/sharedtask.

Transactions of the Association for Computational Linguistics, 1 (2013) 415–428. Action Editor: Brian Roark.
Submitted 7/2013; Revised 9/2013; Published 10/2013.

joint part-of-speech tagging and dependency parsing and report improved accuracy for Czech and German (but also for Chinese and English), although in this case the joint model is limited to basic part-of-speech tags and does not involve the full complex of morphological features. An integrated approach to morphological and syntactic analysis can also be found in grammar-based dependency parsers, such as the ETAP-3 linguistic processor (Apresian et al., 2003), where morphological disambiguation is mostly carried out together with syntactic analysis. Finally, it is worth noting that joint models of morphology and syntax have been more popular in constituency-based statistical parsing (Cowan and Collins, 2005; Tsarfaty, 2006; Cohen and Smith, 2007; Goldberg and Tsarfaty, 2008).

Another hypothesis from the literature is that the high type-token ratio resulting from large morphological paradigms leads to data sparseness when estimating the parameters of a statistical parsing model (Tsarfaty et al., 2010; Tsarfaty et al., 2013). In particular, for many words in the language, only a subset of its morphological forms will be observed at training time. This suggests that using rule-based morphological analyzers or other lexical resources may be a viable strategy to improve coverage and performance. Thus, Goldberg and Elhadad (2013) show that integrating an external wide-coverage lexicon with a treebank-trained PCFG parser improves parsing accuracy for Modern Hebrew, which is in line with earlier studies of part-of-speech tagging for morphologically rich languages (Hajič, 2000). The sparsity of lexical features can also be tackled by the use of distributional word clusters as pioneered by Koo et al. (2008).

In this paper, we present a transition-based model that jointly predicts complex morphological representations and dependency relations, generalizing the approach of Bohnet and Nivre (2012) to include the full range of morphological information. We start by investigating different ways of integrating morphological features into the model, go on to examine the effect of using rule-based morphological analyzers to derive hard or soft constraints on the morphological analysis, and finally add word cluster features to combat lexical sparsity. We evaluate our methods on data from Czech, Finnish, German, Hungarian, and Russian, five morphologically rich languages representing three different language groups. The experiments show that joint prediction of morphology and syntax, rule-based morphological analyzers, and word clusters all contribute to improved parsing accuracy, leading to new state-of-the-art results for all languages.

2 Method

In this section, we define target representations and evaluation metrics (2.1), and describe our transition-based parsing framework, consisting of an abstract transition system (2.2), a feature-based scoring function (2.3), and algorithms for decoding (2.4) and learning (2.5).

2.1 Representations and Metrics

We take an unlabeled dependency tree for a sentence $x = w_1, \ldots, w_n$ to be a directed tree $T = (V_x, A)$, where $V_x = \{0, 1, ..., n\}$, $A \subseteq V_x \times V_x^+$, and 0 is the root of the tree (Kübler et al., 2009). The set V_x of nodes is the set of positive integers up to and including n, each corresponding to the linear position of a word in the sentence, plus an extra artificial root node 0. We use V_x^+ to denote $V_x - \{0\}$. The set A of arcs is a set of pairs (i, j), where i is the head node and j is the dependent node.

To this basic representation of syntactic structure we add four labeling functions for part-of-speech tags, morphological features, lemmas, and dependency relations. The function $\pi : V_x^+ \rightarrow P$ maps each node in V_x^+ to a part-of-speech tag in the set P; the function $\mu : V_x^+ \rightarrow M$ maps each node to a morphological description in the set M; the function $\lambda : V_x^+ \rightarrow Z^*$ maps each node in V_x^+ to a lemma (a string over some character set Z); and the function $\delta : A \rightarrow D$ maps each arc to a dependency label in the set D. The exact nature of P, M and D depends on the data sets used, but normally P and D only contain atomic labels while the members of M are sets of atomic features encoding properties like number, case, tense, etc. For lemmas, we do not assume that there is a fixed lexicon but allow any character string as a legal value.

We define our target representation for a sentence $x = w_1, \ldots, w_n$ as a quintuple $\Gamma = (A, \pi, \mu, \lambda, \delta)$ such that (V_x, A) is an unlabeled dependency tree; π, μ and λ label the nodes with part-of-speech tags,

Transition		Condition
LEFT-ARC$_d$	$([\sigma\|i,j],B,\Gamma) \Rightarrow ([\sigma\|j],B,\Gamma[(j,i)\in A,\delta(j,i)=d])$	$i \neq 0$
RIGHT-ARC$_d$	$([\sigma\|i,j],B,\Gamma) \Rightarrow ([\sigma\|i],B,\Gamma[(i,j)\in A,\delta(i,j)=d])$	
SHIFT$_{p,m,l}$	$(\sigma,[i\|\beta],\Gamma) \Rightarrow ([\sigma\|i],\beta,\Gamma[\pi(i)=p,\mu(i)=m,\lambda(i)=l])$	
SWAP	$([\sigma\|i,j],\beta,\Gamma) \Rightarrow ([\sigma\|j],[i\|\beta],\Gamma)$	$0 < i < j$

Figure 1: Transitions for joint morphological and syntactic analysis. The stack Σ is represented as a list with its head to the right (and tail σ) and the buffer B as a list with its head to the left (and tail β). The notation $\Gamma[q_1,\ldots,q_m]$ is used to denote an MS-parse that is exactly like Γ except that $q_1,\ldots,q_m$ hold true.

morphological features and lemmas; and δ labels the arcs with dependency relations. For convenience, we refer to this type of structure as a morphosyntactic parse (or MS-parse, for short). The following evaluation metrics are used to score an MS-parse with respect to a gold standard:

1. POS: The percentage of nodes in V_x^+ that have the correct part-of-speech tag.
2. MOR: The percentage of nodes in V_x^+ that have the correct morphological description; if the description is set-valued, all members of the set must match exactly.
3. LEM: The percentage of nodes in V_x^+ that have the correct lemma.
4. UAS: The percentage of nodes in V_x^+ that have the correct incoming arc.
5. LAS: The percentage of nodes in V_x^+ that have the correct incoming arc with the correct label.
6. PM: The percentage of nodes in V_x^+ that have the correct part-of-speech tag and the correct morphological description.
7. PMD: The percentage of nodes in V_x^+ that have the correct part-of-speech tag, the correct morphological description, and the correct incoming arc with the correct label.

The POS, UAS and LAS metrics are standard in the dependency parsing literature; the additional metrics will provide us with a more fine-grained picture of the (joint) morphological and syntactic accuracy. All evaluation scores are computed over all tokens, including punctuation. We test statistical significance primarily for the PMD metric, using a two-tailed paired t-test.

2.2 Transition System

A transition system for dependency parsing is a quadruple $S = (C, T, c_s, C_t)$, where C is a set of configurations, T is a set of transitions, each of which is a (partial) function $t : C \rightarrow C$, c_s is an initialization function, mapping a sentence x to a configuration $c \in C$, and $C_t \subseteq C$ is a set of terminal configurations. A transition sequence for a sentence x in S is a sequence of configuration-transition pairs $C_{0,m} = [(c_0,t_0),(c_1,t_1),\ldots,(c_m,t_m)]$ where $c_0 = c_s(x)$, $t_m(c_m) \in C_t$, and $t_i(c_i) = c_{i+1} (0 \leq i < m)$.

In our model for joint prediction of part-of-speech tags, morphological features and dependency trees, the set C of configurations consists of all triples $c = (\Sigma, B, \Gamma)$ such that Σ (the stack) and B (the buffer) are disjoint sublists of the nodes V_x of some sentence x, and $\Gamma = (A, \pi, \mu, \lambda, \delta)$ is an MS-parse for x. We take the initial configuration for a sentence $x = w_1,\ldots,w_n$ to be $c_s(x) = ([0],[1,\ldots,n],(\emptyset,\perp,\perp,\perp,\perp))$, where $\perp$ is the function that is undefined for all arguments, and we take the set C_t of terminal configurations to be the set of all configurations of the form $c = ([0],[\,],\Gamma)$ (for any Γ). The MS-parse defined for x by $c = (\Sigma, B, (A, \pi, \mu, \lambda, \delta))$ is $\Gamma_c = (A, \pi, \mu, \lambda, \delta)$, and the MS-parse defined for x by a complete transition sequence $C_{0,m}$ is $\Gamma_{t_m(c_m)}$.

The set T of transitions is shown in Figure 1. It is based on the system of Nivre (2009), where a dependency tree is built by repeated applications of the LEFT-ARC$_d$ and RIGHT-ARC$_d$ transitions, which add an arc (with some label $d \in D$) between the two topmost nodes on the stack (with the leftmost or rightmost node as the dependent, respectively). The SHIFT transition is used to move nodes from the buffer to the stack, and the SWAP transition is used

to permute nodes in order to allow non-projective dependencies. Bohnet and Nivre (2012) modified this system by replacing the simple SHIFT transition by SHIFT_p, which not only moves a node from the buffer to the stack but also assigns it a part-of-speech tag p, turning it into a system for joint part-of-speech tagging and dependency parsing.[2] Here we add two additional parameters m and l to the SHIFT transition, so that a node moved from the buffer to the stack is assigned not only a tag p but also a morphological description m and a lemma l. In this way, we get a joint model for the prediction of part-of-speech tags, morphological features, lemmas, and dependency trees.

2.3 Scoring

In transition-based parsing, we score parses in an indirect fashion by scoring transition sequences. In general, we assume that the score function s factors by configuration-transition pairs:

$$s(x, C_{0,m}) = \sum_{i=0}^{m} s(x, c_i, t_i) \tag{1}$$

Moreover, when using structured learning, as first proposed for transition-based parsing by Zhang and Clark (2008), we assume that the score is given by a linear model whose feature representations decompose in the same way:

$$\begin{aligned} s(x, C_{0,m}) &= \mathbf{f}(x, C_{0,m}) \cdot \mathbf{w} \\ &= \sum_{i=0}^{m} \mathbf{f}(x, c_i, t_i) \cdot \mathbf{w} \end{aligned} \tag{2}$$

Here, $\mathbf{f}(x, c, t)$ is a high-dimensional feature vector, where each component $\mathbf{f}_i(x, c, t)$ is a nonnegative numerical feature (usually binary), and $\mathbf{w}$ is a weight vector of the same dimensionality, where each component $\mathbf{w}_i$ is the real-valued weight of the feature $\mathbf{f}_i(x, c, t)$. The choice of features to include in $\mathbf{f}(x, c, t)$ is discussed separately for each instantiation of the model in Sections 4–6.

[2] Hatori et al. (2011) previously made the same modification to the arc-standard system (Nivre, 2004), without the SWAP transition. Similarly, Titov and Henderson (2007) added a word parameter to the SHIFT transition to get a joint model of word strings and dependency trees. A similar model was considered but finally not used by Gesmundo et al. (2009).

2.4 Decoding

Exact decoding for transition-based parsing is hard in general.[3] Early transition-based parsers mostly relied on greedy, deterministic decoding, which makes for very efficient parsing (Yamada and Matsumoto, 2003; Nivre, 2003), but research has shown that accuracy can be improved by using beam search instead (Zhang and Clark, 2008; Zhang and Nivre, 2012). While still not exact, beam search decoders explore a larger part of the search space than greedy parsers, which is likely to be especially important for joint models, where the search space is larger than for plain dependency parsing without morphology (even more so with the SWAP transition for non-projectivity). Figure 2 outlines the beam search algorithm used for decoding with our model. Different instantiations of the model will require slightly different implementations of the permissibility condition invoked in line 8, which can be used to filter out labels that are improbable or incompatible with an external lexicon, and the pruning step performed in line 13, where there may be a need to balance the amount of morphological and syntactic variation in the beam. Both these aspects will be discussed in depth in Sections 4–6.

Although the worst-case running time with constant beam size is quadratic in sentence length, the observed running time is linear for natural language data sets, due to the sparsity of non-projective dependencies (Nivre, 2009). The running time is also linear in $|D| + |P \times M|$, which means that joint prediction only gives a linear increase in running time, often quite marginal because $|D| > |P \times M|$. This assumes that the lemma is predicted deterministically given a tag and a morphological description, an assumption that is enforced in all our experiments.

2.5 Learning

In order to learn a weight vector $\mathbf{w}$ from a training set of sentences with gold parses, we use a variant of the structured perceptron, introduced by Collins (2002) and first used for transition-based parsing by Zhang and Clark (2008). We initialize all weights

[3] While there exist exact dynamic programming algorithms for projective transition systems (Huang and Sagae, 2010; Kuhlmann et al., 2011) and even for restricted non-projective systems (Cohen et al., 2011), parsing is intractable for systems like ours that permit arbitrary non-projective trees.

PARSE$(x, \mathbf{w})$
1 $h_0.c \leftarrow c_s(x)$
2 $h_0.s \leftarrow 0.0$
3 $h_0.\mathbf{f} \leftarrow \{0.0\}^{dim(\mathbf{w})}$
4 BEAM $\leftarrow [h_0]$
5 **while** $\exists h \in$ BEAM $: h.c \notin C_t$
6 TMP $\leftarrow [\,]$
7 **foreach** $h \in$ BEAM
8 **foreach** $t \in T :$ PERMISSIBLE$(h.c, t)$
9 $h.\mathbf{f} \leftarrow h.\mathbf{f} + \mathbf{f}(x, h.c, t)$
10 $h.s \leftarrow h.s + \mathbf{f}(x, h.c, t) \cdot \mathbf{w}$
11 $h.c \leftarrow t(h.c)$
12 TMP $\leftarrow$ INSERT$(h,$ TMP$)$
13 BEAM $\leftarrow$ PRUNE(TMP)
14 $h^* \leftarrow$ TOP(BEAM)
15 **return** $\Gamma_{h^*_c}$

Figure 2: Beam search algorithm for finding the best MS-parse for input sentence x with weight vector $\mathbf{w}$. The symbols $h.c$, $h.s$ and $h.\mathbf{f}$ denote, respectively, the configuration, score and feature vector of a hypothesis h; Γ_c denotes the MS-parse defined by c.

to 0.0, make N iterations over the training data and update the weight vector for every sentence x where the transition sequence $C_{0,m}$ corresponding to the gold parse is different from the highest scoring transition sequence $C^*_{0,m'}$.[4] More precisely, we use the passive-aggressive update of Crammer et al. (2006). We also use the early update strategy found beneficial for parsing in several previous studies (Collins and Roark, 2004; Zhang and Clark, 2008; Huang and Sagae, 2010). This means that, at learning time, we terminate the beam search as soon as the hypothesis corresponding to the gold parse is pruned from the beam and then update with respect to the partial transition sequences constructed up to that point. Finally, we use the standard technique of averaging over all weight vectors seen in training, as originally proposed by Collins (2002).

[4]Note that there may be more than one transition sequence corresponding to the gold parse, in which case we pick the canonical transition sequence that processes all left-dependents before right-dependents and applies the lazy swapping strategy of Nivre et al. (2009).

3 Data Sets and Resources

Throughout the paper, we experiment with data from five languages: Czech, Finnish, German, Hungarian, and Russian. For each language, we use a morphologically and syntactically annotated corpus (treebank), divided into a training set, a development set and a test set. In addition, we use a lexicon generated by a rule-based morphological analyzer, and distributional word clusters derived from a large unlabeled corpus. Below we describe the specific resources used for each language. Table 1 provides descriptive statistics about the resources.

Czech For training and test we use the Prague Dependency Treebank (Hajič et al., 2001; Böhmová et al., 2003), Version 2.5, converted to the format used in the CoNLL 2009 shared task (Hajič et al., 2009). The morphological lexicon comes from Hajič and Hladká (1998),[5] and word clusters are derived from a large web corpus (Spoustová and Spousta, 2012).

Finnish The training set is from the Turku Dependency Treebank (Haverinen et al., 2013), and the test set is the hidden test set maintained by the treebank developers. It is worth noting that, while the entire treebank has manually validated syntactic annotation, the morphological annotation is automatic except for a subset of 1204 tokens in the test set, which will be used to estimate the POS, MOR, LEM, PM and PMD scores. The estimated accuracy of the automatic annotation is 97.3% POS and 94.8% PM (Haverinen et al., 2013). Also, because of the limited amount of data, we do not use a development set for Finnish but instead use cross-validation on the training set when tuning parameters. We use the open-source morphological analyzer OMorFi (Pirinen, 2011) and word clusters derived from the entire Finnish Wikipedia.[6]

German Training and test sets are from the Tiger Treebank (Brants et al., 2002) in the improved dependency conversion by Seeker et al. (2010). We use the SMOR morphological analyzer (Schmid et al., 2004), but because the tags and morphological features in the lexicon are not the same as in the

[5]Downloaded from the http://lindat.cz repository as resource PID http://hdl.handle.net/11858/00-097C-0000-0015-A780-9.

[6]Downloaded in March 2012.

	Treebank						Morphology		Clusters	
	Train	Dev	Test	P	M	D	Forms	Lemmas	Tokens	Types
Czech	652,544	87,988	70,348	12	1851	49	98,360	42,058	628,332,859	477,185
Finnish	183,118	–	21,211	12	1917	47	57,127	25,280	50,207,300	257,984
German	648,296	32,065	31,692	54	257	43	76,729	55,220	1,327,701,182	1,621,083
Hungarian	1,101,871	210,068	171,466	22	1105	33	151,971	71,263	200,249,814	538,138
Russian	575,400	72,893	71,664	14	454	78	97,905	35,039	195,897,041	639,446

Table 1: Statistics about data sets and resources used in the experiments. Treebank: number of tokens in data sets; number of labels in label sets. Morphology: number of word forms and lemmas in treebank covered by morphological analyzer. Clusters: number of tokens and types in unlabeled corpus.

treebank annotation we have to rely on a heuristic mapping between the two. Word clusters are derived from the so-called Huge German Corpus.[7]

Hungarian For training and test we use the Szeged Dependency Treebank (Farkas et al., 2012). We use a finite-state morphological analyzer constructed from the morphdb.hu lexical resource (Trón et al., 2006), and word clusters come from the Hungarian National Corpus (Váradi, 2002).

Russian Parsers are trained and tested on data from the SynTagRus Treebank (Boguslavsky et al., 2000; Boguslavsky et al., 2002). The morphological analyzer is a module of the ETAP-3 linguistic processor (Apresian et al., 2003) with a dictionary comprising more than 130,000 lexemes (Iomdin and Sizov, 2008). Word clusters have been produced on the basis of an unlabeled corpus of Russian compiled by the Russian Language Institute of the Russian Academy of Sciences and tokenized by the ETAP-3 analyzer.

4 Joint Morphology and Syntax

We start by exploring different ways of integrating morphology and syntax in a data-driven setting, that is, where our only knowledge source is the annotated training corpus. At both learning and parsing time, we preprocess sentences using a tagger that assigns (up to) k_p part-of-speech tags and k_m morphological descriptions and a lemmatizer that assigns a single best lemma to each word. Complex morphological descriptions consisting of several atomic features are predicted as a whole, both in preprocessing and in parsing. Although it would be possible to predict each atomic morphological feature separately, we believe this would increase the risk of creating inconsistent morphological descriptions. As preprocessors, we use the tagger and lemmatizer included in the MATE tools[8] trained on the same annotated training set, using 10-fold jack-knifing to get predictions for the training set itself. The tagger is a greedy left-to-right tagger trained with the same passive-aggressive online learning as the parsing system, which is run twice over the input to make more effective use of contextual features. The tagger scores are not properly normalized but tend to be in the [0,1] range for both part-of-speech tags and morphological descriptions. In this setting, we consider four different models for deriving a full MS-parse:

1. In the PIPELINE model, we set $k_p = k_m = 1$, which means that the SHIFT transition always selects the 1-best tag, morphological description and lemma for each word. We use a beam size of 40 and prune by simply keeping the 40 highest scoring hypotheses at each step. As the name suggests, this is equivalent to a standard pipeline with no joint prediction.

2. The SIMPLETAG model replicates the model of Bohnet and Nivre (2012) with $k_p = 2$, $k_m = 1$, and a score threshold for tags of 0.25, meaning that the second best tag is only considered if its score is less than 0.25 below that of the best tag. We use two-step beam pruning, where we first extract the 40 highest scoring hypotheses with distinct dependency trees and then add the 8 highest scoring remaining hypotheses (normally morphological variants of hypotheses already included) for a total beam size of 48. This

[7] See http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/hgc.html.

[8] Available at https://code.google.com/p/mate-tools/.

model performs joint tagging and parsing but relies on 1-best morphological features.

3. The COMPLEXTAG model is like SIMPLETAG except that we let tags represent the concatenation of ordinary tags and morphological descriptions (and retrain the preprocessing tagger on this representation). This model performs joint morphological and syntactic analysis as joint tagging and parsing with a fine-grained tag set.

4. The JOINT model has $k_p = k_m = 2$, meaning that the tag and the morphological description can be selected independently by the parser. For morphological descriptions, we use a score threshold of 0.1. For beam pruning, we generalize the previous method by first extracting the 40 highest-scoring hypotheses with distinct dependency trees. For each of these, we then find the highest-scoring hypothesis with the same dependency tree but different tags or morphological features, storing these in two temporary lists TMP_p, for hypotheses that differ with respect to tags, and TMP_m, for hypotheses that differ only with respect to morphological features. Finally, we extract the 8 highest-scoring hypotheses from each of TMP_p and TMP_m and add them to the beam for a total beam size of 56. This model performs joint prediction of part-of-speech tags, morphological descriptions and dependency relations (but still relies on 1-best lemmas, like all the other models.)

The procedures for beam pruning may appear both complex and ad hoc, especially for the JOINT model, but are motivated by the need to achieve a balance between morphological and syntactic ambiguity in the set of hypotheses maintained. As explained by Bohnet and Nivre (2012), just maintaining a single beam does not give enough variety in the beam. The method used for the JOINT model is one way of generalizing this technique to a fully joint model, but other strategies are certainly conceivable.

Another point that may be surprising is the choice to keep k_p and k_m as low as 2, which is fairly close to a pipeline model. Bohnet and Nivre (2012) experimented with higher values for the tag threshold but found no improvement in accuracy, and our own preliminary experiments confirmed this trend for morphological descriptions. In Section 7, we present an empirical analysis that gives further support for this choice, at least for the languages considered in this paper. Note also that the choice is not motivated by efficiency concerns, since increasing the values of k_p and k_m has only a marginal effect on running time, as explained in Section 2.4. Finally, the choice not to consider k-best lemmas is dictated by the fact that our lemmatizer only provides a 1-best analysis.

For the first three models, we use the same feature representations as Bohnet and Nivre (2012),[9] consisting of their adaptation of the features used by Zhang and Nivre (2011), the graph completion features of Bohnet and Kuhn (2012), and the special features over k-best tags introduced specifically for joint tagging and parsing by Bohnet and Nivre (2012). For the JOINT model, we simply add features over the k-best morphological descriptions analogous to the features over k-best tags.[10]

Experimental results for these four models can be found in Table 2. From the PIPELINE results, we see that the 1-best accuracy of the preprocessing tagger ranges from 95.0 (Finnish) to 99.2 (Czech) for POS, and from 89.4 (Finnish) to 96.5 (Hungarian) for MOR. The lemmatizer does a good job for four of the languages (93.9–97.9) but has really poor performance on Finnish (73.7). With respect to syntactic accuracy, the PIPELINE system achieves LAS ranging from 79.9 (Finnish) to 91.8 (German) and UAS ranging from 84.4 to 93.7. It is interesting to note that the highest PMD score, which requires both morphology and syntax to be completely correct, is observed for Hungarian (86.2).

Turning to the results for SIMPLETAG, we note that our results are consistent with those reported by Bohnet and Nivre (2012), with small but consistent improvements in POS and UAS/LAS (and in the compound metrics PM and PMD) for most languages. However, the improvement in the PMD score is statistically significant only for Hungarian and Russian ($p < 0.01$). By contrast, the results for COMPLEXTAG confirm our hypothesis that merging tags and morphological descriptions into a single tag is not an effective way to do joint morphological and

[9] See http://stp.lingfil.uu.se/∼nivre/exp/emnlp12.html.

[10] A complete description of our feature representations is available at http://stp.lingfil.uu.se/∼nivre/exp/tacl13.html.

Czech	POS	MOR	LEM	UAS	LAS	PM	PMD
PIPELINE	99.2	93.2	95.5	88.5	83.1	93.0	78.4
SIMPLETAG	99.2	93.2	95.5	88.5	83.2	93.1	78.4
COMPLEXTAG	98.8	93.3	95.5	87.1	81.2	93.3	77.3
JOINT	99.2	93.7	95.5	88.7	83.3	93.7	79.2
LEXHARD	99.3	93.0	94.3	88.7	83.4	92.9	78.3
LEXSOFT	99.4	94.5	95.9	88.8	83.5	94.4	79.8
CLUSTER	**99.4**	**94.6**	**96.0**	**89.0**	**83.7**	**94.5**	**80.0**
ORACLE	99.8	97.0	–	92.7	89.9	94.1	84.7
GOLD	–	–	–	89.3	84.5	–	–
Finnish	**POS**	**MOR**	**LEM**	**UAS**	**LAS**	**PM**	**PMD**
PIPELINE	95.0	89.4	73.7	84.4	79.9	88.8	71.5
SIMPLETAG	95.6	89.4	73.7	84.8	80.5	89.0	73.0
COMPLEXTAG	93.0	84.9	73.7	80.1	74.5	84.7	65.3
JOINT	95.4	89.2	73.7	84.8	80.6	89.1	72.6
LEXHARD	**95.8**	91.6	93.4	86.1	82.5	91.1	**75.9**
LEXSOFT	95.5	91.9	93.0	86.0	82.3	**91.6**	75.7
CLUSTER	95.7	**92.0**	**94.4**	**86.6**	**83.1**	91.4	75.8
ORACLE	98.0	94.8	–	91.3	89.4	91.8	83.1
German	**POS**	**MOR**	**LEM**	**UAS**	**LAS**	**PM**	**PMD**
PIPELINE	97.6	90.0	**97.9**	93.7	91.8	89.1	82.9
SIMPLETAG	98.0	90.0	**97.9**	93.8	91.9	89.1	83.0
COMPLEXTAG	97.3	87.6	**97.9**	92.3	90.1	86.9	79.7
JOINT	98.1	90.8	**97.9**	93.9	92.0	90.0	83.9
LEXHARD	97.0	65.6	**97.9**	93.7	92.0	64.6	61.3
LEXSOFT	**98.4**	91.9	**97.9**	94.0	92.1	91.2	85.1
CLUSTER	**98.4**	**92.5**	**97.9**	**94.1**	**92.4**	**91.7**	**85.9**
ORACLE	99.6	96.3	–	96.3	95.9	91.9	88.8
GOLD	–	–	–	94.2	92.7	–	–
Hungarian	**POS**	**MOR**	**LEM**	**UAS**	**LAS**	**PM**	**PMD**
PIPELINE	97.6	96.5	93.9	91.0	88.4	96.1	86.2
SIMPLETAG	97.8	96.5	93.9	91.3	88.8	96.1	86.6
COMPLEXTAG	97.5	90.9	93.9	90.6	87.7	90.9	81.2
JOINT	97.8	96.4	93.7	91.3	88.9	96.2	86.7
LEXHARD	**98.5**	97.3	**99.0**	91.5	89.1	97.1	87.4
LEXSOFT	**98.5**	**97.6**	**99.0**	91.4	89.1	**97.4**	87.7
CLUSTER	**98.5**	**97.6**	**99.0**	**91.7**	**89.3**	**97.4**	**88.0**
ORACLE	99.7	99.3	–	94.6	93.3	97.6	91.7
GOLD	–	–	–	91.9	89.8	–	–
Russian	**POS**	**MOR**	**LEM**	**UAS**	**LAS**	**PM**	**PMD**
PIPELINE	98.4	94.0	96.1	92.6	87.4	92.6	82.7
SIMPLETAG	98.5	94.0	96.1	92.6	87.5	92.6	82.9
COMPLEXTAG	97.9	91.4	96.1	91.2	85.1	90.8	79.0
JOINT	98.5	94.4	96.1	92.8	87.6	92.8	83.5
LEXHARD	**98.9**	95.1	94.0	**93.0**	**88.0**	94.5	84.1
LEXSOFT	98.8	**95.7**	96.5	92.9	87.7	95.1	84.5
CLUSTER	98.8	**95.7**	**96.6**	**93.0**	87.9	**95.7**	**84.7**
ORACLE	99.9	98.6	–	95.5	92.9	95.2	89.0
GOLD	–	–	–	94.0	89.1	–	–

Table 2: Test set results for all models. ORACLE = oracle scores for LEXSOFT; GOLD = accuracy for PIPELINE with gold POS, MOR, LEM. Bold marks best result per column and language (excluding ORACLE and GOLD).

syntactic analysis. Here, we see a significant drop in most scores for all languages, but in particular in the accuracy of morphological descriptions (MOR), where the score drops by 5.6 percentage points for Hungarian, 4.5 for Finnish, 2.6 for Russian, and 2.4 for German. The only exception is Czech, where MOR and PM actually improve slightly, but this comes at the expense of a substantial drop in dependency accuracy. In any case, the decrease in PMD is highly significant for all languages ($p < 0.01$).

Finally, we see that the JOINT model, where tags and morphological descriptions are predicted separately during the parsing process, gives significant improvements in MOR accuracy compared to the PIPELINE and SIMPLETAG models for German (+0.8), Czech (+0.5), and Russian (+0.4), with marginal improvements also in the syntactic UAS and LAS scores. For Finnish and Hungarian, on the other hand, there is actually a small drop in accuracy (and for Finnish also a drop in POS accuracy compared to SIMPLETAG). Interestingly, however, for both these languages there is nevertheless a small improvement in the joint PM score, indicating that the JOINT model in general does a better job at selecting a valid complete morphological description than the SIMPLETAG model. Since Finnish and Hungarian are the most morphologically complex languages, it is likely that the lack of a strong positive effect is due in part to sparse data, especially for Finnish where the training set is small. As we shall see in the next section, this problem can be partly overcome through the use of external lexical resources. Still, the improvement in the PMD score over the other three models is highly significant for all languages except Finnish ($p < 0.01$).

5 Lexical Constraints

Our starting point in this section is the JOINT model, which gave the best overall accuracy score (PMD) for all languages except Finnish. To this model we now add constraints derived from a morphological lexicon that maps each word form to a set of possible tags, morphological descriptions and lemmas. We explore two different ways of integrating these constraints:

1. In the LEXHARD model, we use the lexicon to derive hard constraints and filter out tags and

morphological descriptions that are not in the lexicon. More precisely, for word forms that are covered by the lexicon, we let the preprocessing tagger select the k_p best tags and k_m best morphological descriptions that are in the lexicon. We do this both during training and parsing, and we use exactly the same features and beam handling as for the JOINT model in the previous section.

2. In the LEXSOFT model, we instead use soft lexical constraints by adding features that encode whether a tag or morphological description is in the lexicon or not. Again, we add these features both to the preprocessing tagger and to the joint parser, which otherwise remain exactly as before.

One additional modification that we make for both the LEXHARD and the LEXSOFT model is to completely rely on the external lexicon for the prediction of lemmas. After the parser has selected a tag and morphological description for a word, we simply predict the corresponding lemma from the lexicon, breaking ties arbitrarily in the very few cases where the word form, tag and morphological description do not determine a unique lemma, and leaving the lemma empty for word forms that are not contained in the lexicon. This means that, in contrast to the purely data-driven models, the lexicon-enriched models predict the complete morphological analysis jointly with parsing (with the lemma being derived deterministically from the tag and the morphological description). We make an exception only for German, where the lexicon provides lemmas that would require further disambiguation and where we therefore continue to use the data-driven lemmatizer.

As can be seen in Table 2, the results for the LEXHARD model are somewhat mixed. For Finnish, we see a dramatic improvement of the LEM score (from 73.7 to 93.4), indicating that the rule-based morphological analyzer is vastly superior to the data-driven lemmatizer for Finnish. There is also a very nice boost to the MOR score (+2.2) and a smaller improvement on POS (+0.4). These improvements also lead to higher syntactic accuracy, with LAS increasing from 80.6 to 82.5 and UAS from 84.8 to 86.1. For Hungarian, we have nice improvements of the LEM score (+5.3), the MOR score (+0.9) and the POS score (+0.7), but only small improvements in LAS/UAS. For Russian, we observe improvements in POS and MOR, a small drop in LEM, and again minor improvements in UAS/LAS. For Czech and German, finally, we see a drop in MOR (and in LEM for Czech and POS for German), while UAS/LAS is largely unaffected. For German, this result can probably be explained largely by the fact that the morphological descriptions in the lexicon are not fully compatible with those in the treebank, as explained in Section 3. Similarly, for Czech, we think the drop in the LEM score is due to discrepancies caused by updates in the dictionary version released in 2013, deviating from the previously published treebank.

In general, the LEXSOFT model performs considerably better, achieving the best results so far for most languages and metrics. The only clear exception is Finnish, where it performs slightly worse than LEXHARD (but better than all the other models). In addition, there is a marginal drop in POS and LAS/UAS for Russian and in UAS for Hungarian (but again only compared to LEXHARD). The results are particularly striking for German, where the soft lexical constraints are clearly beneficial (especially for the MOR score) despite not being quite compatible with the morphological descriptions in the training set. In terms of statistical signifance, LEXSOFT outperforms the JOINT model with respect to the PMD score for all languages ($p < 0.01$). It is also significantly better than LEXHARD for all languages except Finnish ($p < 0.01$).

6 Word Clusters

Finally, we add word cluster features to the best model for each language (LEXHARD for Finnish, LEXSOFT for the others).[11] We use Brown clusters (Brown et al., 1992), with 800 clusters for all languages, and we use the same feature representation as Bohnet and Nivre (2012). The results in Table 2 show small but consistent improvements in almost all metrics for all languages, confirming the benefit of cluster features for morphologically rich languages. It is worth noting that we see the biggest improvement for Finnish, the language with the smallest training set and therefore most likely to

[11] The best model was selected according to results on the dev set (cross-validation on the training set for Finnish).

suffer from sparse data, where the syntactic accuracy improves substantially (LAS +0.6, UAS +0.5) and lemmatization even more (LEM +1.0). We also see a nice improvement in morphological accuracy for German (MOR +0.6, PM +0.5), which may be related to the lack of a compatible morphological analyzer for this language or simply to the fact that the clusters are derived from a much larger corpus for German than for the other languages. The PMD improvement is statistically significant for all languages except Finnish ($p < 0.01$).

7 Discussion

The experimental results generally support the conclusion that joint prediction of morphology and syntax, where morphology includes rich morphological features as well as basic part-of-speech tags, improves both morphological and syntactic accuracy. The effect is especially clear on the joint evaluation metrics PM and PMD, which indicates that the joint model produces more internally consistent representations. However, we also see evidence that the joint model may suffer from data sparsity, as in the case of Finnish, where a model that only predicts part-of-speech tags jointly with dependency relations achieve better accuracy on some metrics. However, even in this case, the joint model has the best results on the joint evaluation metrics.

The second conclusion that can be drawn from the experiments is that the use of an external lexicon is an effective way of mitigating the sparse data problem and thereby improving accuracy. In general, however, it is more effective to add the lexical constraints in the form of features, or soft constraints, than to apply them as hard constraints and discard all analyses that are not licensed by the lexicon. In particular, this is a useful strategy when the lexical resource is not completely compatible with the annotation in the training set, as seen in the case of German and (to a lesser extent) Czech. The only exception to this generalization is again Finnish, where the hard constraint model works marginally better (except for the MOR and PM metrics), which may again indicate that the training set is too small to make optimal use of the additional features. Still, the soft constraint model improves substantially over the models without lexical resources also for Finnish.

Finally, our experiments confirm that features based on distributional word clusters have a positive impact on syntactic accuracy, but little or no impact on morphological accuracy. This is consistent with previous findings in the literature, mainly from English (Koo et al., 2008; Sagae and Gordon, 2009), and it is interesting to see that it holds also for richly inflected languages and when added on top of features derived from external lexical resources.

One issue worth discussing is the choice to allow the joint model to consider at most 2 tags and 2 morphological descriptions per word, which may seem overly restrictive and very close to a pipeline model. As already mentioned, this was motivated by the results of Bohnet and Nivre (2012), which explored higher values without seeing any improvements, as well as by our own preliminary experiments. In an attempt to shed further light on this issue, we computed oracle scores for the LEXSOFT model, which uses soft lexical constraints but no cluster features. The oracle scores for POS and MOR tell us how often the correct analysis is actually included in the input to the joint model, while the oracle scores for UAS and LAS reports the score of the best dependency tree present in the beam at termination. The results, reported in Table 2, show that the oracle scores are very high, especially for part-of-speech tags (98.0–99.9) but also for morphological descriptions (94.8–99.3). Hence, very few correct analyses are pruned away when setting the k_p and k_m parameters to 2, and increasing the search space further is therefore unlikely to improve accuracy.

For further analysis, Table 2 reports the UAS/LAS scores of the PIPELINE system when given gold standard tags, morphological descriptions and lemmas as input.[12] Viewing this as an upper bound on improvements in parsing accuracy for the joint models, and comparing with the LEXSOFT model, which like PIPELINE does not use cluster features, we see that joint prediction with (soft) lexical constraints gives an average error reduction of about 40% for UAS and about 32% for LAS, which is substantial especially given that the error reduction in the PM score (compared to the perfect morphology underlyling the GOLD scores) is only about

[12] Finnish had to be excluded because gold standard morphological annotation exists only for a small subset of the treebank.

27.5%. It is also worth pointing out that these improvements come at a very modest cost in computational efficiency, as the run times for the LEXSOFT model are on average only 15% higher than for the PIPELINE model, despite having a 40% larger beam size.[13] Interestingly, however, for all languages the LAS/UAS scores are actually higher for ORACLE than for GOLD, indicating that the LEXSOFT model has in its final beam dependency trees that are better than the 1-best trees predicted with perfect morphological input and suggesting that there is room for further improvement of the scoring model.

The final results obtained with joint prediction of morphology and syntax, external lexical constraints, and cluster features represent a new state of the art for syntactic dependency parsing for all five languages. For Czech, the best previous UAS on the standard train-test split of the PDT is 87.32, reported by Koo et al. (2010) with a parser using non-projective head automata and dual decomposition, while the best LAS is 78.82 LAS from Nilsson et al. (2006), using a greedy arc-eager transition-based system with pseudo-projective parsing. Our best results are 1.7 percentage points better for UAS (89.0) and almost 5 percentage points better for LAS (83.7).[14] For Finnish, the only previous results are from Haverinen et al. (2013), who achieve 81.01 LAS and 84.97 UAS with the graph-based parser of Bohnet (2010). We get substantial improvements with 83.1 LAS and 86.6 UAS. We also improve slightly over their best POS score, obtained with the HunPos tagger (Halácsy et al., 2007) together with the OMorFi analyzer (95.7 vs. 95.4). For German, the best previous results on the same train-test split are from Seeker and Kuhn (2012), using the graph-based parser of Bohnet (2010) in a pipeline architecture. With the same evaluation setup as in this paper, they achieve 91.50 LAS and 93.48 UAS – in the original paper, they only report results without punctuation – to be compared with 92.4 LAS and 94.1 UAS for our best model.[15] In addition, our POS score of 98.4 is the highest reported for a tagger trained only on the Tiger Treebank, outperforming the previous best from Bohnet and Nivre (2012) by 0.3 percentage points. The only previous results on Hungarian using the same version of the treebank are from Farkas et al. (2012), who report 87.2 LAS and 90.1 UAS for the graph-based parser of Bohnet (2010). Our best results improve labeled accuracy by 2.1 points (89.3 LAS) and unlabeled accuracy by 1.6 points (91.7 UAS), which is again quite substantial. For Russian, Boguslavsky et al. (2011) report 86.0 LAS and 90.0 UAS using the rule-based ETAP-3 parser with an added statistical model and joint morphological and syntactic disambiguation. The scores are not strictly comparable, because we use a more recent version of the SynTagRus treebank (May 2013 vs. April 2011), but our results nevertheless show substantial improvements, in particular for UAS (93.0) but also for LAS (88.0).

8 Concluding Remarks

We have presented the first system that performs full morphological disambiguation and labeled non-projective dependency parsing in a joint model, and we have demonstrated its usefulness for parsing richly inflected languages. A thorough empirical investigation of joint prediction models, rule-based lexical constraints, and distributional word clusters has shown substantial improvements in accuracy for five languages. In the future, we hope to conduct a detailed error analysis for all languages, which may give us more insight about the benefits of different components and hopefully pave the way for further improvements.

Acknowledgments

Work partly funded by the projects LM2010013 and LH12093 of the MEYS of the Czech Republic and the National Excellence Program of the State of Hungary (TÁMOP 4.2.4. A/2-11-1-2012-0001).

[13] LEXSOFT averages 0.132 ms per sentence on an Intel i7-3930K processor with 6 cores, against 0.112 ms for PIPELINE.

[14] It is worth noting that there are a number of more recent parsing results for Czech, but they all use a different test set (and often a different training set), usually from one of the CoNLL shared tasks in 2006 (Buchholz and Marsi, 2006), 2007 (Nivre et al., 2007) and 2009 (Hajič et al., 2009). For the 2009 data set, the best results are 83.73 LAS and 88.82 UAS from Bohnet and Nivre (2012), who use the SIMPLETAG model but with a beam size of 80. In our setup, we outperform this model by 0.5 points in both LAS and UAS.

[15] As in the case of Czech, there are many recent results for German based on the CoNLL 2009 data sets, but the previous best is with the SIMPLETAG model of Bohnet and Nivre (2012), which we outperform by 0.5/0.3 points in LAS/UAS.

References

Ju. Apresian, I. Boguslavsky, L. Iomdin, A. Lazursky, V. Sannikov, V. Sizov, and L. Tsinman. 2003. ETAP-3 linguistic processor: A full-fledged NLP implementation of the MTT. In *Proceedings of the First International Conference on Meaning-Text Theory*, pages 279–288.

Igor Boguslavsky, Svetlana Grigorieva, Nikolai Grigoriev, Leonid Kreidlin, and Nadezhda Frid. 2000. Dependency treebank for Russian: Concept, tools, types of information. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, pages 987–991.

Igor Boguslavsky, Ivan Chardin, Svetlana Grigorieva, Nikolai Grigoriev, Leonid Iomdin, Leonid Kreidlin, and Nadezhda Frid. 2002. Development of a dependency treebank for Russian and its possible applications in NLP. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pages 852–856.

Igor Boguslavsky, Leonid Iomdin, Victor Sizov, Leonid Tsinman, and Vadim Petrochenkov. 2011. Rule-based dependency parser refined by empirical and corpus statistics. In *Proceedings of the International Conference on Dependency Linguistics*, pages 318–327.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank: A three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Kluwer.

Bernd Bohnet and Jonas Kuhn. 2012. The best of both worlds – a graph-based completion model for transition-based parsers. In *Proceedings of the 13th Conference of the European Chpater of the Association for Computational Linguistics (EACL)*, pages 77–87.

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 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1455–1465.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 89–97.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. TIGER treebank. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT)*, pages 24–42.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 149–164.

Shay B. Cohen and Noah A. Smith. 2007. Joint morphological and syntactic disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 208–217.

Shay B. Cohen, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Exact inference for generative probabilistic non-projective dependency parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1245.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 112–119.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8.

Brooke Cowan and Michael Collins. 2005. Morphology and reranking for the statistical parsing of spanish. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 795–802.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Richárd Farkas, Veronika Vincze, and Helmut Schmid. 2012. Dependency parsing of hungarian: Baseline results and challenges. In *Proceedings of the 13th Conference of the European Chpater of the Association for Computational Linguistics (EACL)*, pages 55–65.

Andrea Gesmundo, James Henderson, Paola Merlo, and Ivan Titov. 2009. A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 37–42.

Yoav Goldberg and Michael Elhadad. 2013. Word segmentation, unknown-word resolution, and morphological agreement in a hebrew parsing system. *Computational Linguistics*, 39:121–160.

Yoav Goldberg and Reut Tsarfaty. 2008. A single generative model for joint morphological segmentation and

syntactic parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 371–379.

Jan Hajič and Barbora Hladká. 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and the 17th International Conference on Computational Linguistics (COLING)*, pages 483–490.

Jan Hajič, Barbora Vidova Hladka, Jarmila Panevová, Eva Hajičová, Petr Sgall, and Petr Pajas. 2001. Prague Dependency Treebank 1.0. LDC, 2001T10.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 1–18.

Jan Hajič. 2000. Morphological tagging: Data vs. dictionaries. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 94–101.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. HunPos – an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics: Companion Volume Proceedings of the Demo and Poster Sessions*, pages 209–212.

Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2011. Incremental joint pos tagging and dependency parsing in chinese. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1216–1224.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2013. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*.

Liang Huang and Kenji Sagae. 2010. Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1077–1086.

Leonid Iomdin and Viktor Sizov. 2008. Lexicographer's companion: A user-friendly software system for enlarging and updating high-profile computerized bilingual dictionaries. In *Lexicographic Tools and Techniques. MONDILEX First Open Workshop*, pages 42–54.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 595–603.

Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1288–1298.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Morgan and Claypool.

Marco Kuhlmann, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 673–682.

John Lee, Jason Naradowsky, and David A. Smith. 2011. A discriminative model for joint morphological disambiguation and dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 885–894.

Jens Nilsson, Joakim Nivre, and Johan Hall. 2006. Graph transformations in data-driven dependency parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 257–264.

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

Joakim Nivre, Marco Kuhlmann, and Johan Hall. 2009. An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 73–76.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 149–160.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together (ACL)*, pages 50–57.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*, pages 351–359.

Tommi A. Pirinen. 2011. Modularisation of finnish finite-state language description – towards wide collaboration in open source development of a morphological analyser. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 299–302.

Kenji Sagae and Andrew S. Gordon. 2009. Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT)*, pages 192–201.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 1263–1266.

Wolfgang Seeker and Jonas Kuhn. 2012. Making ellipses explicit in dependency conversion for a german treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 3132–3139.

Wolfgang Seeker, Bernd Bohnet, Lilja Øvrelid, and Jonas Kuhn. 2010. Informed ways of improving data-driven dependency parsing for german. In *Coling 2010: Posters*, pages 1122–1130.

Johanka Spoustová and Miroslav Spousta. 2012. A high-quality web corpus of czech. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 311–315.

Ivan Titov and James Henderson. 2007. A latent variable model for generative dependency parsing. In *Proceedings of the 10th International Conference on Parsing Technologies (IWPT)*, pages 144–155.

Viktor Trón, Péter Halácsy, Péter Rebrus, András Rung, Eszter Simon, and Péter Vajda. 2006. Morphdb.hu: Hungarian lexical database and morphological grammar. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1670–1673.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kuebler, Yannick Versley, Marie Candito, Jennifer Foster, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (spmrl) what, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 1–12.

Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing morphologicall rich languages: Introduction to the special issue. *Computational Linguistics*, 39:15–22.

Reut Tsarfaty. 2006. Integrated morphological and syntactic disambiguation for modern hebrew. In *Proceedings of the COLING/ACL 2006 Student Research Workshop*, pages 49–54.

Tamás Váradi. 2002. The hungarian national corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC)*, pages 385–389.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, pages 195–206.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 562–571.

Yue Zhang and Joakim Nivre. 2011. Transition-based parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yue Zhang and Joakim Nivre. 2012. Analyzing the effect of global learning and beam-search on transition-based dependency parsing. In *Proceedings of COLING 2012: Posters*, pages 1391–1400.