The Incremental Use of Morphological Information and Lexicalization in Data-Driven Dependency Parsing

Gülşen Eryiğit[1] Joakim Nivre[2] Kemal Oflazer[3]

[1] Department of Computer Engineering, Istanbul Technical Univ., 34469 Turkey
[2] School of Mathematics and Systems Engineering, Växjö Univ., 35195 Sweden
[3] Faculty of Engineering and Natural Sciences, Sabancı Univ., 34956 Turkey

Abstract. Typological diversity among the natural languages of the world poses interesting challenges for the models and algorithms used in syntactic parsing. In this paper, we apply a data-driven dependency parser to Turkish, a language characterized by rich morphology and flexible constituent order, and study the effect of employing varying amounts of morpholexical information on parsing performance. The investigations show that accuracy can be improved by using representations based on inflectional groups rather than word forms, confirming earlier studies. In addition, lexicalization and the use of rich morphological features are found to have a positive effect. By combining all these techniques, we obtain the highest reported accuracy for parsing the Turkish Treebank.

1 Introduction

An important issue in empirically minded research on natural language parsing 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 [1], is applied to a new language, which often leads to a significant decrease in the measured accuracy [2, 3, 4, 5, 6]. However, it is often quite difficult to tease apart the influence of different features of the parsing methodology in the observed degradation of performance.

One topic that is prominent in the literature is the issue of lexicalization, i.e., to what extent the accuracy can be improved by incorporating features that refer to individual lexical items, as opposed to class-based features such as part-of-speech. 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 [7, 4, 8].

Another aspect, which so far has received less attention, is the proper 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 a language like Turkish, it has been shown that parsing accuracy can be improved by taking morphologically defined units rather than word forms as the basic units of syntactic structure [9].

In this paper, we study the role of lexicalization, morphological structure and morphological feature representations in data-driven dependency parsing of Turkish. More precisely, we compare representations based on the notion of *inflectional groups* proposed by Eryiğit and Oflazer [9] to a more traditional representation based on word forms, we experiment with different ways of representing morphological features in the input to the parser, and we compare lexicalized and

unlexicalized models to see how they interact with different representations of morphological structure and morphological features.

The parsing methodology is based on a deterministic parsing algorithm in combination with treebank-induced classifiers for predicting the next parsing action, an approach previously used for the analysis of Japanese [10], English [11, 12], Swedish [13] and Czech [14]. In this way, our study complements that of Eryiğit and Oflazer [9], which considers dependency parsing of Turkish in a probabilistic framework.

2 Turkish

Turkish is a flexible constituent order language. Even though in written texts, the constituent order of sentences generally conforms to the SOV or OSV structures, the constituents may freely change their position depending on the requirements of the discourse context. From the point of view of dependency structure, Turkish is predominantly (but not exclusively) head final.

Turkish has a very rich agglutinative morphological structure. Nouns can give rise to hundreds of different forms and verbs to many more. Furthermore, Turkish words may be formed through productive derivations, and it is not uncommon to find up to five derivations from a simple root. Previous work on Turkish, [15, 16, 9] has represented the morphological structure of Turkish words by splitting them into inflectional groups (IG). The root and derived forms of a word are represented by different IGs separated from each other by derivational boundaries. Each IG is then annotated with its own part-of-speech and any inflectional features. Figure 1 shows the IGs in a simple sentence: "küçük odadayım" (*I'm in the small room*). The word "odadayım" is formed from two IGs; a verb is derived from an inflected noun "odada" (*in the room*).

Dependency relations in a sentence always hold between the final IG of the dependent word and some IG of the head word [16, 9], so it is not sufficient to just identify the words involved in a dependency relation, but the exact IGs. In the example, the adjective "küçük" (*small*) should be connected to the first IG of the second word. It is the word "oda" (*room*) which is modified by the adjective, not the derived verb form "odadayım" (*I'm in the room*). So both the correct head word and the correct IG in the head word should be determined by the parser.

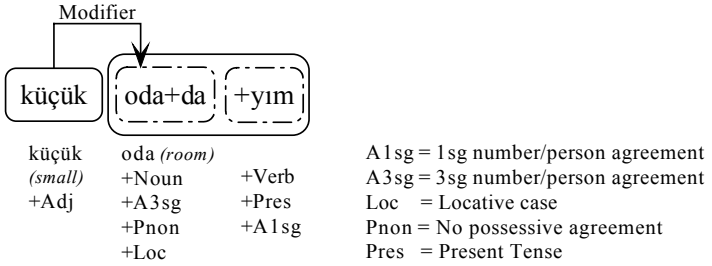

Fig. 1. Word and dependency representations

3 Parsing Framework

A prominent approach to data-driven dependency parsing in recent years is based on the combination of three techniques:

1. Deterministic parsing algorithms for building dependency graphs [10, 17]
2. History-based models for predicting the next parser action [18]
3. Discriminative classifiers to map histories to parser actions [19, 10]

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 or quadratic in the length of the sentence.

For the experiments in this paper we use the arc-standard variant of Nivre's parsing algorithm [17, 20, 21], 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 in a way similar to a shift-reduce parser.

The features of the history-based model can be defined in terms of different linguistic attributes of the input tokens, in particular the token on top of the stack, which we call the *top token*, and the first token of the remaining input, called the *next token*. The top token and the next token are referred to collectively as the *target tokens*, since they are the tokens considered as candidates for a dependency relation by the parsing algorithm. 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 (stem) (LEX), Part-of-speech category (POS), Inflectional features (INF), Dependency type (DEP).

To predict parser actions from histories, represented as feature vectors, we use support vector machines (SVM), which combine the maximum margin strategy introduced by Vapnik [22] 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 [10], Yamada and Matsumoto [11], and Sagae and Lavie [23], among others. To be more specific, we use the LIBSVM library for SVM learning [24], with a polynomial kernel of degree 2, with binarization of symbolic features, and with the one-versus-all strategy for multi-class classification.

4 Experimental Setup

The Turkish Treebank [15], created by METU and Sabancı University, has been used in the experiments. This treebank comprises 5635 sentences with gold standard morphological annotations and labeled dependencies between IGs. In the treebank, 7.2% of the sentences contain at least one dependency relation that is non-projective, not counting punctuation that is not connected to a head.[4] Each dependency link in the treebank starts from the final IG of the dependent word and ends in some IG of the head word.

Since the parsing algorithm can only construct projective dependency structures, we only use projective sentences for training but evaluate our models on the entire treebank.[5] More precisely, we use ten-fold cross-validation, where we randomly divide the treebank data into ten equal parts and in each iteration test the parser on one part, using the projective sentences of the remaining nine parts as training data.

[4] In the experiments reported in this paper, such dangling punctuation has been attached to the immediately following word in order to eliminate this uninteresting source of non-projectivity. Punctuation is also excluded in all evaluation scores.

[5] Our trial to use the pseudo-projective parsing strategy of Nivre and Nilsson [14] in order to process non-projective dependencies did not result in any improvement due to the limited amount of non-projective dependencies in the treebank.

The evaluation metrics used are the unlabeled (AS_U) and labeled (AS_L) attachment score, i.e., the proportion of tokens that are attached to the correct head (with the correct label for AS_L). A correct attachment implies that a dependent 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 is connected to (some IG in) the correct head word. Non-final IGs of a word are assumed to link to the next IG, but these links, referred to as *InnerWord* links, are not considered dependency relations and are excluded in evaluation scores. Results are reported as mean scores of the ten-fold cross-validation, with standard error, complemented if necessary by the mean difference between two models.

We use the following set of features in all the experiments described below:

- POS of the target tokens
- POS of the token immediately below the top token in the stack
- POS of the token immediately after the next token in the remaining input
- POS of the token immediately after the top token in the original input string
- DEP of the leftmost dependent of the top token
- DEP of the rightmost dependent of the top token
- DEP of the leftmost dependent of the next token

This is an unlexicalized feature model, involving only POS and DEP features, but we can get a lexicalized version by adding LEX features for the two target tokens. The value of each LEX feature is the stem of the relevant word or IG, rather than the full form. The reasoning behind this choice, which was corroborated in preliminary experiments, is that since the morphological information carried by the suffixes is also represented in the inflectional features, using the stem instead of the word form should not cause any loss of information and avoid data sparseness to a certain extent. The basic model, with and without lexicalization, is used as the starting point for our experiments. Additional features are explained in the respective subsections. An overview of the results can be found in table 1.

5 Inflectional Groups

In this set of experiments, we compare the use of IGs, as opposed to full word forms, as the basic tokens in parsing, which was found to improve parsing accuracy in the study of Eryiğit and Oflazer[9]. More precisely, we compare three different models:

- A word-based model, where the smallest units in parsing are words represented by the concatenation of their IGs.
- An IG-based model, where the smallest units are IGs and and where *InnerWord* relations are predicted by the SVM classifiers in the same way as real dependency relations.
- An IG-based model, where *InnerWord* relations are processed deterministically without consulting the SVM classifiers.

For these models, we use a reduced version of the inflectional features in the treebank, very similar to the reduced tagset used in the parser of Eryiğit and Oflazer [9]. For each IG, we use the part-of-speech of each IG and in addition include the case and possessive marker features if the IG is a nominal. Using this approach, the POS feature of the word "odadayım" becomes +Noun+Pnon-+Loc+Verb.

When lexicalizing the IG-based models, we use the stem for the first IG of a word but a null value ("_") for the remaining IGs of the same word. This representation also facilitates the deterministic processing of *InnerWord* relations in the third model, since any top token can be directly linked to a next token with LEX="_", provided that the two tokens are adjacent.

Section	Model	Unlexicalized		Lexicalized	
		AS_U	AS_L	AS_U	AS_L
5	Word-based	67.2±0.3	57.9±0.3	70.7±0.3	62.0±0.3
	IG-based	68.3±0.2	58.2±0.2	73.8±0.2	64.9±0.3
	IG-based deterministic	70.6±0.3	60.9±0.3	73.8±0.2	64.9±0.3
6	INF as single feature	71.6±0.2	62.0±0.3	74.4±0.2	65.6±0.3
	INF split	71.9±0.2	62.6±0.3	74.8±0.2	66.0±0.3
8	Optimized			76.0±0.2	67.0±0.3

Table 1. Summary table of experimental results

In order to calculate the accuracy for the word-based models, we assume that the dependent is connected to the first IG of the head word. This assumption is based on the observation that in the treebank, 85.6% of the dependency links land on the first (and possibly the only) IG of the head word, while 14.4% of the dependency links land on an IG other than the first one.

The parsing accuracy obtained with the three models, with and without lexicalization, is shown in table 1. The results are compatible with the findings of Eryiğit and Oflazer [9], despite a different parsing methodology, in that the IG-based models generally give higher parsing accuracy than the word-based model, with an increase of three percentage points for the best models.

However, the results also show that, for the unlexicalized model, it is necessary to process *InnerWord* relations deterministically in order to get the full benefit of IG-based parsing, since the classifiers cannot correctly predict these relations without lexical information. For the lexicalized model, 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.

6 Inflectional Features

Instead of taking a subset of the inflectional features and using them together with the main part-of-speech in the POS field, we now explore their use as separate features for the target tokens. From now on, our POS tag set therefore consists only of the main part-of-speech tags found in the treebank.

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`. Thus, when adding inflectional features to the model, we can either add a single feature for each complex combination, or a single feature for each atomic component. As seen in table 1, both methods improve parsing accuracy by more than one percentage point across all metrics, but splitting features into their atomic components gives a slight advantage over the single feature approach. (The difference is quantitatively small but very consistent, with a mean difference of 0.4±0.1 for the labeled attachment score of the lexicalized models.

Previous research has shown that using case and possessive features for nominals improves parsing accuracy [9]. In order to get a more fine-grained picture of the influence of different inflectional features, we have tested six different sets, where each set includes the previous one and adds some more features. The following list describes each set in relation to the previous one:

1. No inflectional features at all
2. Case and possessive inflectional features for nominals
3. Set 2 + person/number agreement inflectional features for nominals and verbs
4. Set 3 + all inflectional features for nominals
5. Set 4 + all inflectional features for verbs
6. Set 5 + all inflectional features

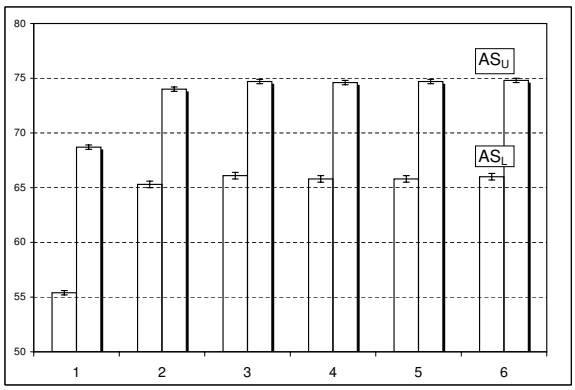

Fig. 2. Labeled and unlabeled accuracy for feature sets 1–6

The results, shown in figure 2, indicate that the parser does not suffer from sparse data even if we use the full set of inflectional features provided by the treebank. They also confirm the previous finding about the impact of case and possessive features. Besides these, the number/person agreement features available for nominals and verbs are also important inflectional features that give a significant increase in accuracy.

7 Lexicalization

Throughout the previous sections, we have seen that lexicalized models consistently give higher parsing accuracy than unlexicalized models. In order to get a more fine-grained view of the role of lexicalization, we have first studied the effect of lexicalizing IGs from individual part-of-speech categories and then from different combinations of them (see figure 3).[6] The results show that only the individual lexicalization of nouns and conjunctions provides a statistically significant improvement in AS_L and AS_U, compared to the totally unlexicalized model. Lexicalization of verbs also gives a noticeable increase in the labeled accuracy even though it is not statistically significant. A further investigation on the minor parts-of-speech of nouns[7] shows that only nouns with the minor part-of-speech "noun" has this positive effect, whereas the lexicalization of proper nouns does not improve accuracy. It can be seen from the chart of combinations that whereas lexicalization certainly improves parsing accuracy for Turkish, only the lexicalization of conjunctions and nouns has a substantial effect on the success.

Although the effect of lexicalization has been discussed in several studies recently [7, 4, 8], it is usually investigated as an all-or-nothing affair. The results for 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 suffering from sparse data), it may even be a better choice to use some kind of limited lexicalization instead of full lexicalization. The results from the previous section suggests that the same is true for morphological information.

[6] These results are not strictly comparable to those of other experiments, since the training data were divided into smaller sets (based on the POS of the next token), which reduced SVM training times without a significant decrease in accuracy.

[7] Nouns appear with six different minor parts-of-speech in the treebank: noun, proper noun, future participle, past participle, infinitive, zero. The latter four never contain lemma information.

a)

POS	AS_L	AS_U
Unlexicalized	62.6±0.3	72.2±0.2
Adjectives	62.7±0.3	72.3±0.2
Adverbs	62.8±0.3	72.4±0.2
Conjunctions	**63.5**±0.3	**73.4**±0.2
Determiners	62.7±0.3	72.2±0.2
Duplications	62.7±0.3	72.2±0.2
Interjections	62.6±0.3	72.2±0.2
Nouns	**64.0**±0.3	**73.1**±0.2
Numbers	62.6±0.3	72.2±0.2
PostPositions	62.7±0.3	72.2±0.2
Pronouns	62.7±0.3	72.2±0.2
Punctuations	62.8±0.3	72.3±0.2
Questions	62.6±0.3	72.2±0.2
Verbs	63.1±0.3	72.2±0.2

b)

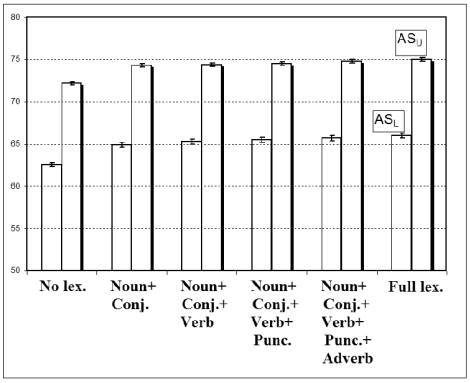

Fig. 3. Results of limited lexicalization: a)Individual b)Combination

8 Optimized Parsing Model

After combining the results of the previous three sections, we performed a final optimization of the feature model. We found that using minor parts-of-speech instead of main parts-of-speech as the values of POS features and adding one more LEX feature for the token after the next token gave a best overall performance of AS_U=76.0±0.2 AS_L=67.0±0.3, and WW_U=82.7±0.5. We also tested our parser on two different subsets of the treebank. The first subset, which is used by Eryiğit and Oflazer [9] in order to evaluate their parser (giving AS_U=73.5±1.0 and WW_U=81.2±1.0), consists of the sentences only containing projective dependencies with the heads residing on the right side of the dependents. We obtained an AS_U=78.3±0.3, AS_L=68.9±0.2 and WW_U=85.5±1.0 on the same dataset again by using ten-fold cross-validation. Using the optimized model but omitting all lexical features resulted in AS_U=76.1±0.3, AS_L=65.9±0.4 and WW_U=82.8±1.2, which shows that the improvement in accuracy cannot be attributed to lexicalization alone. The second subset is the Turkish dataset of the CoNLL-X Shared Task on Multi-lingual Dependency Parsing [25]. We obtained an AS_U=75.82 and AS_L=65.68 which are the best reported accuracies on this dataset.

9 Conclusion

Turkish is a language characterized by flexible constituent order and a very rich, agglutinative morphology. In this paper, we have shown that the accuracy achieved in parsing Turkish with a deterministic data-driven parser can be improved substantially by using inflectional groups as tokens, and by making extensive use of inflectional and lexical information in predicting the next parser action. Combining these techniques leads to the highest reported accuracy for parsing the Turkish Treebank.

However, besides showing that morpholexical information may improve parsing accuracy for languages with rich morphology and flexible word order, the experiments also reveal that the impact of both morphological and lexical information is not uniform across different linguistic categories. We believe that a more fine-grained analysis of the kind initiated in this paper may also throw light upon the apparently contradictory results reported in the literature, especially concerning the value of lexicalization for different languages.

Acknowledgments

This work is partially supported by a research grant from TUBITAK.

References

1. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania (1999)
2. Collins, M., Hajic, J., Ramshaw, L., Tillmann, C.: A statistical parser for Czech. In: Proc. of ACL-1999. (1999) 505–518
3. Bikel, D., Chiang, D.: Two statistical parsing models applied to the Chinese treebank. In: Proc. of the Second Chinese Language Processing Workshop. (2000) 1–6
4. Dubey, A., Keller, F.: Probabilistic parsing for German using sister-head dependencies. In: Proc. of ACL-2003. (2003) 96–103
5. Levy, R., Manning, C.: Is it harder to parse Chinese, or the Chinese treebank? In: Proc. of ACL-2003. (2003) 439–446
6. Corazza, A., Lavelli, A., Satta, G., Zanoli, R.: Analyzing an Italian treebank with state-of-the-art statistical parsers. In: Proc. of the Third Workshop on Treebanks and Linguistic Theories (TLT). (2004) 39–50
7. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proc. of ACL-2003. (2003) 423–430
8. Arun, A., Keller, F.: Lexicalization in crosslinguistic probabilistic parsing: The case of French. In: Proc. of ACL-2005. (2005) 302–313
9. Eryiğit, G., Oflazer, K.: Statistical dependency parsing of Turkish. In: Proc. of EACL-2006. (2006) 89–96
10. Kudo, T., Matsumoto, Y.: Japanese dependency analysis using cascaded chunking. In: Proc. of Conll-2002. (2002) 63–69
11. Yamada, H., Matsumoto, Y.: Statistical dependency analysis with support vector machines. In: Proc. of IWPT-2003. (2003) 195–206
12. Nivre, J., Scholz, M.: Deterministic dependency parsing of English text. In: Proc. of COLING-2004. (2004) 64–70
13. Nivre, J., Hall, J., Nilsson, J.: Memory-based dependency parsing. In: Proc. of Conll-2004. (2004) 49–56
14. Nivre, J., Nilsson, J.: Pseudo-projective dependency parsing. In: Proc. of the ACL-2005. (2005) 99–106
15. Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G.: Building a Turkish treebank. In Abeille, A., ed.: Building and Exploiting Syntactically-annotated Corpora. Kluwer Academic Publishers (2003)
16. Oflazer, K.: Dependency parsing with an extended finite-state approach. Computational Linguistics **29**(4) (2003)
17. Nivre, J.: An efficient algorithm for projective dependency parsing. In: Proc. of IWPT 2003. (2003) 149–160
18. Black, E., Jelinek, F., Lafferty, J.D., Magerman, D.M., Mercer, R.L., Roukos, S.: Towards history-based grammars: Using richer models for probabilistic parsing. In: Proc. of the 5th DARPA Speech and Natural Language Workshop. (1992) 31–37
19. Veenstra, J., Daelemans, W.: A memory-based alternative for connectionist shift-reduce parsing. Technical Report ILK-0012, Tilburg University (2000)
20. Nivre, J.: Inductive Dependency Parsing. Springer (2006)
21. Nivre, J.: Incrementality in deterministic dependency parsing. In Keller, F., Clark, S., Crocker, M., Steedman, M., eds.: Proc. of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together (ACL). (2004) 50–57
22. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer (1995)
23. Sagae, K., Lavie, A.: A classifier-based parser with linear run-time complexity. In: Proc. of IWPT-2005. (2005) 125–132
24. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. (2001) Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.
25. Buchholz, S., Marsi, E., Krymolowski, Y., Dubey, A., eds.: Proc. of the CoNLL-X Shared Task: Multi-lingual Dependency Parsing, New York, SIGNLL (2006)