Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, January 2012

Bootstrapping a Persian Dependency Treebank

Mojgan Seraji, Beáta Megyesi, and Joakim Nivre

Published by CSLI Publications

LiLT volume 7, issue 18 January 2012

Bootstrapping a Persian Dependency Treebank

Mojgan Seraji, Beáta Megyesi, and Joakim Nivre,
Department of Linguistics and Philology, Uppsala University

Abstract

This paper presents an ongoing project whose goal is to create a freely available dependency treebank for Persian. The data is taken from the Bijankhan corpus, which is already annotated for parts of speech, and a syntactic dependency annotation based on the Stanford Typed Dependencies is added through a bootstrapping procedure involving the open-source dependency parser MaltParser. We report preliminary parsing experiments with promising results after training the parser on a manually annotated seed data set of 215 sentences.

LiLT Volume 7, Issue 18, January 2012.
Bootstrapping a Persian Dependency Treebank.

Introduction

There are many languages with millions of speakers that still lack freely available language resources and tools to process language data. Persian belongs to the group of languages with less developed linguistically annotated data sets. There are several existing corpora for Persian but only a few of them are linguistically annotated on the word level with parts of speech and morphological features (Pouramini and Mozayani, 2007). To our knowledge there is no freely available syntactically annotated data sets and so far no treebank has been developed for Persian even though some efforts have been made to develop basic methodological principles and general syntactic annotation criteria in order to create a treebank for Persian.

Pouramini and Mozayani (2007) try to find an appropriate annotation scheme for a Persian treebank based on the syntactic characteristics of Persian. As a result, they select a scheme, focusing on annotating argument structure rather than constituent trees or dependency structures. Another attempt was introduced in Ghayoomi (2011), who proposes a rule-based approach to creating a treebank for Persian. There exists also an unpublished effort of creating a small manually annotated data set for Persian by Jon Dehdari[1] introducing word annotation based on the Leipzig Glossing Rules as well as syntactic annotation inspired by the Danish Dependency Treebank (Kromann, 2003).

Using machine learning with supervised training techniques has been shown to be a successful way to develop large syntactically annotated corpora in limited time. Moreover, data-driven parsers assist to bootstrap the process, since their performance will improve as the size of the treebank grows. In this paper we present our ongoing work on building a Persian treebank through a bootstrapping procedure using the data-driven dependency parser MaltParser. After annotating a small seed data set, we train MaltParser and parse a subset of the corpus to be manually corrected and added to the training set, and so on. The goal of using this approach is to diminish the reliance on intensive and time-consuming manual work in the annotation of training data.

In the following, we first describe the selection of data with existing part-of-speech annotation. We then describe the choice of syntactic annotation scheme and its adaptation to Persian. Before concluding, we report some preliminary parsing experiments using a manually annotated seed data set of 215 sentences.

[1] http://www.ling.ohio-state.edu/~jonsafari

Data Selection

The largest, freely available, linguistically annotated and manually validated corpus for Persian (Farsi) of today is the Bijankhan corpus that was introduced in 2004 (Bijankhan, 2004). The corpus is basically gathered from daily news and common texts, and consists of morphosyntactic and semantic annotation of nearly 2.6 million words. The original tag set contains 550 tags organized in a tree structure containing information about parts of speech with fine-grained morphological analysis and some semantic features used for subcategorization. There is a later updated version of the corpus in Unicode text format and with a reduced tag set containing 40 tags with fewer annotation layers, denoting only main part-of-speech categories with basic morphological features. This version has been used as a starting point for our treebank data set. Possibly we will investigate the use of a more fine-grained version of the part-of-speech tag set for parsing later on.

In the first stage of creating the treebank, we carried out some preprocessing of the Bijankhan corpus, mainly normalization of tokenization and sentence segmentation. In Persian, there are various typing styles where the usage of white space might be optional. In official language such as in press, mass media, and formal communication, words are typed with a so called zero-width non-joiner space or pseudo-space between free and bound morpheme boundaries, while in non-standard language, words are typed either with intervening white space or in attached form. Since all types are almost equally frequent in different types of texts, we needed to include a normalization step to take care of the separation of tokens in a consistent way to keep the word template intact. Since the Bijankhan corpus lacked sentence segmentation, we separated every sentence in the data set.

After text normalization, we extracted 10,000 sentences from the Bijankhan corpus to serve as treebank data. The data set is the same as has been used as gold standard for the evaluation of an open source part-of-speech tagger for Persian (Seraji, 2011). The selected data set includes both long and short sentences with an average of 19 words per sentence. The plan is to use a bootstrapping procedure to add syntactic annotation on top of the existing part-of-speech annotation. In the following two sections, we describe the syntactic annotation scheme as well as the parser used to bootstrap the treebank.

Syntactic Annotation

Over the years, a number of different schemes have been proposed for syntactic annotation, some based on phrase structure and others on dependency structure, some based on specific linguistic theories and others attempting to be theory-neutral (Nivre, 2008). In developing a treebank for Persian, we have opted for a dependency-based annotation, where each head and dependent relation is marked and annotated with functional categories, indicating the grammatical function of the dependent to the head. Dependency-based annotation schemes have become increasingly common in recent years, especially for languages with free or flexible word order. The Prague Dependency Treebank for Czech (Hajič et al., 2001b, Böhmová et al., 2003) has been very influential in this development, and dependency-based treebanks now exist for Arabic (Hajič et al., 2004), Basque (Aduriz et al., 2003), Danish (Kromann, 2003), Greek (Prokopidis et al., 2005), Russian (Boguslavsky et al., 2000), Slovene (Džeroski et al., 2006), and Turkish (Oflazer et al., 2003), among other languages.

Our annotation scheme is based on the Stanford Typed Dependencies (De Marneffe and Manning, 2008), which is developing into a de facto standard for English due to its widespread use in NLP applications. Although originally developed for English, the scheme is designed to be cross-linguistically valid. It has been adapted to Chinese for use with the Stanford Parser, and it has recently been used successfully to build a treebank for Finnish (Haverinen et al., 2010). Our plan is to apply the scheme to Persian and to introduce extensions and/or modifications as needed. So far, we have annotated a seed data set for bootstrapping, consisting of 215 sentences, and have introduced the following additions:

- **Accusative marker:** The relation **acc** is used for the obligatory accusative marker of direct objects.
- **Ezafe construction:** The relation **ez** is used for the enclitic particle in the *ezafe* construction, which combines nominal elements in relations of possession, qualification, etc.[2]
- **Interjection:** The relation **int** is used for interjections without a strong syntactic relation to the rest of the sentence.
- **Light verb construction:** The relation **lvc** is used for the preverbal noun, adjective or adverbial element in light verb constructions.

The syntactic relations in the extended Stanford scheme are listed with explanations in Table 1.

In order to annotate and correct our syntactic annotation in a tree structure we have used the free software tool TrEd (Pajas, 2009).[3] It

[2] The ezafe construction is similar to but not identical to the Arabic idaafa construction.

[3] TrEd is licensed under the GNU General Public License and is available at

Category	**Description**
abbrev	abbreviation modifier
acc	accusative marker
acomp	adjectival complement
advcl	adverbial clause modifier
advmod	adverbial modifier
agent	agent
amod	adjectival modifier
appos	appositional modifier
attr	attributive
aux	auxiliary
auxpass	passive auxiliary
cc	coordination
ccomp	clause complement
complm	complementizer
conj	conjunct
cop	copula
csubj	clause subject
csubjpass	clause passive subject
dep	dependent
det	determiner
dobj	direct object
ez	ezafe construction
expl	expletive
infmod	infinitival modifier
int	interjection
iobj	indirect object
lvc	light verb construction
mark	marker
mwe	multi-word expression
neg	negation modifier
nn	noun compound modifier
npadvmod	noun phrase as adverbial modifier
nsubj	nominal subject
nsubjpass	passive nominal subject
num	numeric modifier
number	element of compound number
parataxis	parataxis
partmod	participal modifier
pcomp	prepositional complement
pobj	object of a preposition
poss	possession modifier
possessive	possessive modifier
preconj	preconjunct
predet	predeterminer
prep	prepositional modifier
prepc	prepositional clause modifier
prt	phrasal verb particle
punct	punctuation
purpcl	purpose clause modifier
quantmod	quantifier phrase modifier
rcmod	relative clause modifier
ref	referent
rel	relative
root	root
tmod	temporal modifier
xcomp	open clausal complement
xsubj	controlling subject

TABLE 1 Extended Stanford Dependencies for the Persian treebank

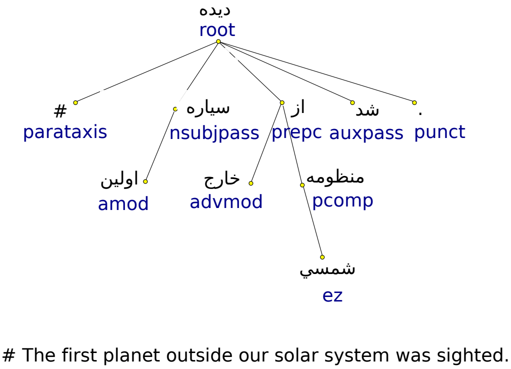

FIGURE 1 Syntactic annotation for a Persian sentence.

is a fully programmable and customizable graphical user interface for tree-like structures and was used as the main annotation tool for The Prague Dependency Treebank (Hajič et al., 2001a). Figure 1 shows the dependency annotation for a sentence from the seed data set, as visualized in TrEd.

Parsing and Bootstrapping

In order to annotate the 10,000 sentences, we will use MaltParser (Nivre et al., 2006), a language-independent system for data-driven dependency parsing, in a bootstrapping scenario. We start by training MaltParser on the seed data of 215 manually validated sentences described in the previous section and use the induced model to parse the rest of the corpus of about 10,000 sentences. We then select a subset of these sentences for manual correction, add them to the training set, retrain the parser, and reparse the remaining corpus. This process is iterated as the size of the treebank grows and the quality of the parser improves. The selection of sentences for human validation can be optimized using active learning (Hwa, 2004, Sassano and Kurohashi, 2010).

In order to get a first impression of the parsing accuracy we start from, an empirical study was carried out where MaltParser was trained on the seed data set (215 sentences) with 10-fold cross validation. The parser was used out of the box with default settings. The result of the experiment is shown in Table 2, where we see that the mean labeled attachment score (percentage of tokens with correct head and depen-

http://ufal.mff.cuni.cz/~pajas/.

Fold	LAS
1	59.3
2	55.2
3	52.5
4	59.5
5	51.1
6	57.5
7	66.6
8	53.3
9	60.6
10	51.5
Mean	56.7
StDev	4.9

TABLE 2 Parsing accuracy with 10-fold cross validation on 215 sentences. LAS = Labeled Attachment Score.

dency label) is a modest 56.7%. Nevertheless, given that the parser for each fold is trained on less than 200 sentences, we find the results promising and believe that parsing accuracy will increase substantially as the training set grows. In addition, it will be possible to optimize the feature model and other parameters of MaltParser.

In order to give a more fine-grained picture of the parsing results, Table 3 shows labeled recall and precision for the 12 most frequent dependency types (with a minimum frequency of 71 in the seed data set). As expected, we see considerable variation, with recall ranging from 37.8% for adverbial modifiers to 72.4% for the ezafe construction, and precision varying between 34.9% for nominal subjects to 82.1% for direct objects. One striking result is that direct objects are parsed much more accurately than subjects, with a drastic difference especially for precision, which is the mirror image of what we typically find for other languages (Nivre, 2006). This can probably be explained in part by the obligatory case marking for direct objects (with the accusative marker itself having reasonably high recall and precision), but it definitely also seems to be the case that the parser over-generalizes the subject relation, resulting in very low precision. It can be expected that this problem will be less severe as the size of the training set increases.

Conclusion

In this paper, we have presented a project aiming to create a dependency treebank for Persian consisting of 10,000 sentences. The data

DepRel	Freq	Rec	Prec
ez	653	72.4	61.8
prepc	409	49.9	52.7
lvc	273	65.9	54.7
conj	254	55.9	50.0
pobj	229	58.1	69.6
root	213	77.5	59.6
nsubj	203	51.7	34.9
pcomp	190	71.6	61.0
dobj	85	64.7	82.1
parataxis	75	38.7	37.7
advmod	74	37.8	39.4
acc	71	63.4	66.2

TABLE 3 Labeled recall and precision for the 12 most frequent dependency types in the seed data set.

has been selected from the Bijankhan corpus and annotation is performed semi-automatically by alternating between data-driven parsing and manual validation. The first manually annotated seed data set comprises 215 sentences, and preliminary experiments indicate that this will give a labeled parsing accuracy of about 57% for the first iteration. The goal is to improve parsing accuracy in each iteration and thereby step by step reduce the effort needed for manual validation.

Acknowledgments

We would like to thank Jon Dehdari for generously sharing his annotation scheme and a set of annotated sentences as a starting point for our annotation efforts. We are also grateful to three anonymous reviewers for useful commments on the first version of the paper. The first author has been partly supported by the Swedish Graduate School in Language Technology (GSLT).

References

Aduriz, I., M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Díaz de Ilarraza, A. Garmendia, and M. Oronoz. 2003. Construction of a Basque dependency treebank. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT)*, pages 201–204.

Bijankhan, Mahmood. 2004. The role of the corpus in writing a grammar: An introduction to a software. *Iranian Journal of Linguistics* 19.

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

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

De Marneffe, Marie-Catherine and Christopher D. Manning. 2008. Stanford typed dependencies representation. In *Proceedings of the COLING'08 Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Džeroski, Sašo, Tomaž Erjavec, Nina Ledinek, Petr Pajas, Zdenek Žabokrtsky, and Andreja Žele. 2006. Towards a Slovene dependency treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*.

Ghayoomi, Masood. 2011. Multi-token units and multi-unit tokens in developing an HPSG-based treebank for Persian. In *Proceedings of Fourth International Conference on Iranian Linguistics, ICIL4*.

Hajič, Jan, Barbora Hladká, and Petr Pajas. 2001a. Prague Dependency Treebank: Annotation structure and support. In *Proceeding of the IRCS Workshop on Linguistic Databases, Philadelphia*, pages 105–114.

Hajič, Jan, Otakar Smrž, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. 2004. Prague Arabic Dependency Treebank: Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools*.

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

Haverinen, Katri, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Filip Ginter, and Tapio Salakoski. 2010. Treebanking Finnish. In *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT)*, pages 79–90.

Hwa, Rebecca. 2004. Sample selection for statistical parsing. *Computational Linguistics* 30:253–276.

Kromann, Matthias Trautner. 2003. The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT)*, pages 217–220.

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

Nivre, Joakim. 2008. Treebanks. In A. Lüdeling and M. Kytö, eds., *Corpus Linguistics: An International Handbook*, vol. 1, pages 225–241. Walter de Gruyter.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *In Proceedings of the 5th International Conference on Language Resources and Evaluation.*

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

Pajas, Petr. 2009. TrEd tree editor. http://ufal.mff.cuni.cz/~pajas/tred.

Pouramini, Ahmad and Naser Mozayani. 2007. An annotation scheme for a Persian treebank. In *Proceedings of Computational Linguistics In the Netherlands, CLIN*.

Prokopidis, P., E. Desypri, M. Koutsombogera, H. Papageorgiou, and S. Piperidis. 2005. Theoretical and practical issues in the construction of a Greek dependency treebank. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT)*, pages 149–160.

Sassano, Manabu and Sadao Kurohashi. 2010. Using smaller constituents rather than sentences in active learning for Japanese dependency parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden*, pages 356–365.

Seraji, Mojgan. 2011. A statistical part-of-speech tagger for Persian. In *Proceedings of the 18th Nordic Conference of Computational Linguistics NODALIDA 2011. NEALT Proceedings Series*, pages 340–343.