Development of a Persian Syntactic Dependency Treebank

Mohammad Sadegh Rasooli
Department of Computer Science
Columbia University
New York, NY
rasooli@cs.columbia.edu

Manouchehr Kouhestani
Department of Linguistics
Tarbiat Modares University
Tehran, Iran
m.kouhestani@modares.ac.ir

Amirsaeid Moloodi
Department of Linguistics
University of Tehran
Tehran, Iran
a.moloodi@ut.ac.ir

Abstract

This paper describes the annotation process and linguistic properties of the Persian syntactic dependency treebank. The treebank consists of approximately 30,000 sentences annotated with syntactic roles in addition to morpho-syntactic features. One of the unique features of this treebank is that there are almost 4800 distinct verb lemmas in its sentences making it a valuable resource for educational goals. The treebank is constructed with a bootstrapping approach by means of available tagging and parsing tools and manually correcting the annotations. The data is splitted into standard train, development and test set in the CoNLL dependency format and is freely available to researchers.

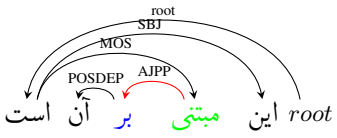
1 Introduction[1]

The process of manually annotating linguistic data from a huge amount of naturally occuring texts is a very expensive and time consuming task. Due to the recent success of machine learning methods and the rapid growth of available electronic texts, language processing tasks have been facilitated greatly. Considering the value of annotated data, a great deal of budget has been allotted to creating such data.

Among all linguistic datasets, treebanks play an important role in the natural language processing tasks especially in parsing because of its applications in tasks such as machine translation. Dependency treebanks are collections of sentences with their corresponding dependency trees. In the last decade, many dependency treebanks have been developed for a large number of languages. There are at least 29 languages for which at least one dependency treebank is available (Zeman et al., 2012). Dependency trees are much more similar to the human understanding of language and can easily represent the free word-order nature of syntactic roles in sentences (Kübler et al., 2009).

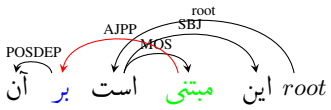
Persian is a language with about 110 million speakers all over the world (Windfuhr, 2009), yet in terms of the availability of teaching materials and annotated data for text processing, it is undoubtedly a low-resourced language. The need for more language teaching materials together with an ever-increasing need for Persian-language data processing has been the incentive for the inception of our project which has defined the development of the syntactic treebank of Persian as its ultimate aim. In this paper, we review the process of creating the Persian syntactic treebank based on dependency grammar. In this treebank, approximately 30,000 sentences from contemporary Persian-language texts are manually tokenized and annotated at morphological and syntactic levels. One valuable aspect of the treebank is its containment of near 5000 distinct verb lemmas in its sentences making it a good resource for educational goals. The dataset is developed after the creation of the syntactic valency lexicon of Persian verbs (Rasooli et al., 2011c). This treebank is developed with a bootstrapping approach by automatically building dependency trees based on the

[1]This research is done while working in Dadegan Research Group, Supreme Council of Information and Communications Technology (SCICT), Tehran, Iran. The project is fully funded by SCICT.



ʔæst ʔɑn bær mobtæni ʔin
is that on based it
V PR PP ADJ PR

(a) A simple projective dependency tree for a Persian sentence: "It is based on that"'.



ʔɑn bær ʔæst mobtæni ʔin
that on is based it
PR PP V ADJ PR

(b) A simple non-projective dependency tree for a Persian sentence: "It is based on that".

Figure 1: Examples of Persian sentences with the dependency-based syntactic trees. 1(a) and 1(b) are examples of a projective and a non-projective dependency tree, respectively. The first lines show the original words in Persian. The pronunciation and their meanings are shown in the second line and the third line respectively. In the fourth line, the part of speech (POS) tags of the words are presented. Note that the words are written from right to left (the direction of Perso-Arabic script). The dependency relations are described in Table 2. The relation is shown with an arc pointing from the head to the dependent.

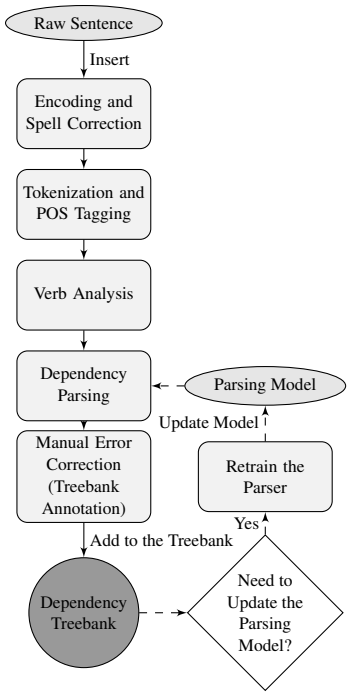


Figure 2: Diagram of bootstrapping approach in the development of the dependency treebank.

previous annotated trees. In the next step, automatic annotation is corrected manually.

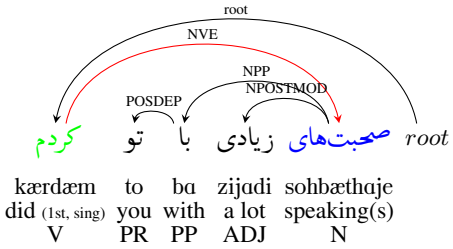
The remainder of this paper is as follows. In Section 2, we briefly review the challenges in Persian language processing. In Sections 3 and 4, the details about the annotation process, linguistic and statistical information about the data and the annotator agreement are reported. In Section 5, the conclusion and suggestions for future research are presented.

2 Persian Language Processing Challenges

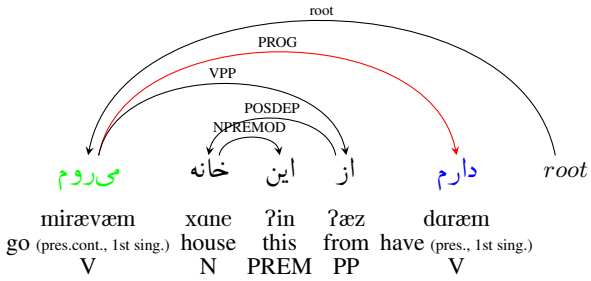
Persian is an Indo-European language that is written in Arabic script. There are lots of problems in its orthography such as encoding problems, hidden diacritics and writing standards (Kashefi et al., 2010). A number of challenges such as the free order of words, the existence of colloquial texts, the pro-drop nature of the Persian language and its complex inflections (Shamsfard, 2011) in addition to the lack of efficient annotated linguistic data have made the processing of Persian texts very difficult; e.g. there are more than 100 conjugates and 2800 declensions for some word forms in Persian (Rasooli et al., 2011b), some words in the Persian language do not have a clear word category (i.e. the lexical category "mismatch") (Karimi-Doostan, 2011a) and many compound verbs (complex predicates) can be separable (i.e. the non-verbal element may be separated from the verbal element by one or more other words) (Karimi-Doostan, 2011b).

After the development of the Bijankhan corpus (Bijankhan, 2004) with the annotation of word categories, other kinds of datasets have been created to address the need for Persian language processing. Among them, a Persian parser based on link grammar (Dehdari and Lonsdale, 2008), a computational grammar based on GPSG (Bahrani et al., 2011), syntactic treebank based on HPSG (Ghayoomi, 2012) and Uppsala dependency treebank (Seraji et al., 2012) are the efforts to satisfy the need for

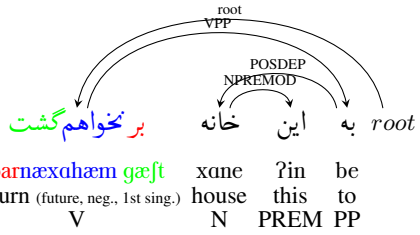
syntactic processing in the Persian language.



(a) A simple dependency tree with compound verb for a Persian sentence: "I spoke with you a lot". The *NVE* is a relation between a light verb and its nonverbal element. As shown in the tree, not only the nonverbal element is not near the light verb, but also it is inflected for plurality (i.e. speakings).



(b) A simple dependency tree for a Persian sentence with a progressive auxiliary: "I am going from this house". The *PROG* is a relation between a verb and its progressive auxiliary.



(c) A simple dependency tree for a Persian sentence with a an inflected form of a prefixed verb "I will not return to this house.". The word بر is the prefix, the word نخواهم is the auxiliary for the future and the word گشت is the main verb. Notice that the prefix is attached to the auxiliary without any space and the remaining part of the verb is separated by a space.

Figure 3: Examples of Persian sentences with the dependency-based syntactic trees. The format of the representation is the same as Figure 1.

3 Persian Dependency Treebank

3.1 Motivation

With the creation of the Virastyar spell checker software (Kashefi et al., 2010), many open-source libraries were released for Persian word processing such as POS tagging, encoding refinement, tokenization, etc. Regarding the need for syntactic analysis of Persian texts, we decided to prepare a valuable linguistic data infrastructure for Persian syntax. In the first step, there was a need for choosing from the existing theories of grammar that best suits Persian. Among grammatical theories, we decided to choose the dependency grammar. In dependency grammar, syntactic relations are shown with dependencies between the words. In computational dependency grammar, each word has one head and the head of the sentence is the dependent of an artificial root word (Kübler et al., 2009). A sample dependency tree is shown in Figure 1(a) for a Persian sentence. Note that Persian sentences are written from right to left.

There are several reasons for the preference of dependency grammar to grammars such as phrase-based structure grammars. Although in both of the representations, one can show the syntactic analysis of a sentence, dependency representation has the power to account for the free word order of many languages such as Turkish (Oflazer et al., 2003) and Czech (Hajic, 1998) and also Persian. As an example, a sample non-projective dependency tree for the Persian language is shown in Figure 1(b). The recent advances in very fast dependency parsing models (e.g. (Nivre, 2009; Bohnet and Nivre, 2012)), has made the syntactic processing task very popular in the recent decade.

In the Persian language, in addition to the abundance of crossings of the arcs, another problem occurs with compound verbs and verbs in the progressive aspect: compound and progressive verbs are multi-word expressions that may be separated depending on the context. Persian compound verbs consist of a light verb and a non-verbal element and the non-verbal element can be a noun, an adjective (in rare cases) or a sequence of a preposition and a noun (Dabir-Moghaddam, 1997). In addition, the nonverbal elements can also be inflected. The distance between the nonverbal element and the light

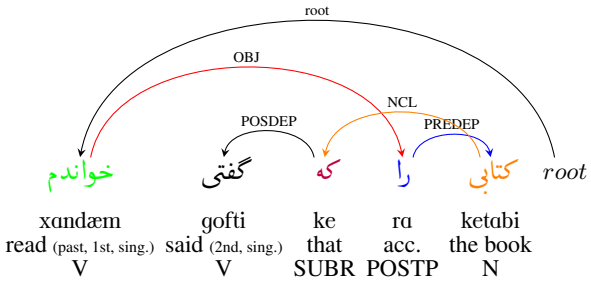
verb on the one hand and the possibility of the nonverbal element being inflected on the other hand have made the task of compound verb identification very difficult. For example, in Bijankhan (Peykare) corpus (Bijankhan et al., 2011), approximately 9% of nonverbal elements of compound verbs are placed away from the light verb for the compound verbs with the light verb کردن /kærdæn/ (to do) (Rasooli et al., 2011a). A group of Persian progressive verbs are composed of two words, the first being the simple past or the simple present form derived from the infinitive داشتن /dɑʃtæn/ (to have) and the second being the past continuous or the present continuous form of the main verb. The first verb (an auxiliary) agrees with the second in number and person. The problem is that the progressive auxiliary can be away from the main verb. The sample trees with compound verbs and progressive auxiliary verbs are shown in Figures 3(a) and 3(b) respectively.

3.2 Representation and Dependency Relation

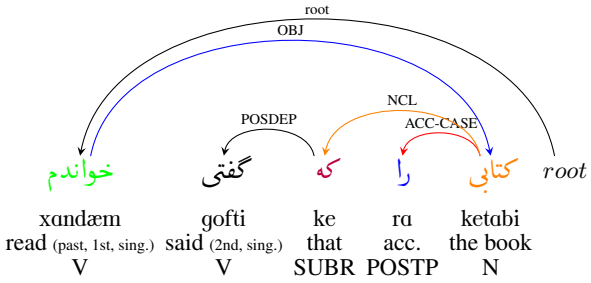
In this treebank, we followed the format of the CoNLL tab-separated format for dependency parsing (Buchholz and Marsi, 2006). In addition to the lemma, we annotated part of speech tags (both coarse and fine grained) and person, number and tense-mood-aspect (only for verbs) of words in sentences. The details of the part of speech tags and other morphosyntactic features and dependency relations are shown in Tables 1 and 2, respectively. The part of speech tag set in this treebank is not the same as that of Bijankhan (Peykare) corpus (Bijankhan et al., 2011) and it is essential to convert the tagset in Peykare corpus to the tagset in this treebank, in order to use both datasets[2]. We also tried to use the writing standard of the Academy of Persian Language and Literature except for the cases where for a word there were several standards all of which were used in Persian written texts (e.g. آنچه and آن‌چه /ʔɑnʧe/ (whatever)).

We also prepared two representations for objects accompanied by the accusative case marker. In the first representation (done manually), we assume the accusative case marker را /rɑ/ as the head of the two-word sequence object plus rɑ. The second representation, that is the automatic conversion of the first, is the reverse order of the first one in which the accusative case marker is the dependent of the direct object and the direct object is considered as the head of the aforementioned sequence. In the first representation, objects are much easier to find by the parser (because of the uniqueness of the accusative case marker in Persian and less distance of it from the verb as its head) but it may increase the number of non-projective arcs to the syntactic tree. We prepared both of the representations in two separate data packs. A sample comparison between the two structures is shown in Figure 4.

In the treebank, all words are single word forms (without spaces). There is only one exception for simple verb inflections where even multi-word tokens of simple verbs are shown as only one unit. The reason is that for many cases such as the case of inflections for prefixed verbs it is more straightforward to analyze the whole part instead of analyzing each



(a) "I read the book that you mentioned.".



(b) "I read the book that you mentioned."

Figure 4: A sample sentence with two kinds of representations of object-verb relation. The first one is done manually and the second automatically by converting the dependencies in the first representation.

[2]It is important to note that the conversion between the coarse-grained POS tags is straightforward and does not need any special effort.

Morphosyntactic features in the Persian dependency treebank				
CPOS	**FPOS**	**Person**	**Number**	**TMA**
ADJ (adjective)	AJP (positive) AJCM (comparitive) AJSUP (superlative)			
ADR (address term)	PRADR (pre-noun) POSADR (post-noun)			
ADV (adverb)	SADV (genuine)			
CONJ (coordinating conjunction)	CONJ (conjunction)			
IDEN (title)	IDEN (title)			
N (noun)	ANM (animate) IANM (inanimate)		SING (singular) PLUR (plural)	
PART (particle)	PART (particle)			
POSNUM (post-noun modifier)	POSNUM (post-noun modifier)			
POSTP (postposition)	POSTP (postposition)			
PR (pronoun)	SEPER (separate personal) JOPER (enclitic personal) DEMON (demonstrative) INTG (interogative) CREFX (common reflexive) UCREFX (noncommon reflexive) RECPR (reciprocal)	1 2 3	SING (singular) PLUR (plural)	
PREM (pre-modifier)	EXAJ (exclamatory) QUAJ (interrogative) DEMAJ (demonstrative) AMBAJ (ambiguous)			
PRENUM (pre-noun numeral)	PRENUM (pre-noun numeral)			
PREP (preposition)	PREP (preposition)			
PSUS (pseudo-sentence)	PSUS (pseudo-sentence)			
PUNC (punctuation)	PUNC (punctuation)			
V (verb)	ACT (active) PAS (passive) MOD (modal)	1 2 3	SING (singular) PLUR (plural)	See Table 3
SUBR (subordinating clause)	SUBR (subordinating clause)			

Table 1: Morphosyntactic features in the Persian dependency treebank. Empty cells indicate that the mentioned feature is not present for the POS. *TMA* stands for *Tense/Mood/Aspect*, *CPOS* for *Coarse grained POS* and *FPOS* for *Fine grained POS*. There is also another feature for representing the typographical connectedness of words that are separated into two or more tokens with the values ISO (isolated word), NXT (attached to the next token) and PRV (attached to the previous token).

part separately[3]. In Table 3, possible types of the Persian verb inflections are shown. As seen in Table 3, 6 forms of 14 inflection types of Persian verbs are multi-word tokens and for passive verbs they may be composed of more words than their active counterparts (since for passive verbs an auxiliary form derived from the infinitive شدن /ʃodæn/ is used). In Figure 3(c), a sample tree with a multi-token prefixed verb is shown. As shown in the case of colored tokens, it seems more beneficial to put all morphemes of the word together before parsing. Furthermore, with the available Persian verb analyzer it is very easy to first preprocess the verbs[4].

[3]In (Seraji et al., 2012), multi-token verbs are considered as separate words.

[4]If it is needed to respect the exact format of CoNLL, spaces between the verb tokes should be replaced by a character such as underscore. Regarding the special fine-grained morphological tags for the verb such as tense-mood-aspect, it is also straightforward to separate all of the multi-word verbs and add new dependency relations between their words.

Abbreviation	Description	Abbreviation	Description
ACL	Complement Clause of Adjective	ADV	Adverb
ADVC	Adverbial Complement of Verb	AJCONJ	Conjunction of Adjective
AJPP	Prepositional Complement of Adjective	AJUCL	Adjunct Clause
APOSTMOD	Adjective Post-Modifer	APP	Apposition
APREMOD	Adjective Pre-Modifier	AVCONJ	Conjunction of Adverb
COMPPP	Comparative Preposition	ENC	Enclitic Non-Verbal Element
LVP	Light Verb Particle	MESU	Measure
MOS	Mosnad	MOZ	Ezafe Dependent
NADV	Adverb of Noun	NCL	Clause of Noun
NCONJ	Conjunction of Noun	NE	Non-Verbal Element of Infinitive
NEZ	Ezafe Complement of Adjective	NPOSTMOD	Post-Modifer of Noun
NPP	Preposition of Noun	NPREMOD	Pre-Modifier of Noun
NPRT	Particle of Infinitive	NVE	Non-Verbal Element
OBJ	Object	OBJ2	Second Object
PARCL	Participle Clause	PART	Interrogative Particle
PCONJ	Conjunction of Preposition	POSDEP	Post-Dependent
PRD	Predicate	PREDEP	Pre-Dependent
PROG	Progressive Auxiliary	PUNC	Punctuation Mark
ROOT	Sentence Root	SBJ	Subject
TAM	Tamiz	VCL	Complement Clause of Verb
VCONJ	Conjunction of Verb	VPP	Prepositional Complement of Verb
VPRT	Verb Particle	ACC-CASE	Accusative Case Marker (2nd. Rep.)

Table 2: Dependency relations in the Persian dependency treebank

Tense/Aspect/Mood	Abbreviation	Examples خوردن xordæn: to eat, 1st, sing.
Imperative	HA	بخور /boxor/
Indicative Future	AY	خواهم خورد /xɑhæm xord/
Indicative Imperfective Perfect	GNES	می‌خورده‌ام /mixorde?æm/
Indicative Imperfective Pluperfect	GBES	می‌خورده بودم /mixorde budæm/
Indicative Imperfective Preterite	GES	می‌خوردم /mixordæm/
Indicative Perfect	GN	خورده‌ام /xorde?æm/
Indicative Pluperfect	GB	خورده بودم /xorde budæm/
Indicative Present	H	می‌خورم /mixoræm /
Indicative Preterite	GS	خوردم /xordæm/
Subjunctive Imperfective Pluperfect	GBESE	می‌خورده بوده باشم /mixorde bude bɑʃæm/
Subjunctive Imperfective Preterite	GESEL	می‌خورده باشم /mixorde bɑʃæm/
Subjunctive Pluperfect	GBEL	خورده بوده باشم /xorde bude bɑʃæm/
Subjunctive Present	HEL	بخورم /boxoræm/
Subjunctive Preterite	GEL	خورده باشم /xorde bɑʃæm/

Table 3: Tense/Mood/Aspect Types in Persian verbs

3.3 Annotation Process

The annotation process consists of several consecutive steps. In Figure 2, a summary of the bootstrapping approach in the annotation process is shown. At first, a collection of independent sentences have been collected randomly from the web. For the first 5000 sentences, we crawled Persian news texts and randomly sampled the sentences. For the remaining sentences, we first listed the absent verb lemmas in the 5000 sentences based on the verb list ex-

tracted from the valency lexicon of Persian verbs (Rasooli et al., 2011c) and collected random sentences that included the absent verb lemmas in their words. We listed all possible inflections and per each verb lemma, sampled at most 8 sentences from the web. These sentences had to contain at least one present tense, one past tense, one passive voice and one future tense inflection unless we could not find them and were obliged to reduce the number. The sentences were not shortened and were kept with their original length and words. Finally, we manually removed sentences containing colloquial words. However, we did not remove loan words or cases of code-switching between latin-script words and Persian words in the sentences. The raw sentences were fed to the encoding and spell checking module. After spell correction, all sentences were tokenized and tagged with part of speech tags. All of the word processing steps were done using Virastyar library (Kashefi et al., 2010). After tokenization and POS tagging, the tokenized sentences were fed to the Persian verb analyzing tool (Rasooli et al., 2011a). In the next step, the preprocessed sentences were given to the dependency parser. We used MST parser (McDonald et al., 2005) for parsing the sentences.

In the final step, annotators corrected the errors of tokenization, POS tagging and parsing. In about every one to two weeks, the parser model was updated by training on the new version of the treebank. This process lasted 9 months and the number of annotators increased by time to speed up the process. In the first 6 months, we used 8 annotators and for the next 5 months, we hired 6 more annotators to speed up the process. The annotators and linguistic experts consisted of 1 PhD graduate (linguistics), 4 PhD candidates (linguistics), and 9 MA graduates or graduate students (7 linguistics, 1 Persian language and literature and 1 computational linguistics). All of the annotators were native Persian speakers.

After finalizing the annotation of all raw sentences, we applied a rule-based potential error finder to find the potentially erroneous sentences. The rules were gradually collected in the process of the annotation by the annotators. All the potentially erroneous sentences were given to the annotators to be checked for potential errors. In Section 4.1, the statistics about the changes after the correction is reported. One of the main reasons for the double checking phase in the process is that based on our manual investigations of the annotations, we found some inevitable mistakes by annotators that could be solved by manual rules. Mistakes such as scrolling the drop-down list unintentionally and changing the part of speech tag or dependency relation and mistakes caused by tiredness and lack of concentration in addition to some of the changes of the linguistic conventions in the annotation. Since all cases of dependency relations in this treebank may be usually either a left-branching relation or a right-branching one and most of the relations are restricted to certain types of parts of speech, it is easy to capture the potential errors in the annotations based on the rules mentioned and to keep track of the changes in the linguistic conventions by searching the cues for those conventions (most of the changed conventions were made to very rare relations in the syntactic structure).

In (Dligach and Palmer, 2011), it is concluded that although doubly annotated corpora are more reliable, annotating more sentences only once is more beneficial; i.e. annotating each sentence only once is less time-consuming and more cost-effective. We annotated all the sentences only once (with an additional checking phase) except for the 5% of the sentences in order to estimate the quality of our linguistic conventions and agreement among the annotators. The statistics about the annotators agreement is reported in Section 4.1.

4 Statistics of the Treebank

Finally, 29,982 sentences were manually annotated. The details about the statistics is shown in Table 4. It is worth mentioning that 39.24% of the words in the treebank are tagged as noun, 12.62% as verb, 11.64% as preposition and 7.39% as adjective. The most frequent dependency relations are post-dependent (15.08%) and Ezafeh construction (10.17%). As shown in Table 5, the number of non-projective arcs in the second representation is a little bit less than the first. As mentioned earlier, the main reason is the dependencies between the direct object and words after the accusative case marker such as the example in Figure 4. The change percentage after the correction of the potential errors is shown in Table 6. It seems that the rules for finding the poten-

Number of Sentences	29,982
Number of Words	498,081
Average Sentence Length	16.61
Number of Distinct Words	37,618
Number of Distinct Lemmas	22,064
Number of Verbs	60,579
Number of Verb Lemmas	4,782
Average Frequency of Verb Lemmas	12.67

Table 4: Statistics about the frequency of words in the Persian dependency treebank.

# Non-Projective	1st Rep.	2nd Rep.
Number of Arcs	12281	8512
Percent of Arcs	2.47	1.71
Number of Sentences	6574	4838
Percent of Sentences	21.93	16.14

Table 5: Statistics about non-projective relations in the Persian dependency treebank for both of the representations.

tial errors were useful for correcting the errors.

4.1 Annotators Agreement

The statistics about the agreement among the annotators is shown in Table 7. We can also use the Kappa (Cohen, 1960) to measure the quality of the annotation based on the agreement among the annotators (pr(a) in Eq. 1) and the expected agreement or probability of chance (pr(e) in Eq. 1). If we consider the accuracy of the parser on the raw text without gold POS tags (approximately 75% labeled and 80% unlabeled accuracy based on our experience during the bootstrapping) and the POS tagger that we used during the annotation process (approximately 94%) as the probability of chance, we see that for all of the tasks in Table 7, the quality of the annotaion is more than 0.81 and is considered as almost perfect according to (Landis and Koch, 1977).

$$k = \frac{pr(a) - pr(e)}{1 - pr(e)} \quad (1)$$

5 Conclusion

As mentioned earlier, Persian is a language with its own challenges. We tried to overcome some of those challenges by preparing valuable linguistic datasets[5]. In addition to the preparation of the treebank, we prepared some useful desktop and web-based tools for searching in the dataset, obtaining statistics and viewing syntactic structures graphically. We hope to report more details about the linguistic aspects and the findings of the project in addition to our detailed experiments on the parsing task in future publications. We believe that this treebank is just the very first step to satisfy the need for Persian language processing. Our future aim is to add a semantic level to the annotation.

Changes to Unlabeled Relations	4.91%
Changes to Labeled Relations	6.29%
Changes to POS Tags	4.23%

Table 6: Statistics about changes in the treebank after the manual correction of the potential errors.

Unlabeled Relations	97.06%
Labeled Relations	95.32%
POS Tags	98.93%

Table 7: Statistics about agreements among the annotators.

Acknowledgments

The project is funded by Iran Supreme Council of Information and Communication Technology (SCICT). We really appreciate the linguists who helped us in annotating: Farzaneh Bakhtiary, Parinaz Dadras, Maryam Faal-Hamedanchi, Saeedeh Ghadrdoost-Nakhchi, Mostafa Mahdavi, Azadeh Mirzaei, Sahar Oulapoor, Neda Poormorteza-Khameneh, Morteza Rezaei-Sharifabadi, Sude Resalatpoo, Akram Shafie, and Salimeh Zamani; and the programmers who helped us in the process of the development of the treebank: Seyed Mahdi Hoseini, Alireza Noorian, Yasser Souri, and Mohsen Hossein-Alizadeh; and also our colleagues who helped us find linguistic sources from the web: Azadeh Abbasi Abyaneh, Shima Zamanpoor, Narmin Ghaderi, Houra Nouri and Seyedeh Maneli Hashemian; and other colleagues especially Mahdi Behniafar. We thank Nizar Habash for his support of this paper and Weiwei Guo and three anonymous reviewers for their useful comments on the paper.

[5] A comprehensive description of the syntactic relations and morphosyntactic features is reported in the treebank official report (Dadegan Research Group, 2012) in the treebank package both in Persian and English.

References

Mohammad Bahrani, Hossein Sameti, and Mehdi Hafezi Manshadi. 2011. A computational grammar for Persian based on GPSG. *Language Resources and Evaluation*, 45(4):387–408.

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

Mahmood Bijankhan. 2004. The role of the corpus in writing a grammar: An introduction to a software. *Iranian Journal of Linguistics*, 19(2).

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL*, pages 1455–1465.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceeding of the Tenth Conference on Computational Natural Language Learning (CoNLL)*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Mohammad Dabir-Moghaddam. 1997. Compound verbs in Persian. *Studies in the Linguistic Sciences*, 27(2):25–59.

Dadegan Research Group. 2012. *Persian Dependency Treebank, Annotation manual and user guide*. Supreme Council of Information and Communication Technology (SCICT), Tehran, Iran.

Jon Dehdari and Deryle Lonsdale. 2008. A link grammar parser for Persian. *Aspects of Iranian Linguistics*, 1.

Dmitriy Dligach and Martha Palmer. 2011. Reducing the need for double annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 65–69.

Masood Ghayoomi. 2012. Bootstrapping the development of an HPSG-based treebank for Persian. *Linguistic Issues in Language Technology*, 7(1).

Jan Hajic. 1998. Building a syntactically annotated corpus: The Prague dependency treebank. *Issues of valency and meaning*, pages 106–132.

Gholamhossein Karimi-Doostan. 2011a. Lexical categories in Persian. *Lingua*, 121(2):207–220.

Gholamhossein Karimi-Doostan. 2011b. Separability of light verb constructions in Persian. *Studia Linguistica*, 65(1):70–95.

Omid Kashefi, Mitra Nasri, and Kamyar Kanani. 2010. *Automatic Spell Checking in Persian Language*. Supreme Council of Information and Communication Technology (SCICT), Tehran, Iran.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 91–98, Sydney, Australia.

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

Kemal Oflazer, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. *Treebanks*, pages 261–277.

Mohammad Sadegh Rasooli, Heshaam Faili, and Behrouz Minaei-Bidgoli. 2011a. Unsupervised identification of Persian compound verbs. In *Proceedings of the Mexican international conference on artificial intelligence (MICAI)*, pages 394–406, Puebla, Mexico.

Mohammad Sadegh Rasooli, Omid Kashefi, and Behrouz Minaei-Bidgoli. 2011b. Effect of adaptive spell checking in Persian. In *7th International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE)*, pages 161–164, Tokushima, Japan.

Mohammad Sadegh Rasooli, Amirsaeid Moloodi, Manouchehr Kouhestani, and Behrouz Minaei-Bidgoli. 2011c. A syntactic valency lexicon for Persian verbs: The first steps towards Persian dependency treebank. In *5th Language & Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 227–231, Poznań, Poland.

Mojgan Seraji, Beáta Magyesi, and Joakim Nivre. 2012. Bootstrapping a Persian dependency treebank. *Linguistic Issues in Language Technology*, 7(1).

Mehrnoosh Shamsfard. 2011. Challenges and open problems in Persian text processing. In *5th Language & Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 65–69, Poznań, Poland.

Gernot Windfuhr. 2009. *The Iranian Languages*. Routledge.

Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zednéck Žabokrtskỳ, and Jan Hajič. 2012. Hamledt: To parse or not to parse. In *Proceedings of the Eighth Conference on International Language Resources and Evaluation (LREC'12), Istanbul, Turkey*.