Dependency Parsing of Turkish

Gülşen Eryiğit*
Istanbul Technical University

Joakim Nivre** †
Växjö University, Uppsala University

Kemal Oflazer‡
Sabancı University

The suitability of different parsing methods for different languages is an important topic in syntactic parsing. Especially lesser-studied languages, typologically different from the languages for which methods have originally been developed, pose interesting challenges in this respect. This article presents an investigation of data-driven dependency parsing of Turkish, an agglutinative, free constituent order language that can be seen as the representative of a wider class of languages of similar type. Our investigations show that morphological structure plays an essential role in finding syntactic relations in such a language. In particular, we show that employing sublexical units called inflectional groups*, rather than word forms, as the basic parsing units improves parsing accuracy. We test our claim on two different parsing methods, one based on a probabilistic model with beam search and the other based on discriminative classifiers and a deterministic parsing strategy, and show that the usefulness of sublexical units holds regardless of the parsing method. We examine the impact of morphological and lexical information in detail and show that, properly used, this kind of information can improve parsing accuracy substantially. Applying the techniques presented in this article, we achieve the highest reported accuracy for parsing the Turkish Treebank.*

1. Introduction

Robust syntactic parsing of natural language is an area in which we have seen tremendous development during the last 10 to 15 years, mainly on the basis of data-driven methods but sometimes in combination with grammar-based approaches. Despite this, most of the approaches in this field have only been tested on a relatively small set of languages, mostly English but to some extent also languages like Chinese, Czech, Japanese, and German.

* Department of Computer Engineering, Istanbul Technical University, 34469 Istanbul, Turkey. E-mail: gulsen.cebiroglu@itu.edu.tr.
** School of Mathematics and Systems Engineering, Växjö University, 35260 Växjö, Sweden. E-mail: joakim.nivre@msi.vxu.se.
† Department of Linguistics and Philology, Uppsala University, Box 635, 75126 Uppsala, Sweden.
‡ Faculty of Engineering and Natural Sciences, Sabancı University, 34956 Istanbul, Turkey. E-mail: oflazer@sabanciuniv.edu.

Submission received: 5 October 2006; revised submission received: 3 April 2007; accepted for publication: 16 May 2007.

An important issue in this context is to what extent our models and algorithms are tailored to properties of specific languages or language groups. This issue is especially pertinent for data-driven approaches, where one of the claimed advantages is portability to new languages. The results so far mainly come from studies where a parser originally developed for English, such as the Collins parser (Collins 1997, 1999), is applied to a new language, which often leads to a significant decrease in the measured accuracy (Collins et al. 1999; Bikel and Chiang 2000; Dubey and Keller 2003; Levy and Manning 2003; Corazza et al. 2004). However, it is often quite difficult to tease apart the influence of different features of the parsing methodology in the observed degradation of performance.

A related issue concerns the suitability of different kinds of syntactic representation for different types of languages. Whereas most of the work on English has been based on constituency-based representations, partly influenced by the availability of data resources such as the Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1993), it has been argued that free constituent order languages can be analyzed more adequately using dependency-based representations, which is also the kind of annotation found, for example, in the Prague Dependency Treebank of Czech (Hajič et al. 2001). Recently, dependency-based parsing has been applied to 13 different languages in the shared task of the 2006 Conference on Computational Natural Language Learning (CoNLL) (Buchholz and Marsi 2006).

In this article, we focus on dependency-based parsing of Turkish, a language that is characterized by rich agglutinative morphology, free constituent order, and predominantly head-final syntactic constructions. Thus, Turkish can be viewed as the representative of a class of languages that are very different from English and most other languages that have been studied in the parsing literature. Using data from the recently released Turkish Treebank (Oflazer et al. 2003), we investigate the impact of different design choices in developing data-driven parsers. There are essentially three sets of issues that are addressed in these experiments.

- The first set includes issues relating to the treatment of *morphology* in syntactic parsing, which becomes crucial when dealing with languages where the most important clues to syntactic functions are often found in the morphology rather than in word order patterns. Thus, for Turkish, it has previously been shown that parsing accuracy can be improved by taking morphologically defined units rather than word forms as the basic units of syntactic structure (Eryiğit and Oflazer 2006). In this article, we corroborate this claim showing that it holds in both approaches we explore. We also study the impact of different morphological feature representations on parsing accuracy.
- The second set of issues concerns *lexicalization*, a topic that has been very prominent in the parsing literature lately. Whereas the best performing parsers for English all make use of lexical information, the real benefits of lexicalization for English as well as other languages remains controversial (Dubey and Keller, 2003; Klein and Manning 2003; Arun and Keller 2005).
- The third set concerns the basic *parsing methodology*, including both parsing algorithms and learning algorithms. We first introduce a statistical parser using a conditional probabilistic model which is very sensitive to the selected representational features and thus clearly exposes the ones

with crucial importance for parsing Turkish. We then implement our models on a deterministic classifier-based parser using discriminative learning, which is one of the best performing dependency parsers evaluated on a wide range of different languages.

Additionally we address the following issues:

- We investigate learning curves and provide an error analysis for the best performing parser.
- For most of our experiments we use as input the gold-standard tags from the treebank. However, in our last experiments we evaluate the impact of automatic statistical morphological disambiguation on the performance of our best performing parser.

The rest of the article is structured as follows. Section 2 gives a very brief introduction to Turkish morphology and syntax and discusses the representation of morphological information and syntactic dependency relations in the Turkish Treebank. Section 3 is devoted to methodological issues, in particular the data sets and evaluation metrics used in experiments. The following two sections present two different dependency parsers trained and evaluated on the Turkish Treebank: a probabilistic parser (Section 4) and a classifier-based parser (Section 5). Section 6 investigates the impact of lexicalization and morphological information on the two parsers, and Section 7 examines their learning curves. Section 8 presents an error analysis for the best performing parser, and Section 9 analyzes the degradation in parsing performance when using automatic morphological disambiguation. Section 10 discusses related work, and Section 11 summarizes the main conclusions from our study.

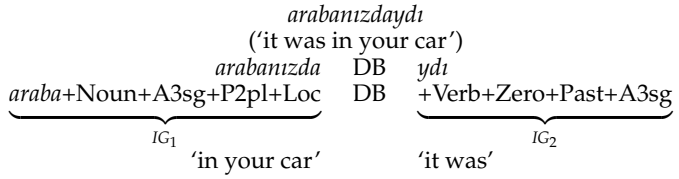
2. Turkish: Morphology and Dependency Relations

Turkish displays rather different characteristics compared to the more well-studied languages in the parsing literature. Most of these characteristics are also found in many agglutinative languages such as Basque, Estonian, Finnish, Hungarian, Japanese, and Korean.[1] Turkish is a flexible constituent order language. Even though in written texts the constituent order predominantly conforms to the SOV order, constituents may freely change their position depending on the requirements of the discourse context (Erguvanlı 1979; Hoffman 1994). However, from a dependency structure point of view, Turkish is predominantly (but not exclusively) head final.

Turkish has a very rich agglutinative morphological structure. Nouns can give rise to about 100 inflected forms and verbs to many more. Furthermore, Turkish words may be formed through very productive derivations, increasing substantially the number of possible word forms that can be generated from a root word. It is not uncommon to find up to four or five derivations in a single word. Previous work on Turkish (Hakkani-Tür, Oflazer, and Tür 2002; Oflazer 2003; Oflazer et al. 2003; Eryiğit and Oflazer 2006) has represented the morphological structure of Turkish words by splitting them into inflectional groups (IGs). The root and derivational elements of a word are represented

1 We, however, do not necessarily suggest that the morphological sublexical representation that we use for Turkish later in this article is applicable to these languages.

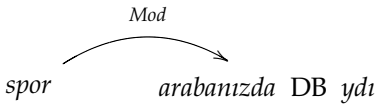
by different IGs, separated from each other by derivational boundaries (DB). Each IG is then annotated with its own part of speech and any inflectional features as illustrated in the following example:[2]



In this example, the root of the word *arabanızdaydı* is *araba* ('car') and its part of speech is noun. From this, a verb is derived in a separate IG. So, the word is composed of two IGs where the first one, *arabanızda* ('in your car'), is a noun in locative case and in second plural possessive form, and the second one is a verbal derivation from this noun in past tense and third person singular form.

2.1 Dependency Relations in Turkish

Because most syntactic information is mediated by morphology, it is not sufficient for the parser to only find dependency relations between orthographic words;[3] the correct IGs involved in the relations should also be identified. We can motivate this with the following very simple example: In the phrase *spor arabanızdaydı* ('it was in your sports car'), the adjective *spor* ('sports') should be connected to the first IG of the second word. It is the word *araba* ('car') which is modified by the adjective, not the derived verb form *arabanızdaydı* ('it was in your car'). So a parser should not just say that the first word is a dependent of the second but also state that the syntactic relation is between the last IG of the first word and the first IG of the second word, as shown:



In Figure 1 we see a complete dependency tree for a Turkish sentence laid on top of the words segmented along IG boundaries. The rounded rectangles show the words and IGs within words are marked with dashed rounded rectangles. The first thing to note in this figure is that the dependency links always emanate from the last IG of a word, because that IG determines the role of that word as a dependent. The dependency links land on one of the IGs of a (head) word (almost always to the right). The non-final IGs (e.g., the first IG of the word *okuldaki* in Figure 1) may only have incoming dependency

2 +A3sg = 3sg number agreement, +P2pl = 2pl possessive agreement, +Loc = Locative Case.

3 For the same reason, Bozsahin (2002) uses morphemes as sublexical constituents in a CCG framework. Because the lexicon was organized in terms of morphemes each with its own CCG functor, the grammar had to account for both the morphotactics and the syntax at the same time.

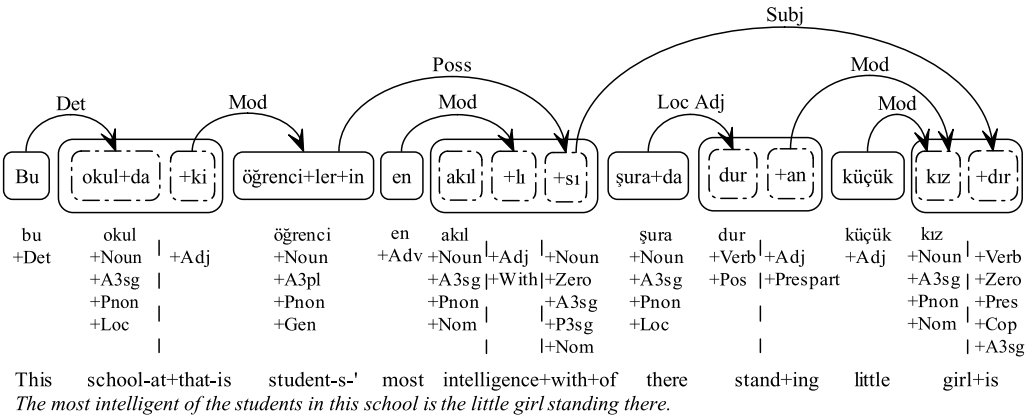


Figure 1
Dependency links in an example Turkish sentence.
'+'s indicate morpheme boundaries. The rounded rectangles show words, and IGs within words that have more than one IG are indicated by the dashed rounded rectangles. The inflectional features of each IG as produced by the morphological analyzer are listed below the IG.

links and are assumed to be morphologically linked to the next IG to the right (but we do not explicitly show these links).[4]

The noun phrase formed by the three words *öğrencilerin en akıllısı* in this example highlights the importance of the IG-based representation of syntactic relations. Here in the word *akıllısı*, we have three IGs: The first contains the singular noun *akıl* ('intelligence'), the second IG indicates the derivation into an adjective *akıllı* ('intelligence-with' → 'intelligent'). The preceding word *en* ('most'), an intensifier adverb, is linked to this IG as a modifier (thus forming 'most intelligent'). The third IG indicates another derivation into a noun ('a singular entity that is most intelligent'). This last IG is the head of a dependency link emanating from the word *öğrencilerin* with genitive case-marking ('of the students' or 'students' ') which acts as the possessor of the last noun IG of the third word *akıllısı*. Finally, this word is the subject of the verb IG of the last word, through its last IG.

2.2 The Turkish Treebank

We have used the Turkish Treebank (Oflazer et al. 2003), created by the Middle East Technical University and Sabancı University, in the experiments we report in this article. The Turkish Treebank is based on a small subset of the Metu Turkish Corpus (`www.ii.metu.edu.tr/~corpus/corpus.html`), a balanced collection of post-1990 Turkish text from 10 genres. The version that has been used in this article is the version used in the CoNLL-X shared task publicly available from `www.ii.metu.edu.tr/~corpus/treebank.html`.

This treebank comprises 5,635 sentences in which words are represented with IG-based gold-standard morphological representation and dependency links between IGs.

4 It is worth pointing out that arrows in this representation point from dependents to heads, because representations with arrows in the opposite direction also exist in the literature.

The average number of IGs per word is 1.26 in running text, but the figure is higher for open class words and 1 for high frequency function words which do not inflect. Of all the dependencies in the treebank, 95% are head-final[5] and 97.5% are projective.[6]

Even though the number of sentences in the Turkish Treebank is in the same range as for many other available treebanks for languages such as Danish (Kromann 2003), Swedish (Nivre, Nilsson, and Hall 2006), and Bulgarian (Simov, Popova, and Osenova 2002), the number of words is considerably smaller (54K as opposed to 70–100K for the other treebanks). This corresponds to a relatively short average sentence length in the treebank of about 8.6 words, which is mainly due to the richness of the morphological structure, because often one word in Turkish may correspond to a whole sentence in another language.

3. Dependency Parsing of Turkish

In the following sections, we investigate different approaches to dependency parsing of Turkish and show that using parsing units smaller than words improves the parsing accuracy. We start by describing our evaluation metrics and the data sets used, and continue by presenting our baseline parsers: two naive parsers, which link a dependent to an IG in the next word, and one rule-based parser. We then present our data-driven parsers in the subsequent sections: a statistical parser using a conditional probabilistic model (from now on referred to as the **probabilistic parser**) in Section 4 and a deterministic classifier-based parser using discriminative learning (from now on referred to as the **classifier-based parser**) in Section 5.

3.1 Data Sets and Evaluation Metrics

Our experiments are carried out on the entire treebank and all our results are reported for this data set. We use ten-fold cross-validation for the evaluation of the parsers, except for the baseline parsers which do not need to be trained. We divide the treebank data into ten equal parts and in each iteration use nine parts as training data and test the parser on the remaining part.

We report the results as mean scores of the ten-fold cross-validation, with standard error. The main evaluation metrics that we use are the **unlabeled attachment score** (AS_U) and **labeled attachment score** (AS_L), namely, the proportion of IGs that are attached to the correct head (with the correct label for AS_L). A correct attachment is one in which the dependent IG (the last IG in the dependent word) is *not only* attached to the correct head word *but also to the correct IG within the head word*. Where relevant, we also report the (unlabeled) word-to-word score (WW_U), which only measures whether a dependent word is connected to (some IG in) the correct head word. It should be clear from the discussion in Section 2.1 and from Figure 1 that the IG-to-IG evaluation is the right one to use for Turkish even though it is more stringent than word-to-word evaluation. Dependency links emanating from punctuation are excluded in all

5 Half of the head-initial dependencies are actually not real head-initial structures; these are caused by some enclitics (addressed in detail in the following sections) which can easily be recovered with some predefined rules.

6 A dependency between a dependent *i* and a head *j* is projective if and only if all the words or IGs that occur between *i* and *j* in the linear order of the sentence are dominated by *j*. A dependency analysis with only projective dependencies corresponds to a constituent analysis with only continuous constituents.

evaluation scores. Non-final IGs of a word are assumed to link to the next IG within the word, but these links, referred to as **InnerWord** links, are not considered as dependency relations and are excluded in evaluation scoring.

3.2 Baseline Parsers

We implemented three baseline parsers to assess the performance of our probabilistic and classifier-based parsers. The first baseline parser attaches each word (from the last IG) to the first IG of the next word while the second parser attaches each word to the final IG of the next word. Obviously these two baseline parsers behave the same when the head word has only one IG. The final punctuation of each sentence is assumed to be the root of the sentence and it is not connected to any head. The first two lines of Table 1 give the unlabeled attachment scores of these parsers. We observe that attaching the link to the first IG instead of the last one gives better results.

The third baseline parser is a rule-based parser that uses a modified version of the deterministic parsing algorithm by Nivre (2006). This parsing algorithm, which will be explained in detail in Section 5, is a linear-time algorithm that derives a dependency graph in one left-to-right pass over the input, using a stack to store partially processed tokens and a list to store remaining input tokens in a way similar to a shift-reduce parser. In the rule-based baseline parser, the next parsing action is determined according to 31 predefined hand-written rules (Eryiğit 2006; Eryiğit, Adalı, and Oflazer 2006). The rules determine whether or not to connect the units (words or IGs) on top of the stack and at the head of the input list (regardless of dependency labels). It can be seen that the rule-based parser provides an improvement of about 15 percentage points compared to the relatively naive simpler baseline parsers which cannot recover head-initial dependencies.

4. Probabilistic Dependency Parser

A well-studied approach to dependency parsing is a statistical approach where the parser takes a morphologically tagged and disambiguated sentence as input, and outputs the most probable dependency tree by using probabilities induced from the training data. Such an approach comprises three components:

1. A parsing algorithm for building the dependency analyses (Eisner 1996; Sekine, Uchimoto, and Isahara 2000)
2. A conditional probability model to score the analyses (Collins 1996)

Table 1
Unlabeled attachment scores and unlabeled word-to-word scores for the baseline parsers.

Parsing Model	AS_U	WW_U
Attach-to-next (first IG)	56.0	63.3
Attach-to-next (last IG)	54.1	63.3
Rule-based	70.5	79.3

3. Maximum likelihood estimation to make inferences about the underlying probability models (Collins 1996; Chung and Rim 2004)

4.1 Methodology

The aim of our probabilistic model is to assign a probability to each candidate dependency link by using the frequencies of similar dependencies computed from a training set. The aim of the parsing algorithm is then to explore the search space in order to find the most probable dependency tree. This can be formulated with Equation (1) where S is a sequence of n units (words or IGs) and T ranges over possible dependency trees consisting of dependency links $dep(u_i, u_{H(i)})$, with $u_{H(i)}$ denoting the head unit to which the dependent unit u_i is linked and the probability of a given tree is the product of the dependency links that it comprises.

$$T^* = \underset{T}{\operatorname{argmax}}\, P(T|S) = \underset{T}{\operatorname{argmax}} \prod_{i=1}^{n-1} P(dep\,(u_i, u_{\mathcal{H}(i)}) \,|\, S) \tag{1}$$

The observation that 95% of the dependencies in the Turkish treebank are head-final dependencies motivated us to employ the backward beam search dependency parsing algorithm by Sekine, Uchimoto, and Isahara (2000) (originally designed for Japanese, another head-final language), adapted to our morphological representation with IGs. This algorithm parses a sentence starting from the end moving towards the beginning, trying at each step to link the dependents to a unit to the right. It uses a beam which keeps track of the most probable dependency structures for the partially processed sentence. However, in order to handle head-initial dependencies, it employs three predefined lexicalized rules[7] (also used in our rule-based baseline parser). For every new word, the parser starts by checking if any of the rules apply. If so, it constructs a right-to-left link where $H(i) < i$ and directly assigns 1.0 as the dependency probability ($P(dep\,(u_i, u_{\mathcal{H}(i)}) \,|\, S) = 1.0$). If none of the rules apply, it instead uses probabilities for head-final dependencies.

For the probability model, we adopt the approach by Chung and Rim (2004), which itself is a modified version of the statistical model used in Collins (1996).[8] In this model in Equation (2), the probability of a dependency link $P(dep\,(u_i, u_{\mathcal{H}(i)}) \,|\, S)$ linking u_i to a head $u_{H(i)}$ is approximated with the product of two probabilities:

$$\begin{aligned} P(dep\,(u_i, u_{\mathcal{H}(i)}) \,|\, S) \approx\; & P(link(u_i, u_{\mathcal{H}(i)}) \,|\, \Phi_i\ \Phi_{\mathcal{H}(i)}) \\ & P(u_i \text{ links to some head } dist(i, H(i)) \text{ away} \,|\, \Phi_i) \end{aligned} \tag{2}$$

7 The rules check for enclitics such as *de*, *ki*, *mi*, written on the right side of and separately from the word they attach to, for the verb *değil*, which gives a negative meaning to the word coming before it and for nominals which do not have any verbs on their right side.

8 The statistical model in Collins (1996) is actually used in a phrase-structure-based parsing approach, but it uses the same idea of computing probabilities between dependents and head units. We also tried to employ the statistical model of Collins, where the distance measure $\Delta_{i,\mathcal{H}(i)}$ is included in the link probability formula ($P(dep\,(u_i, u_{\mathcal{H}(i)}) \,|\, S) \approx P(link(u_i, u_{\mathcal{H}(i)}) \,|\, \Delta_{i,\mathcal{H}(i)})$) , but we obtained worse results with this.

In this equation,

- $P(link(u_i, u_{\mathcal{H}(i)}) \mid \Phi_i\ \Phi_{\mathcal{H}(i)})$ is the probability of seeing the same dependency within a similar context where Φ_i represents the context around the dependent u_i and $\Phi_{\mathcal{H}(i)}$ represents the context around the head $u_{H(i)}$, and
- $P(u_i$ links to some head $dist(i, H(i))$ away $\mid \Phi_i)$ is the probability of seeing the dependent linking to some head a distance $dist(i, H(i))$ away, in the context Φ_i.

In all of the following models, $dist(i, H(i))$ is taken as the number of actual word boundaries between the dependent and the head unit regardless of whether full words or IGs were used as units of parsing.[9]

To alleviate the data sparseness, we use the interpolation of other estimates while calculating the probabilities in Equation (2).[10] We use a strategy similar to Collins (1996) and we interpolate with estimates based on less context:

$$P(x|y) \approx \lambda \cdot P_1(x|y) + (1 - \lambda) \cdot P_2(x) \tag{3}$$

$$\text{where} \quad \lambda = \delta/(\delta + 1) \ \text{ and } \ \delta \text{ is the count of the x occurrences}$$

During the actual runs, the smoothed probability $P(link(u_i, u_{\mathcal{H}(i)}) \mid \Phi_i\ \Phi_{\mathcal{H}(i)})$ is estimated by interpolating two unsmoothed empirical estimates extracted from the treebank: $P_1(link(u_i, u_{\mathcal{H}(i)}) \mid \Phi_i\ \Phi_{\mathcal{H}(i)})$ and $P_2(link(u_i, u_{\mathcal{H}(i)}))$. A similar approach was employed for $P(u_i$ links to some head $dist(i, H(i))$ away $\mid \Phi_i)$ and it is estimated by interpolating $P_1(u_i$ links to some head $dist(i, H(i))$ away $\mid \Phi_i)$ and $P_2(u_i$ links to some head $dist(i, H(i))$ away). If even after interpolation, the probability is 0, then a very small value is used. Further, distances larger than a certain threshold value were assigned the same probability, as explained later.

4.2 The Choice of Parsing Units

In the probabilistic dependency parsing experiments, we experimented with three different ways of choosing and representing the units for parsing:[11]

1. **Word-based model #1**: In this model, the units of parsing are the actual words and each word is represented by a combination of the representations of *all* the IGs that make it up. Note that although all IGs are used in representing a word, not all the information provided by an IG has to be used, as we will see shortly. This representation, however, raises the following question: If we use the words as the parsing units and

9 We also tried other distance functions, for example, the number of IGs between dependent and head units, but this choice fared better than the alternatives.

10 We tried many other different interpolation and backoff models where we tried to remove the neighbors one by one or the inflectional features. But we obtained the best results with a two-level interpolation by removing the contextual information all at once.

11 Clarifying examples of these representations will be provided in the immediately following section.

find the dependencies between these, how can we translate these to the dependencies between the IGs, since our goal is to find dependencies between IGs? The selection of the IG of the dependent word is an easy decision, as it is the last IG in the word. The selection of the head IG is obviously more difficult. Because such a word-based model will not provide much information about the underlying IGs structure, we will have to make some assumptions about the head IG. The observation that 85.6% of the dependency links in the treebank land on the first (and possibly the only) IG of the head word and the fact that our first baseline model (attaching to the first IG) gives better performance than our second baseline model (attaching to the last IG), suggest that after identifying the correct word, choosing the first IG as the head IG may be a reasonable heuristic. Another approach to determining the correct IG in the head word could be to develop a post-processor which selects this IG using additional rules. Such a post-processor could be worth developing if the WW_U accuracy obtained with this model proves to be higher than all of the other models, that is, if this is the best way of finding the correct dependencies between words without considering which IGs are connected. However, as we will see in Section 4.4, this model does not give the best WW_U.

2. **Word-based model #2**: This model is just like the previous model but we represent a word using its final IGs rather than the concatenation of all their IGs when *it is used as a dependent*. The representation is the same as in Word-based model #1 when the word is a head. This results in a dynamic selection of the representation during parsing as the representation of a word will be determined according to its role at that moment. The representation of the neighboring units in context will again be selected with respect to the word in question: any context unit on the left will be represented with its dependent representation (just the last IG) and any neighbor on the right will be represented with its representation as a head. The selection of the IG in the head word is the same as in the first model.

3. **IG-based model**: In this model, we use IGs as units in parsing. We split the IG-based representation of each word and reindex these IGs in order to use them as single units in parsing. Figure 2 shows this transfer to the IG-based model. We still, however, need to know which IGs are word-final as they will be the dependent IGs (shown in the figure by asterisks). The contextual elements that are used in this model are the IGs to the left (starting with the last IG of the preceding word) and to the right of the dependent and the head IG.

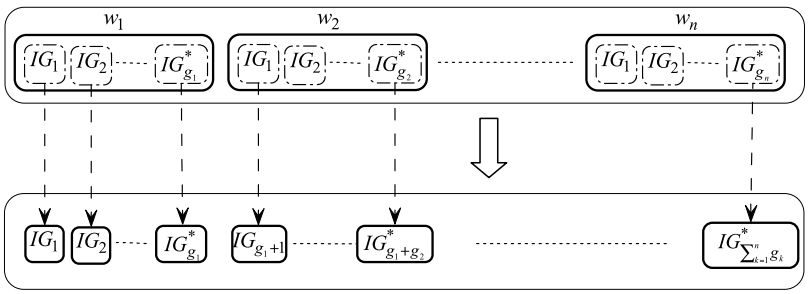


Figure 2
Mapping from word-based to IG-based representation of a sentence.

4.3 Reduced Dynamic Representations for IGs

In all three models, it is certainly possible to use all the information supplied by the full morphological analysis in representing the IGs.[12] This includes the root words themselves, major and minor parts of speech,[13] number and person agreement markers, possessive agreement markers, case markers, tense, aspect, mood markers, and other miscellaneous inflectional and semantic markers especially for derivations. Not all of these features may be relevant to the parsing task, and further, different features may be relevant depending on whether the IG is being used as a dependent or a head. Also, in order to alleviate the data sparseness problem that may result from the relatively modest size of the treebank, an "unlexicalized" representation that does not contain the root word needs to be considered so that statistics from IGs that are otherwise the same except for the root word (if any) can be conflated.[14] After some preliminary experimentation, we decided that a reduced representation for IGs that is dynamically selected depending on head or dependent status would give us the best performance. We explain the representation of the IGs and the parameters that we used in the three models.

- When used as a dependent (or part of a dependent word in models 1 and 2) during parsing;
  - Nominal IGs (nouns, pronouns, and other derived forms that inflect with the same paradigm as nouns, including infinitives, past and future participles) are represented only with the case marker, because that essentially determines the syntactic function of that IG as a dependent, and only nominals have cases.
  - Any other IG is just represented with its minor part of speech.
- When used as a head (or part of a head word in models 1 and 2);
  - Nominal IGs and adjective IGs with participle minor part of speech[15] are represented with the minor part of speech and the possessive agreement marker.
  - Any other IG is just represented with its minor part of speech.

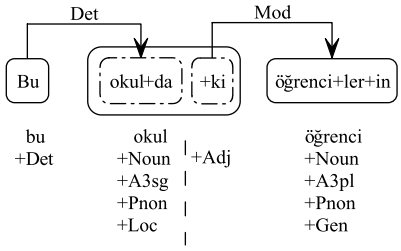
Figures 3–5 show, for the first three words in Figure 1, the unlexicalized reduced representations that are used in the three models when units are used as dependents and heads during parsing.

12 See Figure 1 for a sample of such information.

13 A minor part-of-speech category is available for some major part-of-speech categories: pronouns are further divided into personal pronouns, demonstrative pronouns, interrogative pronouns, and so on. The minor part-of-speech category always implies the major part of speech. For derived IGs, the minor part of speech mostly indicates a finer syntactic or semantic characterization of the derived word. When no minor part of speech is available the major part of speech is used.

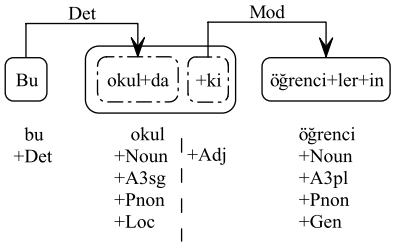
14 Remember that only the first IG in a word has the root word.

15 These are modifiers derived from verbs. They have adjective as their major part of speech and past/future participle as their minor part of speech. They are the only types of IGs that have possessive agreement markers other than nominals.



dependent	<+Det>	<+Loc+Adj>	<+Gen>
head	<+Det>	<+Noun+Pnon+Adj>	<+Noun+Pnon>

Figure 3
Reduced IG representation for Word-based model #1.



dependent	<+Det>	<+Adj>	<+Gen>
head	<+Det>	<+Noun+Pnon+Adj>	<+Noun+Pnon>

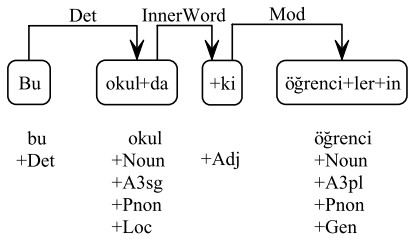
Figure 4
Reduced IG representation for Word-based model #2.

4.4 Experimental Results

In this section, we first evaluate the performance of the models described in Section 4.2. We then investigate the impact of different choices of morphological features on the best performing IG-based model. In addition to the parsing model, the parser is given the following parameters:

- the number of left and right neighbors of the dependent (D_l, D_r) to define the dependent context Φ_i,[16]
- the number of left and right neighbors of the head (H_l, H_r) to define the head context $\Phi_{H(i)}$,
- the size of the beam (**beamsize**), and

16 In terms of parsing units, the number of words for word-based models and the number of IGs for IG-based models.



dependent	<+Det>	<+Loc>	<+Adj>	<+Gen>
head	<+Det>	<+Noun+Pnon>	<+Adj>	<+Noun+Pnon>

Figure 5
Reduced IG representation for IG-based model.

- the distance **threshold** value beyond which $P(u_i$ links to some head $dist(i, H(i))$ away $\mid \Phi_i)$ is assigned the same probability.

Table 2 gives the AS_U scores for the word-based and IG-based models for the best combinations of contexts used for each case. We also provide WW_U scores for comparison, but again stress that the main evaluation criterion is the AS_U score. For all three models, the beamsize value is selected as 3 and distance threshold is selected as 6.[17] It can be seen that the performance of the word-based models is lower than our rule-based baseline parser (Table 1) with AS_U = 70.5, even though it is better than the first two rather naive baselines. On the other hand, the IG-based model outperforms all of the baseline parsers and word-based models. It should also be noted that the IG-based model improves not only the AS_U accuracy but also the word-to-word accuracy compared, to the word-based models. Thus, the IG-based model not only helps to recover the relations between correct IGs but also to find the correct head word.

In Table 3, we also present results from experiments employing different representations for the IGs. A more detailed investigation about the use of limited lexicalization and inflectional features will be presented later in Section 6. Here, we will see what would have happened if we had used alternative reduced IG representations compared to the representation described earlier, which is used in the best performing IG-based model.

Table 3 gives the results for each change to the representational model. One can see that none of these representational changes improves the performance of the best performing model. Only employing major part-of-speech tags (#1) actually comes close, and the difference is not statistically significant. Lexicalization of the model results in a drastic decrease in performance: Using the surface form (#6) gives somewhat better

17 As stated earlier in Section 4.1, our distance function is calculated according to the word boundaries between the dependent and the head units. In the treebank, 95% of the dependency links link to a word that is less than six words away. Thus all the distances larger than or equal to six are conflated into the same small probability.

Table 2
Unlabeled attachment scores and unlabeled word-to-word scores for the probabilistic parser.

Parsing Model (parameters)		AS_U	WW_U
Word-based model #1	*(D_l=1, D_r=1, H_l=1, H_r=1)*	68.1±0.4	77.1±0.7
Word-based model #2	*(D_l=1, D_r=1, H_l=1, H_r=1)*	68.3±0.3	77.6±0.5
IG-based model	*(D_l=1, D_r=1, H_l=0, H_r=1)*	72.1±0.3	79.0±0.7

results than using root information (#5). Also, dynamic selection of tags seems to help performance (#3) but using all available inflectional information performs significantly worse possibly due to data sparseness.

5. Classifier-Based Dependency Parser

Our second data-driven parser is based on a parsing strategy that has achieved a high parsing accuracy across a variety of different languages (Nivre et al. 2006, 2007). This strategy consists of the combination of the following three techniques:

1. Deterministic parsing algorithms for building dependency graphs (Kudo and Matsumoto 2002; Nivre 2003; Yamada and Matsumoto 2003)

Table 3
Unlabeled attachment scores for different choices for morphological features.

#	**Model**	AS_U
	IG-based model (D_l=1, D_r=1, H_l=0, H_r=1)	72.1±0.3
1	Using major part of speech instead of minor part of speech	71.2±0.2
2	Using only minor part of speech and no other inflectional features	68.3±0.2
3	Using minor part of speech for all types of IGs together with case and possessive markers for nominals and possessive marker for adjectives (but no dynamic selection)	71.0±0.3
4	Using all inflectional features in addition to minor part of speech	46.5±0.4
5	Adding root information to the best performing IG-based model	53.7±0.2
6	Adding surface form information to the best performing IG-based model	54.4±0.2

2. History-based models for predicting the next parser action (Black et al. 1992; Magerman 1995; Ratnaparkhi 1997; Collins 1999)
3. Discriminative classifiers to map histories to parser actions (Kudo and Matsumoto 2002; Yamada and Matsumoto 2003; Nivre, Hall, and Nilsson 2004)

A system of this kind employs no grammar but relies completely on inductive learning from treebank data for the analysis of new sentences, and on deterministic parsing for disambiguation. This combination of methods guarantees that the parser is robust, never failing to produce an analysis for an input sentence, and efficient, typically deriving this analysis in time that is linear in the length of the sentence.

In the following sections, we will first present the parsing methodology and then results that show that the IG-based model again outperforms the word-based model. We will then explore how we can further improve the accuracy by exploiting the advantages of this parser. All experiments are performed using the freely available implementation MaltParser.[18]

5.1 Methodology

For the experiments in this article, we use a variant of the parsing algorithm proposed by Nivre (2003, 2006), a linear-time algorithm that derives a labeled dependency graph in one left-to-right pass over the input, using a stack to store partially processed tokens and a list to store remaining input tokens. However, in contrast to the original arc-eager parsing strategy, we use an arc-standard bottom-up algorithm, as described in Nivre (2004). Like many algorithms used for dependency parsing, this algorithm is restricted to projective dependency graphs.

The parser uses two elementary data structures, a stack σ of partially analyzed tokens and an input list τ of remaining input tokens. The parser is initialized with an empty stack and with all the tokens of a sentence in the input list; it terminates as soon as the input list is empty. In the following, we use subscripted indices, starting from 0, to refer to particular tokens in σ and τ. Thus, σ_0 is the token on top of the stack σ (the **top token**) and τ_0 is the first token in the input list τ (the **next token**); σ_0 and τ_0 are collectively referred to as the **target tokens**, because they are the tokens considered as candidates for a dependency relation by the parsing algorithm.

There are three different parsing actions, or transitions, that can be performed in any non-terminal configuration of the parser:

- **Shift**: Push the next token onto the stack.
- **Left-Arc**$_r$: Add a dependency arc from the next token to the top token, labeled r, then pop the stack.
- **Right-Arc**$_r$: Add a dependency arc from the top token to the next token, labeled r, then replace the next token by the top token at the head of the input list.

18 http://w3.msi.vxu.se/users/nivre/research/MaltParser.html.

In order to perform deterministic parsing in linear time, we need to be able to predict the correct parsing action (including the choice of a dependency type r for **Left-Arc**$_r$ and **Right-Arc**$_r$) at any point during the parsing of a sentence. This is what we use a history-based classifier for.

The features of the history-based model can be defined in terms of different linguistic features of tokens, in particular the target tokens. In addition to the target tokens, features can be based on neighboring tokens, both on the stack and in the remaining input, as well as dependents or heads of these tokens in the partially built dependency graph. The linguistic attributes available for a given token are the following:

- Lexical form (root) (LEX)
- Part-of-speech category (POS)
- Inflectional features (INF)
- Dependency type to the head if available (DEP)

To predict parser actions from histories, represented as feature vectors, we use support vector machines (SVMs), which combine the maximum margin strategy introduced by Vapnik (1995) with the use of kernel functions to map the original feature space to a higher-dimensional space. This type of classifier has been used successfully in deterministic parsing by Kudo and Matsumoto (2002), Yamada and Matsumoto (2003), and Sagae and Lavie (2005), among others. To be more specific, we use the LIBSVM library for SVM learning (Chang and Lin 2001), with a polynomial kernel of degree 2, with binarization of symbolic features, and with the one-versus-one strategy for multiclass classification.[19]

This approach has some advantages over the probabilistic parser, in that

- it can process both left-to-right and right-to-left dependencies due to its parsing algorithm,
- it assigns dependency labels simultaneously with dependencies and can use these as features in the history-based model, and
- it does not necessarily require expert knowledge about the choice of linguistically relevant features to use in the representations because SVM training involves implicit feature selection.

However, we still exclude sentences with non-projective dependencies during training.[20] Because the classifier-based parser not only builds dependency structures but also assigns dependency labels, we give AS_L scores as well as AS_U scores.

19 Experiments have also been performed using memory-based learning (Daelemans and Bosch 2005). They were found to give lower parsing accuracy.

20 Because the frequency of non-projective dependencies in the Turkish Treebank is not high enough to learn such dependencies and mostly due to the unconnected punctuations with which we are dealing by adding an extra dependency label, we did not observe any improvement when applying the pseudo-projective processing of Nivre and Nilsson (2005), which is reported to improve accuracy for other languages.

5.2 Experimental Results

In this section, our first aim is to confirm the claim that using IGs as the units in parsing improves performance. For this purpose, we start by using models similar to those described in the previous section. We use an unlexicalized feature model where the parser uses only the minor POS and the DEP of tokens and compare the results with the probabilistic parser. We then show in the second part how we can improve accuracy by exploiting the morphological structure of Turkish and taking advantage of the special features of this parser.

5.2.1 Comparison with the Probabilistic Parser. In order to compare with the results of the previous section, we adopt the same strategy that we used earlier in order to present inflectional groups. We employ two representation models:

- **Word-based model**, where each word is represented by the concatenation of its IGs,
- **IG-based model**, where the units are inflectional groups.

We take the minor POS category plus the case and possessive agreement markers for nominals and participle adjectives to make up the POS feature of each IG.[21] However, we do not employ dynamic selection of these features and just use the same strategy for both dependents and the heads. The reason is that, in this parser, we do not make the assumption that the head is always on the right side of the dependent, but also try to find head-initial dependencies, and the parser does not know at a given stage if a unit is a candidate head or dependent. In the IG-based model, *InnerWord* relations (Figure 5), which are actually determined by the morphological analyzer, are processed deterministically without consulting the SVM classifiers.[22]

The feature model (Feature Model #1) to be used in these experiments is shown in Figure 6. This feature model uses five POS features, defined by the POS of the two topmost stack tokens (σ_0, σ_1), the first two tokens of the remaining input (τ_0, τ_1), and the token which comes just after the topmost stack token in the actual sentence ($\sigma_0 + 1$). The dependency type features involve the top token on the stack (σ_0), its leftmost and rightmost dependent ($l(\sigma_0)$, $r(\sigma_0)$), and the leftmost dependent of the next input token ($l(\tau_0)$).

The results for this feature model and the two representation models can be seen in Table 4. We again see that the IG-based model outperforms the word-based model. When we compare the unlabeled (AS_U) scores with the results of the probabilistic parser (from Table 2), we see that we do not obtain any improvements neither for the IG-based model nor for the word-based model. This is probably the combined effect of not using

21 Thus, we are actually combining some inflectional features with the part-of-speech category and use them together in the POS feature.

22 Because only the first IG of a word carries the stem information (and the remaining IGs has null "_" values for this field), a lexicalized model can easily determine the InnerWord links without need for a deterministic model. For the unlexicalized models, it is necessary to process *InnerWord* relations deterministically in order to get the full benefit of IG-based parsing, because the classifiers cannot correctly predict these relations without lexical information (Eryiğit, Nivre, and Oflazer 2006). However, for the lexicalized models, adding deterministic *InnerWord* processing has no impact at all on parsing accuracy, but it reduces training and parsing time by reducing the number of training instances for the SVM classifiers.

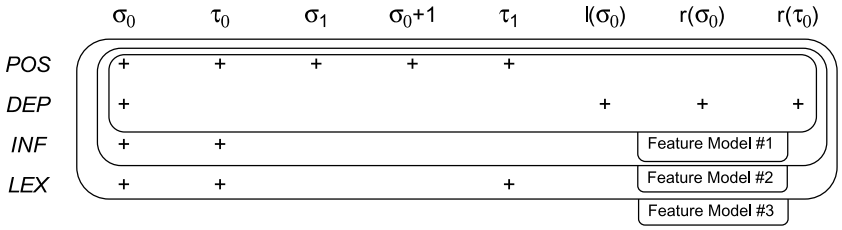


Figure 6
Feature models for the classifier-based parser.

Table 4
Unlabeled and labeled attachment scores for the unlexicalized classifier-based parser.

Parsing Model	AS_U	AS_L
Word-based model	67.1±0.3	57.8±0.3
IG-based model	70.6±0.2	60.9±0.3

the lexical information for head-initial dependencies that we use in our rules in the probabilistic parser, and of not using dynamic selection.[23]

5.2.2 Exploiting the Advantages of the Classifier-Based Parser. To exploit the advantages of the classifier-based parser, we now describe a setting which does not rely on any linguistic knowledge on the selection of inflectional features and lets the classifier of the parser select the useful combinations of the features. As SVMs can perform such tasks successfully, we now explore different representations of the morphological data in the IG-based model to see if the performance can be improved.

As shown in earlier examples, the inflectional information available for a given token normally consists of a complex combination of atomic features such as +A3sg, +Pnon, and +Loc. Eryiğit, Nivre, and Oflazer (2006) showed that adding inflectional features as atomic values to the feature models was better than taking certain subsets with linguistic intuition and trying to improve on them. Thus we now present results with the feature model where the POS component only comprises the minor part of speech and the INF comprises all the other inflectional features provided by the treebank without any reduction. We investigate the impact of this approach first with an unlexicalized model (Feature Model #2 in Figure 6) and then with a lexicalized model (Feature Model #3 in Figure 6) where we investigate two different kinds of lexicalization: one using just the root information and one using the complete surface form as lexical features.

Table 5 gives the results for both unlexicalized and lexicalized models with INF features included in the feature model. We can see the benefit of using inflectional features separately and split into atomic components, by comparing the first line of the table with the best results for the IG-based model in Table 4. We can also note

23 Actually, the equivalent of this IG-based model is the probabilistic model #3 in Table 3 (with no dynamic selection), which does not do significantly better than this classifier-based model.

Table 5
Unlabeled and labeled attachment scores for enhancements of the IG-based model.

Feature Model	AS_U	AS_L
Feature Model #2 (no lexicalization)	72.4±0.2	63.1±0.3
Feature Model #3 (lex. with surface forms)	75.7±0.2	66.6±0.3
Feature Model #3 (lex. with roots)	76.0±0.2	67.0±0.3

the improvement that lexicalized models bring:[24] In contrast to the probabilistic parser, lexicalization using root information rather than surface form gives better performance, even though the difference is not statistically significant. The improvement in AS_U score is 3.9 percentage points for the lexicalized model (with root) over the IG-based model of the probabilistic parser with AS_U=72.1±0.3. A similar case can be observed for WW_U accuracies: Including INF and lexicalization with roots gives WW_U=82.7±0.5 on the entire treebank, which provides an improvement of 3.3 percentage points over the IG-based model of the probabilistic parser (with WW_U=79.0±0.7).

6. The Impact of Inflectional Features and Lexicalization

In the previous sections, we presented our parsers using optimized parameters and feature representations. We have observed that using complete inflectional features and lexicalized models improves the accuracy of the classifier-based parser significantly, whereas for the probabilistic parser adding these features has a negative impact on accuracy. In this section, we investigate the influence of different inflectional features and lexical information on both parsers using the best performing IG-based models, in order to get a more fine-grained picture. The results of the experiments with the classifier-based parser are not strictly comparable to those of other experiments, because the training data have here been divided into smaller sets (based on the major part of speech category of the next token) as a way of reducing SVM training times without a significant decrease in accuracy. For the probabilistic parser, we have not used dynamic selection while investigating the impact of inflectional features.

6.1 Inflectional Features

In order to see the influence of inflectional features, we tested six different sets, where each set includes the previous one and adds some more inflectional features. The following list describes each set in relation to the previous one:

Set 1 No inflectional features except for minor part of speech
Set 2 Set 1 + case and possessive markers for nominals, possessive markers for participle adjectives
Set 3 Set 2 + person/number agreement features for nominals and verbs
Set 4 Set 3 + all inflectional features for nominals

24 The unlabeled exact match score (that is, the percentage of sentences for which *all* dependencies are correctly determined) for this best performing model is 37.5% upon IG-based evaluation and 46.5% upon word-based evaluation.

Set 5 Set 4 + all inflectional features for verbs
Set 6 Set 5 + all inflectional features

Figure 7 shows the results for both the probabilistic and the classifier-based parser. The results shown in Figures 7b confirm the importance of case and possessive features, which was presupposed in the manual selection of features in Section 4. Besides these, the number/person agreement features available for nominals and verbs are also important inflectional features even though they do not provide any statistically significant increase in accuracy (except for AS_U in Figure 7b [Set 3]). Another point that merits attention is the fact that the labeled accuracy is affected more by the usage of inflectional features compared to unlabeled accuracy. The difference between Set 1 and Set 2 (in Figure 7b) is nearly 4 percentage points for AS_U and 10 percentage points for AS_L. It thus appears that inflectional features are especially important in order to determine the type of the relationship between the dependent and head units. This is logical because in Turkish it is usually not the word order that determines the roles of the constituents in a sentence, but the inflectional features (especially the case markers). We again see from these figures that the classifier-based parser does not suffer from sparse data even if we use the full set of inflectional features (Set 6) provided by the treebank, whereas the probabilistic parser starts having this problem even with Set 3 (Figure 7a). The problem gets worse when we add the complete set of inflectional features.

6.2 Lexicalization

In order to get a more fine-grained view of the role of lexicalization, we have investigated the effect of lexicalizing IGs from different major part-of-speech categories. We expand this analysis into POS categories where relevant. The results are shown in Table 6, where the first column gives the part-of-speech tag of the lexicalized units, and the second and third columns give the total frequency and the frequency of distinct roots for that part-of-speech tag. We again see that the probabilistic parser suffers from sparse data especially for part-of-speech tags that appear with a high number of distinct roots. We cannot observe any increase with the lexicalization of any category. The situation is different for the classifier-based parser. None of the individual lexicalizations causes a decrease. We see that the lexicalization of nouns causes a significant increase in accuracy.

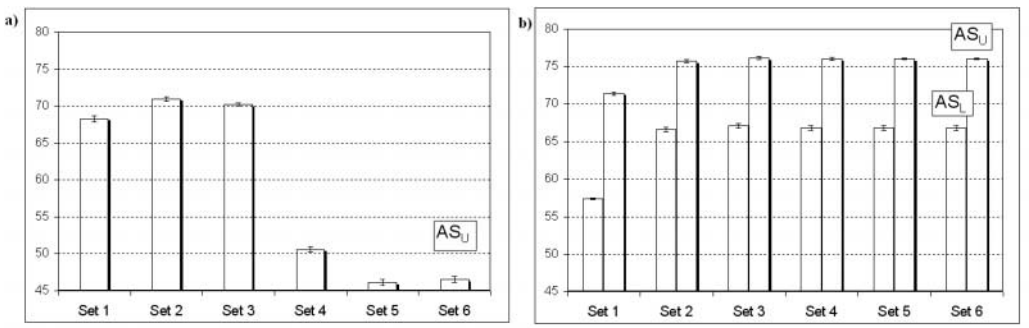


Figure 7
Accuracy for feature sets 1–6:
a) Unlabeled accuracy for probabilistic parser
b) Unlabeled and labeled accuracy for classifier-based parser

Table 6
Unlabeled and labeled attachment scores for limited lexicalization (n = count, d = number of distinct roots).

			Probabilistic	Classifier-based	
	n	*d*	AS_U	AS_U	AS_L
None	-	-	*72.1*±0.3	*72.8*±0.2	*63.2*±0.3
Adjectives	6446	735	68.7±0.2	72.9±0.2	63.2±0.3
Adverbs	3033	221	69.8±0.3	73.1±0.2	63.4±0.3
Conjunctions	2200	44	67.8±0.4	**74.1**±0.2	**64.2**±0.3
Determiners	1998	13	71.8±0.3	72.8±0.2	63.3±0.3
Duplications	11	9	72.0±0.3	72.8±0.2	63.2±0.3
Interjections	100	34	72.0±0.3	72.8±0.2	63.2±0.3
Nouns	21860	3935	53.7±0.3	**73.9**±0.2	**64.6**±0.3
Numbers	850	226	71.4±0.3	72.9±0.2	63.3±0.3
Post-positions	1250	46	70.9±0.3	72.9±0.2	63.2±0.3
Pronouns	2145	28	72.0±0.2	72.8±0.2	63.2±0.3
Punctuations	10420	16	72.1±0.3	73.4±0.2	63.7±0.3
Questions	228	6	71.9±0.2	72.8±0.2	63.2±0.3
Verbs	14641	1256	59.9±0.4	72.9±0.2	**63.8**±0.3

Lexicalization of verbs also gives a noticeable increase in the labeled accuracy even though this is not statistically significant. A further investigation on the minor parts of speech of nouns[25] shows that only common nouns have this positive effect, whereas the lexicalization of proper nouns does not improve accuracy. We see that the lexicalization of conjunctions also improves the accuracy significantly. This improvement can be attributed to the enclitics (such as *de*, *ki*, *mi*, written on the right side of and separately from the word they attach to), which give rise to head-initial dependencies. These enclitics, which are annotated as conjunctions in the treebank, can be differentiated from other conjunctions by lexicalization which makes it very easy to connect them to their head on the left.

Because we did not observe any improvement in the probabilistic parser, we continued further experimentation only with the classifier-based parser. We tried partially lexicalized models by lexicalizing various combinations of certain POS categories (see Figure 8). The results show that, whereas lexicalization certainly improves parsing accuracy for Turkish, only the lexicalization of conjunctions and nouns together has an impact on accuracy. Similarly to the experiments on inflectional features, we again see that the classifier-based parser has no sparse data problem even if we use a totally lexicalized model.

Although the effect of lexicalization has been discussed in several studies recently (Dubey and Keller 2003; Klein and Manning 2003; Arun and Keller 2005), it is often investigated as an all-or-nothing affair, except for a few studies that analyze the distributions of lexical items, for example, Bikel (2004) and Gildea (2001). The results for

25 IGs with a noun part-of-speech tag other than common nouns are marked with an additional minor part of speech that indicates whether the nominal is a proper noun or a derived form—one of future participle, past participle, infinitive, or a form involving a zero-morpheme derivation. These latter four do not contain any root information.

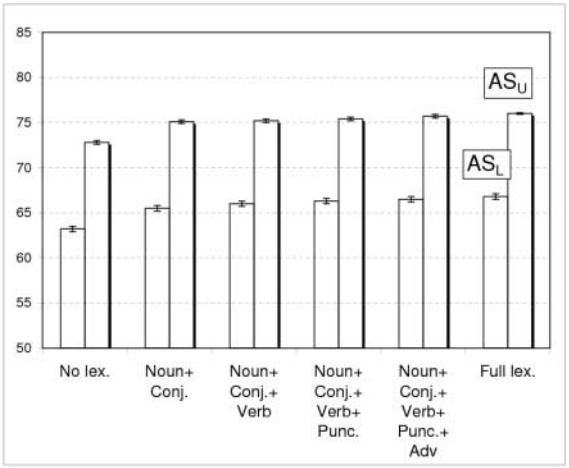


Figure 8
Unlabeled and labeled attachment scores for incrementally extended lexicalization for the classifier-based parser.

Turkish clearly show that the effect of lexicalization is not uniform across syntactic categories, and that a more fine-grained analysis is necessary to determine in what respects lexicalization may have a positive or negative influence. For some models (especially those suffering from sparse data), it may even be a better choice to use some kind of limited lexicalization instead of full lexicalization, although the experiments in this article do not show any example of that. The results from the previous section suggests that the same is true for morphological information, but this time showing that limited addition of inflectional features (instead of using them fully) helps to improve the accuracy of the probabilistic parser.

7. The Impact of Training Set Size

In order to see the influence of the training set size on the performance of our parsers, we designed the experiments shown in Figure 9, where the x-axis shows the number of cross validation subsets that we used for training in each step. Figure 9 gives the AS_U scores for the probabilistic parser (unlexicalized except for head-initial rules) and the classifier-based parser (unlexicalized and lexicalized). We observe that the relative improvement with growing training set size is largest for the classifier-based lexicalized model with a relative difference of 5.2 ± 0.2 between using nine training subsets and one training subset, whereas this number is 4.6 ± 0.3 for the unlexicalized classifier-based model and 2.5 ± 0.2 for the unlexicalized probabilistic model. We can state that despite its lower accuracy, the probabilistic model is less affected by the size of the training data. We can see from this chart that the relative ranking of the models remain the same, except for sizes 1–3, where the probabilistic parser does better (or no worse than) the unlexicalized classifier-based models. Another conclusion may be that classifier-based models are better at extracting information with the increasing size of the data in hand, whereas the probabilistic model cannot be improved very much with the increasing size of the data. We can observe this situation especially in the lexicalized model which is improved significantly between size = 6 subsets and size = 9 subsets, whereas there is no significant improvement on the unlexicalized models within this interval.

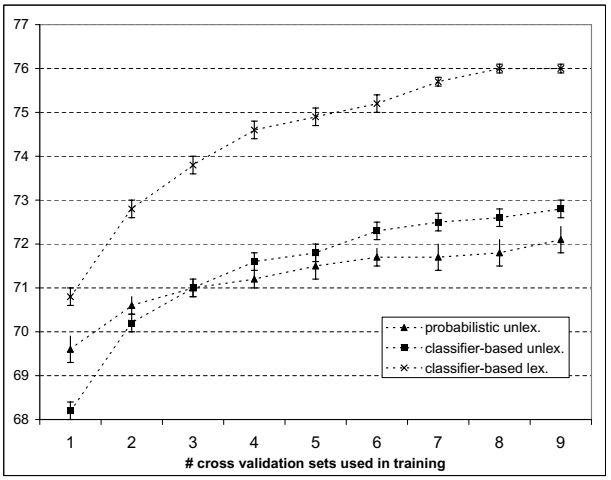


Figure 9
Unlabeled attachment score for different training set sizes.

8. Error Analysis

In this section, we present a detailed error analysis on the results of our best performing parser. We first evaluate our results on different dependency types. We then investigate the error distribution in terms of distance between the head assigned by the parser and the actual head. Finally, we look at the error distribution in relation to sentence length. In the analysis, the results are aggregated over all ten folds of the cross-validation.

8.1 Accuracy per Dependency Type

Table 7 gives the AS_U, labeled precision, labeled recall and labeled F-score for individual dependency types. The table is sorted according to the AS_U results, and the average distance between head and dependent is given for each type.

We see that the parser cannot find labeled dependencies for the types that have fewer than 100 occurrences in the treebank, with the single exception of RELATIVIZER, the enclitic *ki* (conjunction), written separately from the word it attaches to. Because this dependency type always occurs with the same particle, there is no sparse data problem.

If we exclude the low-frequency types, we can divide the results into three main groups. The first group consists of determiners, particles, and nominals that have an AS_U score over 79% and link to nearby heads. The second group mainly contains subjects, objects, and different kinds of adjuncts, with a score in the range 55–79% and a distance of 1.8–4.6 IGs to their head. This is the group where inflectional features are most important for finding the correct dependency. The third group contains distant dependencies with a much lower accuracy. These are generally relations like sentence modifier, vocative, and apposition, which are hard to find for the parser because they cannot be differentiated from other nominals used as subjects, objects, or normal modifiers. Another construction that is hard to parse correctly is coordination, which may require a special treatment.

Table 7
Attachment score (AS_U), labeled precision (P), labeled recall (R) and labeled F-score for each dependency type in the treebank (n = count, dist = dependency length).

Label	*n*	*dist*	$\mathbf{AS}_U$	P	R	F
SENTENCE	7,252	1.5	90.5	87.4	89.2	88.3
DETERMINER	1,952	1.3	90.0	84.6	85.3	85.0
QUESTION.PARTICLE	288	1.3	86.1	80.0	76.4	78.2
INTENSIFIER	903	1.2	85.9	80.7	80.3	80.5
RELATIVIZER	85	1.2	84.7	56.6	50.6	53.4
CLASSIFIER	2,048	1.2	83.7	74.6	71.7	73.1
POSSESSOR	1,516	1.9	79.4	81.6	73.6	77.4
NEGATIVE.PARTICLE	160	1.4	79.4	76.4	68.8	72.4
OBJECT	7,956	1.8	75.9	63.3	62.5	62.9
MODIFIER	11,685	2.6	71.9	66.5	64.8	65.7
DATIVE.ADJUNCT	1,360	2.4	70.8	46.4	50.2	48.2
FOCUS.PARTICLE	23	1.1	69.6	0.0	0.0	0.0
SUBJECT	4,479	4.6	68.6	50.9	56.2	53.4
ABLATIVE.ADJUNCT	523	2.5	68.1	44.0	54.5	48.7
INSTRUMENTAL.ADJUNCT	271	3.0	62.7	29.8	21.8	25.2
ETOL	10	4.2	60.0	0.0	0.0	0.0
LOCATIVE.ADJUNCT	1,142	4.2	56.9	43.3	48.4	45.7
COORDINATION	814	3.4	54.1	53.1	49.8	51.4
S.MODIFIER	594	9.6	50.8	42.2	45.8	43.9
EQU.ADJUNCT	16	3.7	50.0	0.0	0.0	0.0
APPOSITION	187	6.4	49.2	49.2	16.6	24.8
VOCATIVE	241	3.4	42.3	27.2	18.3	21.8
COLLOCATION	51	3.3	41.2	0.0	0.0	0.0
ROOT	16	-	0.0	0.0	0.0	0.0
Total	43,572	2.5	76.0	67.0	67.0	67.0

8.2 Error Distance

When we evaluate our parser based on the dependency direction, we obtain an AS_U of 72.2 for head-initial dependencies and 76.2 for head-final ones. Figure 10a and Figure 10b give the error distance distributions for head-initial and head-final dependencies based on the unlabeled performance of the parser. The *x*-axis in the figures gives the difference between indexes of the assigned head IG and the real head IG.

As stated previously, the head-initial dependencies constitute 5% of the entire dependencies in the treebank. Figure 10a shows that for head-initial dependencies the parser has a tendency to connect the dependents to a head closer than the real head or in the wrong direction. When we investigate these dependencies, we see that 70% of them are connected to a head adjacent to the dependent and the parser finds 90% of these dependencies correctly. Thus, we can say that the parser has no problem in finding adjacent head-initial dependencies. Moreover, 87% of the errors where the error distance is equal to 1 (Figure 10a)[26] are due to the dependents being connected to the wrong IG

26 Meaning that the actual head and assigned head are adjacent.

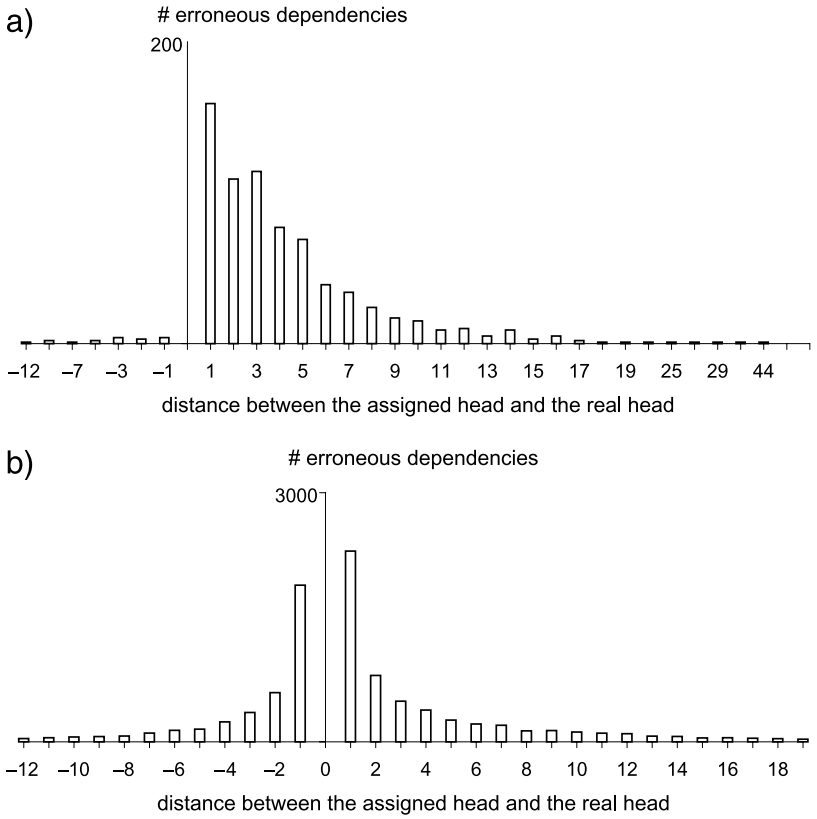


Figure 10
Error distance distributions a) for head-initial dependencies b) for head-final dependencies.

of the correct head word. When we investigate the ability of the parser in finding the dependency direction, we see that our parser has a high precision value (91%) and a relatively lower recall value (80%).

The parser is 100% successful in finding the direction of head-final dependencies. Furthermore, the errors that it makes while determining the correct head have a roughly normal distance distribution, as can be seen from Figure 10b.[27] We can see from the same figure that 57% of the errors fall within the interval of ± 2 IGs away from the actual head.

8.3 Sentence Length

Figure 11 shows the distribution of errors over sentences of different lengths. The x-axis shows the sentence length (measured in number of dependencies), the y-axis shows the error count, and the z-axis shows the number of sentences. As expected, the distribution is dominated by short sentences with few errors (especially sentences of up to seven dependencies with one error). The mean number of errors appears to be a linear function of sentence length, which would imply that the error probability

27 Error distances with less than 40 occurrences are excluded from the figure.

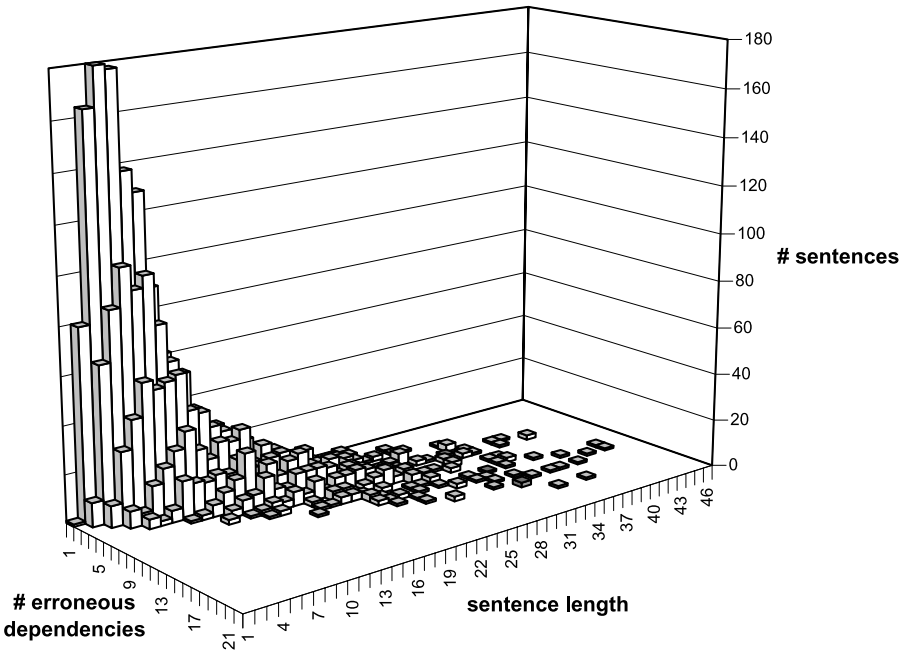


Figure 11
Error distribution versus sentence length.

per word does not increase with sentence length. This is interesting in that it seems to indicate that the classifier-based parser does not suffer from error propagation despite its greedy, deterministic parsing strategy.

9. The Impact of Morphological Disambiguation

In all of the experiments reported herein, we have used the gold-standard tags provided by the treebank. Another point that deserves investigation is therefore the impact of using tags automatically assigned by a morphological disambiguator, in other words the accuracy of the parser on raw text. The role of morphological disambiguators for highly inflectional languages is far more complex than assigning a single main POS category (e.g., Noun, Verb, Adj) to a word, and also involves assigning the correct morphological information which is crucial for higher level applications. The complexity of morphological disambiguation in an agglutinative language like Turkish is due to the large number of morphological feature tag combinations that can be assigned to words. The number of potential morphological tag combinations in Turkish for all practical purposes is very large due to productively derived forms.[28]

The two subsequent examples, for the words *kalemi* and *asmadan*, expose the two phenomena that a Turkish morphological disambiguator should deal with. The outputs of the morphological analyzer are listed below the words. The first example shows that all three possible analyses of the word *kalemi* have "Noun" as the POS category but they differ in that they have different stems and inflectional features. In the second example

28 For the treebank data, the number of distinct combinations of morphological features is 718 for the word-based model of the classifier-based parser and 108 for the IG-based model.

we see that the possible analyses also have different IG segmentations; the first two analyses of the word *asmadan* consists of two IGs whereas the last one has one IG.

kalemi
kale +*Noun*+*A3sg*+*P1sg*+*Acc* ('my castle' in accusative form)
kalem +*Noun*+*A3sg*+*P3sg*+*Nom* ('his pencil')
kalem +*Noun*+*A3sg*+*Pnon*+*Acc* ('the pencil' in accusative form)

asmadan
as +*Verb*+*Pos DB* +*Adverb*+*WithoutHavingDoneSo* ('without having hanged (it)')
as +*Verb*+*Pos DB* +*Noun*+*Inf2*+*A3sg*+*Pnon*+*Abl* ('from hanging (it)')
asma +*Noun*+*A3sg*+*Pnon*+*Abl* ('from the vine')

The task of the morphological disambiguator is to choose one of the possible morphological analyses and thus to find the correct inflectional features including parts of speech, and the IG structure. We first used the two-level morphological analyzer of Oflazer (1994) to analyze all the words in the treebank.[29] This morphological analyzer simultaneously produces the IG segmentation and the relevant features encoded in all analyses of a word form. We then used the morphological disambiguator of Yüret and Türe (2006), which has a reported accuracy of 96% for Turkish.

When tested on our treebank data, the accuracy of the morphological disambiguator is 88.4%, including punctuation (which is unambiguous) and using a lookup table for the words that are not recognized by the morphological analyzer.[30] The lower accuracy of the morphological disambiguator on the treebank can be due to different selections in the annotation process of the morphological disambiguator training data (Yüret and Türe 2006), which is totally different from the treebank data.

In order to investigate the impact of morphological disambiguation errors, we used our best IG-based model and a lexicalized word-based model with our classifier-based parser.[31] We again evaluated our parsing models with AS_U, AS_L, and WW_U scores. There is no problem when evaluating with WW_U scores because this metric only takes into account whether the head word assigned to a dependent is correct or not, which means that any errors of the morphological disambiguator can be ignored. Similarly, in calculating AS_U and AS_L scores for the word-based model, dependencies are assumed to be connected to the first IG of the head word without taking into consideration any errors in tags caused by the morphological disambiguator. But when evaluating with the AS_U and AS_L scores for the IG-based model, one problem that may appear is that the disambiguator may have assigned a totally different IG structure to the head word, compared to the gold standard (cf. the three analyses of the word *asmadan*). In this case, we accept a dependency link to be correct if the dependent is connected to the correct head word and the head IG has the same POS category as the gold-standard. This is reasonable because we know that some of the errors in inflectional features do not affect the type of dependency very much. For example, if we put the adjective *küçük* ('small')

29 We noted that 39% of the words were ambiguous and 17% had more than two distinct morphological analyses.

30 The words not recognized by the morphological analyzer are generally proper nouns, numbers, and some combined words that are created in the development stage of the treebank and constitute 6.2% of the whole treebank. If these words are excluded, the accuracy of the tagger is 84.6%.

31 For this model, we added LEX features for σ_0, τ_0, τ_1 to the feature model of our word-based model in Table 4.

Table 8
Impact of morphological disambiguation on unlabeled and labeled attachment scores and word-to-word scores.

		AS_U	AS_L	WW_U
Word-based	*Gold standard*	*71.2*±0.3	*62.3*±0.3	*82.1*±0.9
	Tagged	69.5±0.3	59.3±0.3	80.2±0.9
IG-based	*Gold standard*	*76.0*±0.2	*67.0*±0.3	*82.7*±0.5
	Tagged	73.3±0.3	63.2±0.3	80.6±0.7

in front of the example given previously (*küçük kalemi*), then the choice of morphological analysis of the noun has no impact on the fact that the adjective should be connected to the noun with dependency type "MODIFIER". Moreover, most of the errors in POS categories will actually prevent the parser from finding the correct head word, which can be observed from the drop in WW_U accuracy.

Table 8 shows that the IG-based model and the word-based model are equally affected by the tagging errors and have a drop in accuracy within similar ranges. (It can also be seen that, even with automatically tagged data, the IG-based model gives better accuracy than the word-based model.) We can say that the use of an automatic morphological analyzer and disambiguator causes a drop in the range of 3 percentage points for unlabeled accuracy and 4 percentage points for labeled accuracy (for both word-based and IG-based models).

10. Related Work

The first results on the Turkish Treebank come from Eryiğit and Oflazer (2006) where the authors used only a subset of the treebank sentences containing exclusively head-final and projective dependencies. The parser used in that paper is a preliminary version of the probabilistic parser used in this article. The first results on the entire treebank appear in Nivre et al. (2007), where the authors use memory-based learning to predict parser actions, and in Eryiğit, Adalı, and Oflazer (2006), which introduces the rule-based parser used in this article.

The Turkish Treebank has recently been parsed by 17 research groups in the CoNLL-X shared task on multilingual dependency parsing (Buchholz and Marsi 2006), where it was seen as the most difficult language by the organizers and most of the groups.[32] The following quote is taken from Buchholz and Marsi (page 161): "The most difficult data set is clearly the Turkish one. It is rather small, and in contrast to Arabic and Slovene, which are equally small or smaller, it covers 8 genres, which results in a high percentage of new FORM and LEMMA values in the test set."

The results for Turkish are given in Table 9. Our classifier-based parser obtained the best results for Turkish (with AS_U=75.8 and AS_L=65.7) and also for Japanese, which is the only agglutinative and head-final language in the shared task other than Turkish (Nivre et al. 2006). The groups were asked to find the correct IG-to-IG dependency links. When we look at the results, we observe that most of the best performing parsers use

32 The Turkish data used in the shared task is actually a modified version of the treebank used in this article; some conversions are made on punctuation structures in order to keep consistency between all languages.

Table 9
CoNLL-X shared task results on Turkish (taken from Table 5 in Buchholz and Marsi [2006]).

Teams	AS_U	AS_L
Nivre et al. (2006)	75.8	65.7
Johansson and Nugues (2006)	73.6	63.4
McDonald, Lerman, and Pereira (2006)	74.7	63.2
Corston-Oliver and Aue (2006)	73.1	61.7
Cheng, Asahara, and Matsumoto (2006)	74.5	61.2
Chang, Do, and Roth (2006)	73.2	60.5
Yüret (2006)	71.5	60.3
Riedel, Çakıcı, and Meza-Ruiz (2006)	74.1	58.6
Carreras, Surdeanu, and Marquez (2006)	70.1	58.1
Wu, Lee, and Yang (2006)	69.3	55.1
Shimizu (2006)	68.8	54.2
Bick (2006)	65.5	53.9
Canisius et al. (2006)	64.2	51.1
Schiehlen and Spranger (2006)	61.6	49.8
Dreyer, Smith, and Smith (2006)	60.5	46.1
Liu et al. (2006)	56.9	41.7
Attardi (2006)	65.3	37.8

one of the parsing algorithms of Eisner (1996), Nivre (2003), or Yamada and Matsumoto (2003) together with a learning method based on the maximum margin strategy. We can also see that a common property of the parsers which fall below the average (AS_L=55.4) is that they do not make use of inflectional features, which is crucial for Turkish.[33]

Another recent study that has promising results is Çakıcı and Baldridge (2006), where the authors use the MSTParser (McDonald, Lerman, and Pereira 2006), also used in the CoNLL-X shared task (line 3 in Table 9). Following the work of Eryiğit and Oflazer (2006) and Nivre et al. (2006), they use the stem information and the case information for nominals and they also report an increase in performance by using these features. Similar to one of the models ("INF as a single feature") in Eryiğit, Nivre, and Oflazer (2006), where the feature names of the suffixes provided by the morphological analyzer are concatenated and used as a feature to the classifier, they use the surface forms of the suffixes as a whole. We can say that the models in this article cover this approach in that each suffix is used as a single feature name (which is shown to perform better than using them concatenated to each other in Eryiğit, Nivre, and Oflazer [2006]). Because in Turkish, the same suffixes take different forms under vowel harmony[34] and the surface forms of some different suffixes are structurally ambiguous,[35] using them with their feature names is actually more meaningful. Çakıcı and Baldridge (2006) report a word-to-word accuracy of 84.9%, which seems competitive, but unfortunately from this we

33 Actually, there are two parsers (Bick 2006 and Attardi 2006 in Table 9) in this group which try to use parts of the inflectional features under special circumstances.

34 For example, in the words *ev+de* ('at home') and *okul+da* ('at school'), the suffixes *-de* and *-da* are the same locative case suffixes (+*Loc*) but they take different forms due to vowel harmony.

35 For example, in the word *ev+in*, the surface morpheme *-in* may indicate both a second singular possessive suffix (+*P2sg*) which will give the word the meaning of 'your house' and a genitive case (+*Gen*) which will give the word the meaning of 'of the house', as the underlying *lexical* morphemes are different.

are not able to gauge the IG-to-IG accuracy which we have argued *is* the right metric to use for Turkish, and their results are not comparable to any of the results in the literature, because they have not based their experiments on any of the official releases of the treebank. In addition, they use an evaluation metric different from the ones in the literature in that they only excluded some of the punctuations from the evaluation score.

11. Conclusions

In this article, we have investigated a number of issues in data-driven dependency parsing of Turkish. One of the main results is that IG-based models consistently outperform word-based models. This result holds regardless of whether we evaluate accuracy on the word level or on the IG level; it holds regardless of whether we use the probabilistic parser or the classifier-based parser; and it holds even if we take into account the problem caused by errors in automatic morphological analysis and disambiguation.

Another important conclusion is that the use of morphological information can increase parsing accuracy substantially. Again, this result has been obtained both for the probabilistic and the classifier-based parser, although the probabilistic parser requires careful manual selection of relevant features to counter the effect of data sparseness. A similar result has been obtained with respect to lexicalization, although in this case an improvement has only been demonstrated for the classifier-based parser, which is probably due to its greater resilience to data sparseness.

By combining the deterministic classifier-based parsing approach with an adequate use of IG-based representations, morphological information, and lexicalization, we have been able to achieve the highest reported accuracy for parsing the Turkish Treebank.

Acknowledgments

We are grateful for the financial support from TUBITAK (The Scientific and Technical Research Council of Turkey) and Istanbul Technical University. We want to thank Johan Hall and Jens Nilsson in the language technology group at Växjö University for their contributions to the classifier-based parser framework (MaltParser) within which we developed the classifier-based parser for Turkish. We also want to thank Deniz Yüret for providing us with his morphological disambiguator, and Eşref Adalı for his valuable comments. Finally, we want to thank our three anonymous reviewers for insightful comments and suggestions that helped us improve the final version of the article.

References

Arun, Abhishek and Frank Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of French. In *Proceedings of ACL'05*, pages 302–313, Ann Arbor, MI.

Attardi, Giuseppe. 2006. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of CONLL-X*, pages 166–170, New York, NY.

Bick, Eckhard. 2006. Lingpars, a linguistically inspired, language-independent machine learner for dependency treebanks. In *Proceedings of CONLL-X*, pages 171–175, New York, NY.

Bikel, Daniel M. 2004. A distributional analysis of a lexicalized statistical parsing model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 182–189, Barcelona.

Bikel, Daniel M. and David Chiang. 2000. Two statistical parsing models applied to the Chinese treebank. In *Proceedings of the 2nd Chinese Language Processing Workshop*, pages 1–6, Hong Kong.

Black, Ezra, Frederick Jelinek, John D. Lafferty, David M. Magerman, Robert L. Mercer, and Salim Roukos. 1992. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 5th DARPA Speech and Natural Language Workshop*, pages 31–37, New York, NY.

Bozşahin, Cem. 2002. The combinatory morphemic lexicon. *Computational Linguistics*, 28(2):145–186.

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

Çakıcı, Ruket and Jason Baldridge. 2006. Projective and non-projective Turkish parsing. In *Proceedings of the 5th International Treebanks and Linguistic Theories Conference*, pages 43–54, Prague.

Canisius, Sander, Toine Bogers, Antal van den Bosch, Jeroen Geertzen, and Erik Tjong Kim Sang. 2006. Dependency parsing by inference over high-recall dependency predictions. In *Proceedings of CONLL-X*, pages 176–180, New York, NY.

Carreras, Xavier, Mihai Surdeanu, and Lluis Marquez. 2006. Projective dependency parsing with perceptron. In *Proceedings of CONLL-X*, pages 181–185, New York, NY.

Chang, Chih-Chung and Chih-Jen Lin, 2001. *LIBSVM: A Library for Support Vector Machines*. Software available at `www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chang, Ming-Wei, Quang Do, and Dan Roth. 2006. A pipeline model for bottom-up dependency parsing. In *Proceedings of CONLL-X*, pages 186–190, New York, NY.

Cheng, Yuchang, Masayuki Asahara, and Yuji Matsumoto. 2006. Multi-lingual dependency parsing at NAIST. In *Proceedings of CONLL-X*, pages 191–195, New York, NY.

Chung, Hoojung and Hae-Chang Rim. 2004. Unlexicalized dependency parser for variable word order languages based on local contextual pattern. In *Proceedings of the 5th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 109–120, Seoul.

Collins, Michael. 1996. A new statistical parser based on bigram lexical dependencies. In *Proceedings of ACL'96*, pages 184–191, Santa Cruz, CA.

Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL'97*, pages 16–23, Madrid.

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

Collins, Michael, Jan Hajic, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser for Czech. In *Proceedings of ACL'99*, pages 505–518, College Park, MD.

Corazza, Anna, Alberto Lavelli, Giorgio Satta, and Roberto Zanoli. 2004. Analyzing an Italian treebank with state-of-the-art statistical parsers. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories*, pages 39–50, Tübingen.

Corston-Oliver, Simon and Anthony Aue. 2006. Dependency parsing with reference to Slovene, Spanish and Swedish. In *Proceedings of CONLL-X*, pages 196–200, New York, NY.

Daelemans, Walter and Antal Vanden Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press, Cambridge.

Dreyer, Markus, David A. Smith, and Noah A. Smith. 2006. Vine parsing and minimum risk reranking for speed and precision. In *Proceedings of CONLL-X*, pages 201–205, New York, NY.

Dubey, Amit and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of ACL'03*, pages 96–103, Sapporo.

Eisner, Jason. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 340–345, Copenhagen.

Erguvanlı, Eser Emine. 1979. *The Function of Word Order in Turkish Grammar*. Ph.D. thesis, UCLA.

Eryiğit, Gülşen. 2006. *Türkçenin Bağlılık Ayrıştırması (Dependency Parsing of Turkish)*. Ph.D. thesis, Istanbul Technical University.

Eryiğit, Gülşen, Eşref Adalı, and Kemal Oflazer. 2006. Türkçe cümlelerin kural tabanlı bağlılık analizi [Rule-based dependency parsing of Turkish sentences]. In *Proceedings of the 15th Turkish Symposium on Artificial Intelligence and Neural Networks*, pages 17–24, Muğla.

Eryiğit, Gülşen, Joakim Nivre, and Kemal Oflazer. 2006. The incremental use of morphological information and lexicalization in data-driven dependency parsing. In *Computer Processing of Oriental Languages, Beyond the Orient: The Research Challenges Ahead*, pages 498–507, Singapore.

Eryiğit, Gülşen and Kemal Oflazer. 2006. Statistical dependency parsing of Turkish. In *Proceedings of EACL'06*, pages 89–96, Trento.

Gildea, Daniel. 2001. Corpus variation and parser performance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 167–202, Pittsburgh, PA.

Hajič, Jan, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, and Barbora Hladká. 2001. Prague dependency treebank 1.0 (final production label). CDROM CAT: LDC2001T10., ISBN 1-58563-212-0.

Hakkani-Tür, Dilek, Kemal Oflazer, and Gökhan Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Journal of Computers and Humanities*, 36(4):381–410.

Hoffman, Beryl. 1994. Generating context appropriate word orders in Turkish. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 117–126, Kennebunkport, ME.

Johansson, Richard and Pierre Nugues. 2006. Investigating multilingual dependency parsing. In *Proceedings of CONLL-X*, pages 206–210, New York, NY.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL'03*, pages 423–430, Sapporo.

Kromann, Matthias T. 2003. The Danish dependency treebank and the underlying linguistic theory. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, pages 217–220, Växjö.

Kudo, Taku and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 63–69, Taipei.

Levy, Roger and Christopher Manning. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *Proceedings of ACL'03*, pages 439–446, Sapporo.

Liu, Ting, Jinshan Ma, Huijia Zhu, and Sheng Li. 2006. Dependency parsing based on dynamic local optimization. In *Proceedings of CONLL-X*, pages 211–215, New York, NY.

Magerman, David M. 1995. Statistical decision-tree models for parsing. In *Proceedings of ACL'95*, pages 276–283, Cambridge, MA.

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

McDonald, Ryan, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CONLL-X*, pages 216–220, New York, NY.

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

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

Nivre, Joakim. 2006. *Inductive Dependency Parsing*. Springer, Dordrecht.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 49–56, Boston, MA.

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

Nivre, Joakim, Johan Hall, Jens Nilsson, Gülşen Eryiğit, and Stetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of CONLL-X*, pages 221–225, New York, NY.

Nivre, Joakim and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of ACL'05*, pages 99–106, Ann Arbor, MI.

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

Oflazer, Kemal. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.

Oflazer, Kemal. 2003. Dependency parsing with an extended finite-state approach. *Computational Linguistics*, 29(4):515–544.

Oflazer, Kemal, Bilge Say, Dilek Z. Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*. Kluwer, London, pages 261–277.

Ratnaparkhi, Adwait. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Providence, RI.

Riedel, Sebastian, Ruket Çakıcı, and Ivan Meza-Ruiz. 2006. Multi-lingual dependency parsing with incremental integer linear programming. In *Proceedings of CONLL-X*, pages 226–230, New York, NY.

Sagae, Kenji and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the 9th International Workshop on Parsing Technologies*, pages 125–132, Vancouver.

Schiehlen, Michael and Kristina Spranger. 2006. Language independent probabilistic context-free parsing bolstered by machine learning. In *Proceedings of CONLL-X*, pages 231–235, New York, NY.

Sekine, Satoshi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2000. Backward beam search algorithm for dependency analysis of Japanese. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 754–760, Saarbrücken.

Shimizu, Nobuyuki. 2006. Maximum spanning tree algorithm for non-projective labeled dependency parsing. In *Proceedings of CONLL-X*, pages 236–240, New York, NY.

Simov, Kiril, Gergana Popova, and Petya Osenova. 2002. HPSG-based syntactic treebank of Bulgarian (BulTreeBank). In Andrew Wilson, Paul Rayson, and Tony McEnery, editors, *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*. Lincom-Europa, Munich, pages 135–142.

Vapnik, Vladimir N. 1995. *The Nature of Statistical Learning Theory*. Springer, New York, NY.

Wu, Yu-Chieh, Yue-Shi Lee, and Jie-Chi Yang. 2006. The exploration of deterministic and efficient dependency parsing. In *Proceedings of CONLL-X*, pages 241–245, New York, NY.

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

Yüret, Deniz. 2006. Dependency parsing as a classification problem. In *Proceedings of CONLL-X*, pages 246–250, New York, NY.

Yüret, Deniz and Ferhan Türe. 2006. Learning morphological disambiguation rules for Turkish. In *Proceedings of HLT/NAACL'06*, pages 328–334, New York, NY.