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

Bootstrapping the Development of an HPSG-based Treebank for Persian

Masood Ghayoomi

Published by CSLI Publications

LiLT volume 7, issue 19 January 2012

Bootstrapping the Development of an HPSG-based Treebank for Persian

Masood Ghayoomi, *Freie Universität Berlin*

Abstract

In this paper, we describe an ongoing research to develop an HPSG-based treebank for Persian. To this aim, we use a bootstrapping approach for the data annotation. In the first step, a set of seed rules are defined as regular expressions in the CLaRK system. Then, the data is shallow processed with this set of rules. In the next step, a human annotator completes the annotation of sentences manually. To increase automatic annotation, we extract the manual applied rules and iteratively augment the seed rules with the rules applied frequently in the manual annotation. Our experiment in building the Persian treebank which currently contains 1000 sentences shows that the proposed method reduces human intervention from 74.05% in first iterations to 39.01% in last iterations.

LiLT Volume 7, Issue 19, January 2012.
Bootstrapping the Development of an HPSG-based Treebank for Persian.

1 Introduction

Nowadays the importance of availability or development of annotated data becomes crucial to feed linguistic investigation and also to pave the way for data driven approaches in human language technologies. Some languages like English and German are given a great amount of consideration which results to various types of data sources; while some other languages like Persian are less developed in terms of availability of annotated data. We aim to bridge the gap and start to develop a Persian treebank used for (computational) linguistic applications.

One advantage of using data-driven analyses is that a bootstrapping process can be used. Therefore, growing the size of the annotated data makes a steadily improvement on the performance. In our research, we use a bootstrapping process for developing a Persian treebank.

Generally, treebank development can be theory independent or it can be dependent on a linguistic theory. If the latter approach is chosen, then based on the selected grammar formalism various treebanks might be developed for a specific language; such as CFG (Marcus et al., 1993), DG (Rambow et al., 2002), HPSG (Oepen et al., 2002), LFG (Cahill et al., 2002), LTAG (Shen and Joshi, 2004), and CCG (Hockenamier and Steedman, 2007) treebanks developed for English.

Moreover, taking the advantage of a bootstrapping approach, this method is utilized in developing mono-lingual treebanks (Huang et al., 2000, Nivre and Megyesi, 2007, Dridan and Baldwin, 2010) and even parallel treebanks (Volk and Samuelsson, 2004).

Unfortunately, Persian is not rich in terms of availability of data sources and tools to process this language. To the best of our knowledge, there is no public available treebank for Persian; therefore, there are no tools at hand for data annotation, statistical parser, nor gold standard data for evaluation. As a result, we should create the treebank from scratch. These lacks, which make the processing of this language hard, motivated us to develop the Persian treebank and make it publicly available[2].

The HPSG (Pollard and Sag, 1994) properties and also the recent available methodologies for Persian HPSG at both theoretical level (Taghvaipour, 2005, Samvelian, 2007, Samvelian and Tseng, 2010) and system development level (Müller, 2010, Müller and Ghayoomi, 2010) motivated us to choose this formalism as the backbone of our treebank.

The structure of the paper is as follows: after introducing the data and the tool in Section 2, a bootstrapping algorithm will be described in

[2]The developed Persian treebank is accessible from this link: http://hpsg.fu-berlin.de/~ghayoomi/PTB.html

Section 3. Section 4 deals with the methodology used in the treebank development. Section 5 is devoted to the experimental results. And finally, the paper will be summarized in Section 6.

2 The Data Source and Tool

In our research, we use the freely accessible Bijankhan Corpus[3]. The register of this corpus is written texts such as magazine, book, and newspaper; and it contains around 2.5 million word tokens. Bijankhan Corpus is a sub-corpus of Peykare (Bijankhan, 2004, Bijankhan et al., 2011) which is tagged automatically based on the EAGLES guidelines (Mohseni and Minaei-bidgoli, 2010).

For our study, we selected the first 1000 sentences with the total size of 27731 word tokens from Bijankhan corpus. This small fraction of data is used to start the first phase of our treebank development. Of course, our aim is to develop a comprehensive data source for Persian.

To ease the annotation process, we benefit the CLaRK system designed to create the BulTreeBank (Simov et al., 2001). The system is an XML-based tool used for data annotation which helps the annotator by minimizing the human work. All XML documents are controlled with a DTD. The DTD plays two roles in the system: 1) it represents the sort hierarchy similar to TRALE (Meurers et al., 2002); 2) it functions as a constraint on dominance schemas in HPSG. It should be added that there are no feature structures in the system but basic properties of HPSG such as structure sharing, defining the type of dependents (subject, complement, adjunct) by using the dominance schemas (head-subject, head-complement, head-adjunct), and even binding the slashed elements via head-filler schema are simulated.

There is a deterministic finite state automaton behind CLaRK. When a rule is defined as a Regular Expression (RE) to be applied on the XML document, firstly the expression is translated to an automaton, and then to an XPath query language.
The grammar defined in CLaRK is based on Abney's cascaded grammar (Abney, 1996); i.e. the output of one rule is the input to another rule and in each rule only a fraction of data and not the whole string is annotated. Consequently, there is an order and a hierarchy on the rules.

Figure 1 represents the parse analysis of sentence (1) developed by the CLaRK system:

[3] http://ece.ut.ac.ir/dbrg/bijankhan/

(1) بورن هنرمندی است که دنیای واقعیت را با تخیل و رویا پیوند می‌زند.

born honarmand-i ast ke donyāye vāq?iyyat rā bā
Born artist-Iinflection is that world-EZ reality RĀ with
taxayyol va ro?yā peyvand mi-zan-ad.
imagination and dream link IMPF-hit-3SG
'Born is an artist who links the real world with imagination and dream.'

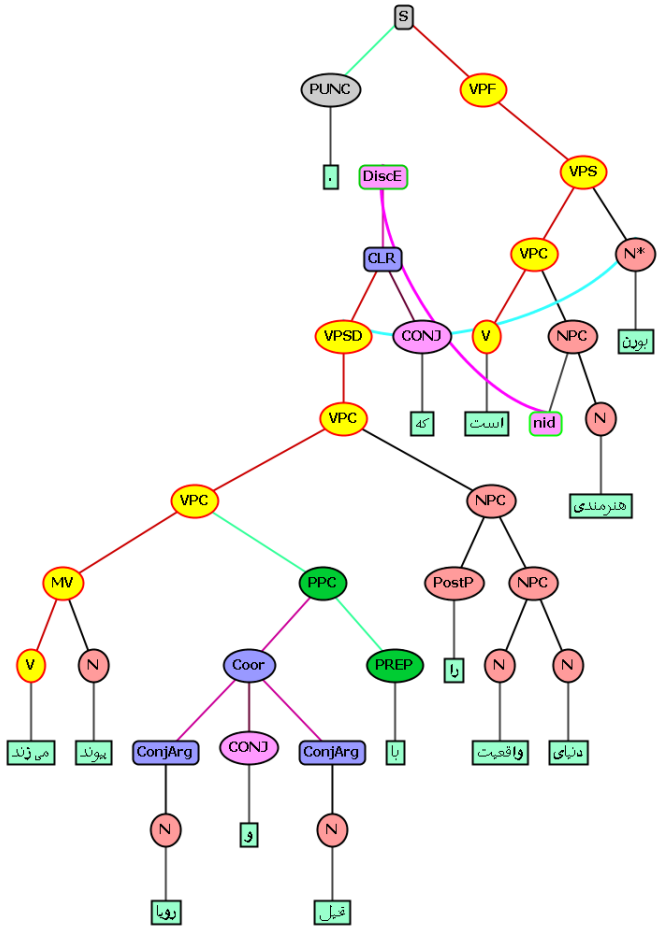

FIGURE 1 Parse analysis of sentence (1) from Bijankhan corpus

3 Bootstrapping the Treebank Development

To define rules as REs in CLaRK, we established a bootstrapping process introduced in Algorithm 1 to speed up the treebank development.

Algorithm 1 Bootstrapping Process in Treebank Development

Input: Set of Sentences from Bijankhan corpus,
Set of Seed Rules R_s defined in CLaRK
while all sentences are added to the treebank **do**
Choose N sentences S from the corpus
Use R_s to annotate S automatically
Complete the annotation of S manually
Add the annotated S to Treebank T
Extract all applied manual rules R_m from T
Select the K most frequent rules from R_m
Define the K selected rules as REs in CLaRK
Augment R_s with the K selected rules and remove them from R_m
end while

In this approach, the total 1000 sentences of our data set are preprocessed to realize multi tokens. Then, to provide the seed rules (R_s) and start the annotation process, bigrams which construct constituents are defined as REs in CLaRK. Using bigrams to define seed rules is described in Section 1.4.2. The result of applying R_s is shallow processing of the sentences. These rules are used to annotate the first 50 sentences of the data set. The result of these applied rules are checked not to over-generate. In case of over-generation, constraints are imposed on them to limit their domain to the local context.

After defining seed rules, the bootstrapping process is initialized. To this aim, the remained sentences (950 sentences) are segmented into sets containing N sentences ($N=10$). In each iteration, the N sentences are annotated automatically with seed rules. The next step is manual annotation to have complete analyses of the sentences. In this step, we extract the manual rules (R_m) applied to annotated sentences. The extracted manual rules are sorted in a descending order of their frequency. Then, the K most frequent rules ($K=5$) are selected and defined as new REs in CLaRK. R_s is augmented with the newly selected rules from R_m. Finally, the modified version of R_s is used for next iterations. The bootstrapping process continues iteratively and it terminates when the total 950 sentences are annotated completely.

TABLE 1 Conversion of original POS tag into MulText-East framework

Persian word	Transliteration	Meaning	Original tag	Converted tag
چک	ček/čak	cheque;check/whack	N,COM,SING	Ncs−−
چک	ček-e/čak-e	cheque/whack	N,COM,SING,EZ	Ncs−z
چک	ček	Czeck	N,PR,SING	Nas−−
چک	ček	Czeck	N,PR,SING,LOC	Nask−

4 Methodology of the Treebank Development

1.4.1 Defining a New Tag Format

We found two shortcomings on the format of the POS tags in the corpus used for our research. As can be seen in Table 1, the length of the tags and the position that a certain information is declared are not fixed. To solve this problem, the original tags were converted into the MulText-East[4] framework to encode the morpho-syntactic and semantic information as a single tag. In this new tag format, the length of a tag with respect to its main syntactic category is fixed and each position in the tag corresponds to one specific feature of the word[5]. If an information is unspecified, then the symbol '−' is used.

1.4.2 Steps of Data Annotation

Pre-processing Step

Processing a Persian text (written or spoken) requires a lot of preprocessing (Ghayoomi et al., 2010). In addition, the problem of multitokens (Atashgah and Bijankhan, 2009, Ghayoomi and Müller, 2011) must be resolved.

It is totally natural that the POS tag of a certain word be changed depending on a different context. Therefore, we tried to collect and store as much lexical information as possible for each lexical item from the corpus. To store this information, we defined two attributes for each word: 'gc' (global context) attribute which contains the various POS tags of a word in different contexts; and 'lc' (local context) attribute which represents the POS tag of the word in the local context.

We also lemmatized the word forms by removing the inflectional suffixes of nouns and adjectives automatically such as plural, Ezafe and indefinite clitics (if written), comparative, and superlative suffixes.

[4] http://nl.ijs.si/ME/

[5] This is the order of information presented for nouns: the main POS as noun (N); type (common (c)/proper (a)); number (singular (s)/plural (l)); semantic information mostly used to disambiguate homographs such as location (k); presence of clitics (Ezafe (z)/indefinite (y))

Since Persian has borrowed a lot of Arabic words, irregularities for plural form of nouns, and superlative and comparative forms of adjectives are unavoidable. These cases are lemmatized semi-automatically. Moreover, the infinitive forms, and the past and present stems of the verbs are defined semi-automatically.

Examples 2 and 3 display the available meta-information for the noun چک /ček/ 'Czech' and the verb دادن /dādan/ 'give'.

(2) <w gc="Nas−− ; Nask- ; Ncs−− ; Ncs−z" lc="Nask−" clitic="empty" ne_sort="loc" lemma="چک">چک</w>

(3) <w gc="Nas−− ; Ncs−− ; Ncs−z ; Vpyssht− − −" lc="Vpyssht−−−" clitic="empty" inf_form="دادن" past_stem="داد" pres_stem="ده">داد</w>[6]

To have a multi-functional data resource, we defined the types of named entities. Five named entities like 'person', 'location', 'organization', 'time', and 'other' are defined in the data set. It should be added that the name entity 'time' refers to any time expressions and dates, contrary to BBN named entity annotation[7] in which they are realized separately.

So far, we provided as much lexical information as possible for each lexical item since in HPSG a huge amount of lexical knowledge is required. This information is useful for the next steps of data annotation.

Main-processing Step

In the main processing step, we shallow process Persian sentences. To this aim, rules should be defined as REs and be ordered hierarchically in CLaRK.

For shallow processing, the frequent sequences of elements which construct constituents are labeled. To initialize the bootstrapping process in Algorithm 1, seed rules are defined in the system. To this aim, we extracted bigrams from the corpus since we want to have binary branching. The extracted bigrams were in two formats: the bigrams of the POS tags only, and the bigrams of the words with their corresponding POS tags. Based on these two sources, the most frequent sequences of elements which construct a constituent are good candidates to be defined as REs. 'به (E−) سمت (Ncs−z)' /be samt-e/ 'towards/to' or 'از (E−) سمت (Ncs−z)' /?az samt-e/ 'from' are two bigram examples which are used to define RE−1 in Table 2. This approach ensures that the defined rules are frequent enough which cover a large portion of data. To avoid over-generation, left and right contexts are taken

[6] This is the order of information available for verbs: the main POS as verb (V); polarity (negative (n)/positive (p)); auxiliary (x) or main verb (y); type (simple (s)/copulative (k)/infinitive (i)); number (singular (s)/plural (l)); person (first (o)/second (t)/third (h)); tense (present (s)/past (t)/future (u)); aspect (perfect (f)/imperfect (i)/cimperfect (q)); mood (subjunctive (u)/imperative (m)/past-participle (p)); impersonal (n)

[7] http://www.ldc.upenn.edu/Catalog/docs/LDC2005T33/BBN-Types-Subtypes.html

into consideration in defining REs and XPath queries. The defined rules are then ordered hierarchically to have a cascaded grammar.

RE−1 in Table 2 is a RE to realize compound prepositions like 'به سمت' /be samt-e/ 'towards/to' or 'از سمت' /?az samt-e/ 'from' which are constructed of a simple preposition such as 'به' /be/ 'to' or 'از' /?az/ 'from', and the noun 'سمت' /samt/ 'direction' marked with Ezafe. 'E−' is the POS tag of simple prepositions which are not marked with Ezafe. RE−2 is the RE to realize a PP such that the output of RE−1 is the input to RE−2 and the compound preposition is followed by a locative proper noun. Since the noun is a complement dependent of the compound preposition, the product is labeled as PPC.

TABLE 2 A sample of cascaded regular expressions to realize a PP

	Left RE	**RE**	**Right RE**	**Return Markup**
RE−1		<"E−">,<" سمت ">		<MP clitic="ezafe">\w</MP>
RE−2		<<MP>>,<"Nask−">		<PPC clitic="empty">\w</PPC>

Post-processing Step

In post-processing, a human annotator finishes the annotation manually based on an annotation scheme which is not described here due to lack of space. In this step, the HPSG properties described in Section 2 are taken into consideration.

Similar to (Marcus et al., 1993) and (Simov et al., 2002), only one analysis is provided for sentences with syntactic ambiguities; and this analysis relies on the intuition of the annotator.

5 Experimental Results

After shallow processing of the first 50 sentences of the data set and checking the result of the seed rules not over-generate, the annotation process is completed manually. Annotating more data via a bootstrapping process, the number of rules required to complete the annotation of each sentence manually is recorded in each iteration. Figure 2 displays the human annotation rate for each five iterations (each 50 sentences). As can be seen in the figure, as the number of the rules defined in the system grows, the human's effort to annotate the data decreases steadily which results to reduction of human intervention in developing the data source. In the first iterations of the bootstrapping process, 74.05% of the analyses were done manually; while they are reduced to 39.01% in the last iterations. Since the length of sentences vary and longer sentences are more complicated than the shorter ones, the presented results are normalized with respect to the length of the sentences.

While reducing the human annotation effort steadily, it is increased for some sets of sentences, since their complexities are increased and they require more effort for manual annotation. In further iterations, there is a gradual

decrease in human annotation again which means that the defined rules of the seen contexts are enough for analyzing.

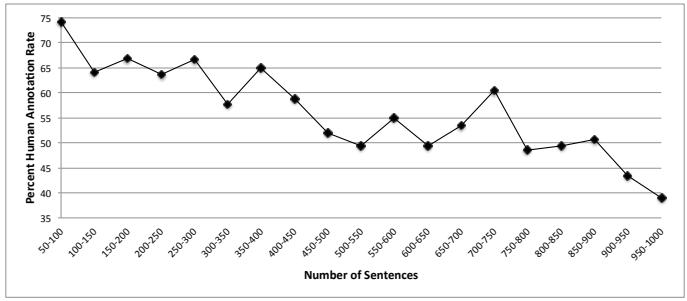

FIGURE 2 Human annotation rate in manual annotation

In Table 3, the coverage of the rules (automatic vs. manual) is reported. To annotate this small treebank for Persian, for each sentence, which is in average 27.67 words long, in average 15.84 rules that is around 57.25% of the task are applied manually and the rest automatically.

TABLE 3 Summary of the bootstrapping result in the treebank development

num. of sentences	average length of sentences (words)	average num. of automatic rules	average num. of manual rules
1000	27.67	12.19	15.84

6 Summary

This paper described a bootstrapping process to develop a treebank for Persian. Developing this annotated data was the first attempt for building a Persian treebank based on the HPSG formalism as its backbone. Defining a set of seed rules as REs in CLaRK, it was used for automatic partial annotation. Then, the annotation of the shallow processed sentences are completed manually. In the next step, the manual applied rules are extracted and iteratively the seed rules are augmented with the rules applied frequently in the manual annotation. The method used in our data annotation resulted almost 35.04% reduction of human intervention in comparison to the first iterations of data annotation. Based on the results, it can be concluded that more seen contexts result to more rule definitions and gradual reduction of human intervention for further data annotation.

Acknowledgments

The author's special gratitude goes to Stefan Müller, Kiril Simov, and Petya Osenova and the anonymous reviewers for their helpful comments in the research. However, the responsibility for the content of this study lies with the author alone. Masood Ghayoomi is funded by the German research council DFG under the contract number MU 2822/3-1.

References

Abney, Steven. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering* 2(4):337–344.

Atashgah, Masood Sharifi and Mahmood Bijankhan. 2009. Corpus-based analysis for multi-token units in Persian. In *Third Workshop on Computational Approaches to Arabic Script-based Languages [at] MT*. Ottawa, Canada.

Bijankhan, Mahmood. 2004. The role of corpora in writing grammar. *Journal of Linguistics* 19(2):48–67. Tehran: Iran University Press.

Bijankhan, Mahmood, Javad Sheykhzadegan, Mohammad Bahrani, and Masood Ghayoomi. 2011. Lessons from building a Persian written corpus: Peykare. *Language Resources and Evaluation* 45(2):143–164.

Cahill, Aoife, Mairead Mccarthy, Josef Van Genabith, and Andy Way. 2002. Automatic annotation of the Penn treebank with LFG F-structure information. In *LREC Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data*, pages 8–15.

Dridan, Rebecca and Timothy Baldwin. 2010. Unsupervised parse selection for HPSG. In *EMNLP'10*, pages 694–704.

Ghayoomi, Masood, Saeedeh Momtazi, and Mahmood Bijankhan. 2010. A study of corpus development for Persian. *International Journal on ALP* 20(1):17–33.

Ghayoomi, Masood and Stefan Müller. 2011. Multi-token units and multi-unit tokens in developing an HPSG-based treebank for Persian. In *ICIL'11*.

Hockenamier, Julia and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn treebank. *CL* .

Huang, Chu-Ren, Feng-Yi Chen, Keh-Jiann Chen, Zhao ming Gaos, and Kuang-Yu Che. 2000. Sinica treebank: Design criteria, annotation guidelines, and on-line interface. In *2nd Chinese Language Processing Workshop, ACL*.

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

Meurers, Walt Detmar, Gerald Penn, and Frank Richter. 2002. A web-based instructional platform for constraint-based grammar formalisms and parsing. In *Effective Tools and Methodologies for Teaching NLP and CL*, pages 18–25.

Mohseni, Mahdi and Behrouz Minaei-bidgoli. 2010. A Persian part-of-speech tagger based on morphological analysis. In *LREC'10*, pages 1253–1257. Valletta, Malta.

Müller, Stefan. 2010. Persian complex predicates and the limits of inheritance-based analyses. *Journal of Linguistics* 46(3):601–655.

Müller, Stefan and Masood Ghayoomi. 2010. PerGram: A TRALE implementation of an HPSG fragment of Persian. In *Int. Multiconf. on CS and IT*, pages 461–467.

Nivre, Joakim and Béata Bandmann Megyesi. 2007. Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. In *TLT 2007*.

Oepen, Stephan, Dan Flickinger, Kristina Toutanova, and Christoper D. Manning. 2002. LinGO Redwoods. A rich and dynamic treebank for HPSG. In *TLT 2002*, pages 139–149.

Pollard, Carl J. and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Rambow, Owen, Cassandre Creswell, Rachel Szekely, Harriet Taber, and Marilyn Walker. 2002. A dependency treebank for English. In *LREC'02*, pages 857–863.

Samvelian, Pollet. 2007. A (phrasal) affix analysis of the Persian Ezafe. *Journal of Linguistics* 43:605–645.

Samvelian, Pollet and Jesse Tseng. 2010. Persian object clitics and the syntax-morphology interface. In *17th Int. Conf. on HPSG*, pages 212–232.

Shen, Libin and Aravind K. Joshi. 2004. Extracting deeper information from richer resource: EM models for LTAG treebank induction. In *IJNLP*.

Simov, Kiril, Zdravko Peev, Milen Kouylekov, Alexander Simov, Marin Dimitrov, and Atanas Kiryakov. 2001. CLaRK - An XML-based system for corpora development. In *Corpus Linguistics Conference*, pages 558–560. Lancaster,UK.

Simov, Kiril, Gergana Popova, and Petya Osenova. 2002. HPSG-based syntactic treebank of Bulgarian (BulTreeBank). In *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*, pages 135–142.

Taghvaipour, Mehran A. 2005. *Persian Relative Clauses in HPSG*. Ph.D. thesis, Department of Language and Linguistics, University of Essex.

Volk, Martin and Yvonne Samuelsson. 2004. Bootstrapping parallel treebanks. In *5th Int. Workshop on Linguistically Interpreted Corpora*, pages 63–70.