A Basic Language Resource Kit for Persian

Mojgan Seraji, Beáta Megyesi, Joakim Nivre

Uppsala University, Department of Linguistics and Philology
E-mail: `firstname.lastname@lingfil.uu.se`

Abstract

Persian with its about 100,000,000 speakers in the world belongs to the group of languages with less developed linguistically annotated resources and tools. The few existing resources and tools are neither open source nor freely available. Thus, our goal is to develop open source resources such as corpora and treebanks, and tools for data-driven linguistic analysis of Persian. We do this by exploring the reusability of existing resources and adapting state-of-the-art methods for the linguistic annotation. We present fully functional tools for text normalization, sentence segmentation, tokenization, part-of-speech tagging, and parsing. As for resources, we describe the Uppsala PErsian Corpus (UPEC) which is a modified version of the Bijankhan corpus with additional sentence segmentation and consistent tokenization modified for more appropriate syntactic annotation. The corpus consists of 2,782,109 tokens and is annotated with parts of speech and morphological features. A treebank is derived from UPEC with an annotation scheme based on Stanford Typed Dependencies and is planned to consist of 10,000 sentences of which 215 have already been annotated.

Keywords: BLARK for Persian, PoS tagged corpus, Persian treebank

1. Introduction

Having a standard and publicly available basic language resource kit (BLARK) containing resources such as dictionaries, lexicons, general and specialized corpora as well as tools for processing language data is necessary and much needed for language studies, language technology applications and language teaching. The number of existing resources and tools varies greatly from language to language, and only a few has a large number of resources and tools for processing language data. Compared to languages like English, Persian despite its large number of speakers in the world belongs to the group of languages with less developed linguistically annotated data sets and tools, and only a few of them are freely available.

In this paper we present various BLARK components for Persian, all freely available and developed by re-using existing resources and applying data-driven state-of-the-art methods to the linguistic analysis of Persian. We report our work on text processing tools, namely tools for text normalization, sentence segmentation, tokenization, part-of-speech tagging, and parsing. In addition, we present the Uppsala PErsian Corpus (UPEC) and our efforts on developing the Uppsala PErsian Dependency Treebank (UPEDT).

This paper is divided into the following sections. In Section 2, we start with a short description of Persian and its main characteristics, as well as a discussion of problems concerning text processing. Section 3 briefly reviews already existing resources and tools for Persian, and in Section 4 we introduce our work on the BLARK components. Finally, in Section 5 we conclude our paper.

2. Persian

Persian belongs to the Indo-Iranian branch of the Indo-European family. There are three varieties of the language, Western Persian referred to as Parsi or Farsi (Parsi has been arabicized to Farsi due to the lack of the phoneme /p/ in Arabic) spoken in Iran, Eastern Persian referred to as Dari spoken in Afghanistan, and Tajiki spoken in Tajikistan and Uzbekistan. Persian has also had a strong influence on neighboring languages such as Turkish, Armenian, Azerbaijanian, Urdu, Pashto, and Punjabi.

We start with a description of some important characteristics of Persian orthography that need to be considered when developing language resources and tools. Then we continue with an overview of the morphological and syntactic structure of Persian.

2.1. Orthography

The Persian writing system is based on the Arabic alphabet with 28 letters and four additional letters: گ، ژ، چ، پ, which are the sounds of /p/, /ʧ/, /ʒ/, /g/. However, similar to all Indo-European languages, it does not follow the Arabic consonantal root system which characterizes the Semitic languages. Persian uses cursive script, i.e., characters have different forms depending on their position in the word. All characters can be divided into two groups on the basis of how they connect to other characters: "dual-joining" and "right-joining". In dual-joining, characters have two distinct shapes depending on their position in the word: initial or medial, and final or isolated respectively. However, three characters in this group, namely ع /'eyn/, غ /qeyn/, and ه /he/ (he-ye do-češm) appear in four distinct shapes. There are also two characters in this group, ط /tâ/ and ظ /zâ/, which have only one shape irrespective of their position in the word. Table 1 displays the initial, medial, final, and isolated forms of the characters in the dual-joining group. The right-joining characters do not accept any connection from their left hand side and have only one shape without any distinctive initial, medial, final, or isolated forms. These characters are illustrated in Table 2.

Phonological and lexical ambiguity is as common as in other languages. Lexical ambiguity in Persian occurs especially when short vowels are left out from the token, remaining only a string of consonants. Short vowels are used as phonetic guides to identify the meaning of consonantal words with multiple senses. Since diacritic signs are

Isolated	Final	Medial	Initial	Name
ب	ب	ﺑ	ﺑ	be
پ	پ	ﭘ	ﭘ	pe
ت	ت	ﺗ	ﺗ	te
ث	ث	ﺛ	ﺛ	se
ج	ج	ﺟ	ﺟ	jim
چ	چ	ﭼ	ﭼ	če
ح	ح	ﺣ	ﺣ	he-ye jimi (literally jim-like he)
خ	خ	ﺧ	ﺧ	khe
س	س	ﺳ	ﺳ	sin
ش	ش	ﺷ	ﺷ	shin
ص	ص	ﺻ	ﺻ	sâd
ض	ض	ﺿ	ﺿ	zâd
ط	ط	ﻃ	ﻃ	tâ
ظ	ظ	ﻇ	ﻇ	zâ
ع	ع	ﻌ	ﻋ	’eyn
غ	غ	ﻐ	ﻏ	qeyn
ف	ف	ﻓ	ﻓ	fe
ق	ق	ﻗ	ﻗ	qâf
ک	ک	ﮐ	ﮐ	kâf
گ	گ	ﮔ	ﮔ	gâf
ل	ل	ﻟ	ﻟ	lâm
م	م	ﻣ	ﻣ	mim
ن	ن	ﻧ	ﻧ	nun
ه	ﻪ	ﻬ	ﻫ	he-ye do-češm (literally two-eyed he)
ی	ی	ﯾ	ﯾ	ye

Table 1: Dual-joining Persian characters

Isolated	Final	Medial	Initial	Name
ا	ا	ا	ا	alef
د	د	د	د	dâl
ذ	ذ	ذ	ذ	zâl
ر	ر	ر	ر	re
ز	ز	ز	ز	ze
ژ	ژ	ژ	ژ	ẑe
و	و	و	و	vâv

Table 2: Right-joining Persian characters

only used for beginner learners and it is expected that adult native speakers have already developed cognitive strategies for efficient linguistic performance, those are unwritten in texts. The absence of the short vowels from written texts create lexical ambiguity for words such as: مرد "man" or "died", شکر "sugar" or "thank", and کلفت "thick" or "maid".

کتاب‌خانه ها ی	کتاب‌خانه‌های
کتاب‌خانه های	کتاب خانه های
کتابخانه های	کتاب خانه‌ها ی
کتابخانه ها ی	کتاب خانه ها ی
کتابخانه‌ها ی	کتاب‌خانه‌ها ی
کتابخانه‌های	کتاب خانه‌های

Table 3: 12 different types of writing the plural and definite form of the compound word "کتاب‌خانه‌های" (the libraries).

A phoneme may be represented by various letters which may cause disparities in letter substitution, especially in case of transliterating foreign words when deciding a proper grapheme for a desired phoneme. There are various letters for the same phoneme such as the phoneme /t/ represented by the two letters, ت and ط, the phoneme /h/ by ح and ه, /s/ by the three letters ث, س, and ص, and finally the phoneme /z/ by the four letters ذ, ز, ض, and ظ.

We can also find two different types of space characters with different Unicode encodings, i.e., the white space and the zero-width non-joiner (ZWNJ). The white space in Persian designates word boundaries as in many languages while the ZWNJ marks boundaries inside a word. The ZWNJ, also known as pseudo-space, zero-space or virtual space, is a non-printing character in computerized typesetting of some cursive script placed between two characters to be printed in the final and initial forms to each other. The ZWNJ keeps the word forms intact and close together without being attached to each other. Considering the wide range of varieties of typing styles and the optionality of shifting between white space and ZWNJ, one word may be written in various ways in a text. Compound words and inflectional affixes are highly affected and can be typed either as attached to (when ignoring both spaces thereby losing the internal word boundaries) or detached (when using white space instead of ZWNJ) from their adjacent word, which in both cases raise issues in text processing. Inflectional suffixes may follow compound words, as the word کتاب‌خانه(library) followed by the plural suffix ها- /-hâ/ together with the ezafe particle ی- /-ye/ and might appear in 12 forms as shown in Table 3.

Although Persian and Arabic share almost the same character encodings, apart from the four extra letters in Persian, there are a few stylistic disparities in the two letters ی (ye) and ک (kaf) with different Unicode characters for Persian and Arabic. Due to the fact that the various operating systems have traditionally Arabic Unicode characters as their default for Persian, a huge number of texts use Arabic letters and digits instead of Persian. Despite the existence of Unicode characters provided for Persian digits, the Western digits on Persian keyboards are still used. People have basically two options when choosing character encoding, applying either Arabic encoding and Arabic digits or Persian encoding and Western digits. Hence, as a consequence, all texts are a mixture of different encoding systems which need to be taken care of in natural language processing tasks.

2.2. Persian Morphology

Persian has a rich morphology dominated by an affixal system. There is no grammatical gender. Verbs can express tense, aspect, and mood, and agree in person and number with the subjects. Possessiveness is expressed by the genitive clitic ezafe -*e*[1] or by pronominal genitive clitics (-am, -at, -aš, -emân, -etân, -ešân) which are the bound forms of full personal pronouns. There are several plural markers -*hâ*, -*ân*, (with variants -*gân* and -*yân*), and some Arabic plural markers -*ât*, -*in*, *un*, attaching only to Arabic loanwords. Arabic broken plural also exists in Persian that follows Arabic template morphology and is directly inherited with nouns borrowed from Arabic. Adjectives vary in the suffixes -تر /-tar/ and -ترین -/tarin/ for comparative and superlative forms respectively. Pronouns often appear as pronominal clitics (م- /-am/ 1sg, ت- /-at/ 2sg, ش- /-aš/ 3sg, مان- /-emân/ 1pl, تان- /-etân/ 2pl, شان- /-ešân/ 3pl) which are the bound forms of personal pronouns. Pronominal clitics can be attached to nouns, adjectives/adverbs, prepositions, and verbs. Their function is as possessive genitive کتابش (his/her book), partitive genetive چندتاش (some of that), object of a preposition ازش (of him/her), non-canonical subject بدش می‌آید (he/she dislike), and direct object زدندش (they hit him/her). For instance زدندش (they hit him/her) consists of the verb stem زد (hit), the personal agreement marker ند- (they) and the pronominal clitic ش- (him/her). The native word formation in Persian is based on combining verbal stems, adjectives, and nouns with affixes.

2.3. Persian Syntax

The Persian word order is SOV and consistent with the language being verb final, the head word usually follows its dependent. Note that, the language direction indicates a tendency between left-branching (head-final) and right-branching (head-initial) structure (Stilo, 2005). Therefore the syntactic pattern has a mixed typology, and according to Stilo (Stilo, 2005), "this syntactic pattern often serves as a transition or buffer zone that represents a hybridization of two opposite patterns between a group of typically VO languages (as in Arabic) and a group of typically OV languages (as in Turkish)". Table 4 shows the set of interrelated syntactic features in Persian. However, prepositions and superlative adjectives always precede their dependent (nouns).

Sentences consist of an optional subject, and object followed by a compulsory verb, i.e., (S) (O) V. Subjects, however, can be placed anywhere in a sentence or they may completely be omitted as Persian is a pro-drop language with an inflectional verb system (where person and number are inflected on verb). The use and the order of the optional constituents are relatively arbitrary and this scrambling characteristic makes Persian word order highly flexible.

Left Branching	**Right Branching**
Object–Verb	Head–Modifier
Demonstrative Adjective–Noun	Preposition–Object
Numeral–Noun	Noun–Genitive
Adverb–Adjective	Noun–Adjective
	Noun–Relative Clause

Table 4: The syntactic patterns of Persian.

3. Existing Resources and Tools for Persian

The first linguistically annotated Persian corpus was the Bijankhan corpus (Bijankhan, 2004) released in 2004. The corpus consists of newspaper articles and common texts consisting of in total 4300 different topics such as cultural, technical, fiction, and art. The texts of total 2,597,939 words are annotated with morpho-syntactic and partly semantic features. The original tagset contains 550 tags organized in a tree structure. The tags follow a hierarchical annotation scheme starting with the most general tag and continues with the names of the subcategories. There is an updated version of the corpus with a reduced tagset of 40 tags containing only main part-of-speech categories with basic morphological features. The corpus comes with software for the calculation and extraction of language features such as: conditional distribution probability, word frequency, and recognition of homonyms, synonyms, concordances and lexical order.

Another linguistically annotated, but not freely available, corpus is the Persian Linguistic Data Base (PLDB) (Assi, 2005) containing information about pronunciation and grammatical annotation with a morpho-syntactic tagset of 44 tags. The database consists of more than 56 million words of contemporary texts.

The Tehran English-Persian subtitle corpus is another open source resource consisting of 554,621 aligned sentences. 57,465 word/phrase pairs extracted from a bilingual dictionary have been also added to this subtitle corpus (Pilevar et al., 2011).

Among other corpora for Persian, we can mention the Persian 1984 corpus, containing the translation of the novel *1984* by George Orwell annotated in the MULTEXT-East framework (QasemiZadeh and Rahimi, 2006). The corpus consist of 6,604 sentences, and about 100,000 words annotated with parts of speech. The corpus is part of the MULTEXT-East parallel corpus (Erjavec et al., 2003).

There are several speech databases such as, FARSDAT with the recordings of 405 sentences by 300 Persian speakers of different ages, sexes, educational levels, and 10 different regional dialects of Iran. The utterances have been manually segmented and labeled phonetically and phonemically using IPA characters. Among other speech corpora we can mention OGI (Oregon Graduate Institute of Science & Technology) Multilingual Corpus for speech recognition, CALLFRIEND FARSI for language identification, and TFARSDAT involving 7:56:7 hours of Persian monologue telephone speech of 64 Persian native speakers used for speech recognition and language identification. The

[1] An ezafe (-ez) is an unstressed enclitic particle that links the elements within a noun phrase, adjective phrase or prepositional phrase indicating the semantic relation between the joined elements and is represented by the short vowel /e/ after consonants or /ye/ after vowels.

Persian dialog telephone database comprises 100 hours of 200 Persian native speaker dialogs (Ghayoomi et al., 2010). In addition, there exist two open source lexicons for Persian, namely the Dehkhoda lexicon (Dehkhoda, 2006) and PerLex (Sagot et al., 2011). The Dehkhoda lexicon is an electronic monolingual Farsi lexicon containing 343,466 entries with information about the morphological structure of compound words (Dehkhoda, 2006). PerLex (Sagot et al., 2011) is another morphological lexicon consisting of about 36,000 lexical entries from the Bijankhan corpus (Bijankhan, 2004) and Wikipedia.

In parallel to our work, two different freely available treebanks were released with different annotation schemes. The Persian Dependency Treebank (Rasooli et al., 2012) consists of 10,000 manually annotated sentences representing dependency relations of 43 categories. The sentences are also annotated with morpho-syntactic features. The Persian Tree Bank (Ghayoomi, 2012) on the other hand, is based on HPSG grammar and has a rule-based approach by defining a set of rules in CLaRK. The treebank data set is taken from the Bijankhan corpus and contains 1000 trees.

As tools for Persian, a standard text preparation (STeP-1) (Shamsfard et al., 2009) has been designed to pre-process texts. The system employs a tokenizer, a morphological analyzer, and a spell checker to normalize texts into a standard one. Unfortunately the software is not open source.

To our knowledge there is only one freely available open source processing tool for Persian, namely the link grammar parser (Dehdari, 2006) based on the dependency-like link grammar (Sleator and Temperley, 1993). The link grammar parser produces an output where links are represented in such a way that every node involved in a link cannot be uniquely tied to a token position in the sentence. The parser provides no explicit way to extract the head of the sentence.

4. An Open Source BLARK for Persian

We describe the basic language resources and tools for processing Persian, which we develop and make freely available. Our BLARK components include a set of resources and tools for processing Persian texts such as a preprocessor **PrePer**, a sentence segmenter and tokenizer **SeTPer**, a part-of-speech tagger **TagPer**, and a parser **ParsPer**. As for resources, we present the Uppsala PErsian Corpus (UPEC), and the Uppsala PErsian Dependency Treebank (UPEDT) , which is still under development. An overview of the open source BLARK for Persian is given in Figure 4. Each component is described in detail in the following sections.

4.1. Uppsala PErsian Corpus: UPEC

The importance of having large scale annotated corpora for natural language processing is well-known. The Bijankhan corpus (described in section 3) is a good basic resource for Persian since the corpus is large, freely available and linguistically annotated. However, the corpus is created from on-line material and includes a wide range of typing styles with different character encodings which have an impact on the accuracy of natural language processing tools. In addition, the current release of the corpus lacks sentence segmentation. Therefore we decided to release a modified and normalized version of the corpus which is called Uppsala PErsian Corpus (UPEC). The normalization steps include the following:

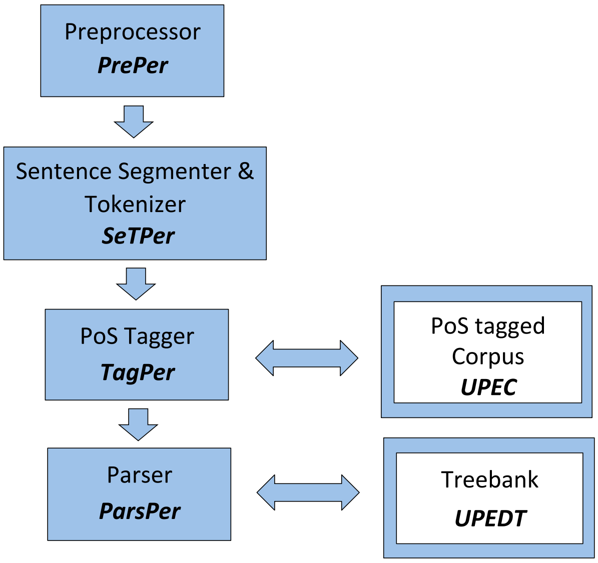

Figure 1: Open source resources and tools for Persian.

- In the Bijankhan corpus, the object marker را (*râ*) was annotated as a preposition P. In UPEC we modified the annotation to a CLITIC as the object marker *râ* always follows the object in Persian, i.e., it can be considered to be an enclitic case marker rather than a preposition.

- All past participle verbs with the tag V_PA (past tense verb) were modified to V_PP (past participle verb).

- Multi-word expressions, such as از آنجایی که (since, where), به دلیل این که (for the sake of, because), از این رو (hence), از آن جهت (thence), were treated inconsistently in the corpus; sometimes as one single token and sometimes as several. For a more consistent analysis we separated all multi-word expressions into their distinct tokens.

- Prepositional phrases made by prepositions and pronominal clitics such as ازت (of you), بهت (to you), or prepositions and demonstrative pronouns such as بدین (to this), بدان (to that) with the tag P were replaced by the tag P_PRO.

- Words with erroneous tags were also corrected and we are planning to correct the whole corpus as time allows.

- White spaces in compound words and inflectional affixes were changed to ZWNJ.

- All letters in Arabic style with Arabic Unicode characters were transformed to Persian style and Persian Unicode encoding.

- Arabic and Western digits were all converted to Persian digits.

4.2. A Pre-processor for Persian: Pre-Per

Persian texts, as mentioned earlier, can be written by different authors in various styles using different encodings, making Persian text processing issues extra challenging. Therefore, when processing Persian texts, the input texts need to undergo some preparations to be converted and cleaned up into pure standard texts. A standard text is a text written in standard style where the internal word boundaries are marked based on official orthography introduced by the Academy of Persian Language and Literature (APLL).

Our goal in creating a pre-processor for Persian was to normalize different writing styles with various encodings. Hence, in our BLARK pipeline we have inserted a pre-processor developed for Persian, named PrePer, as our first module to take care of various encodings with different Unicode characters, and various typing styles in different genres. Therefore, the input text first needs to pass through the pre-processor module.

PrePer is a software program developed in Ruby for the task of editing and cleaning up texts in Persian. The program uses the existing Virastar module for some formatting tasks (Barghi, 2011). The present PrePer handles miscellaneous cases and performs functions to normalize texts into computational standard script. PrePer via Virastar also takes care of the occurrences of mixed character encodings. By preprocessing texts all letters in Arabic style with Arabic Unicode characters are edited to Persian style with mapping to Persian Unicode encoding. In addition, Arabic and Western digits are all converted to Persian digits. PrePer also converts white space to ZWNJ between:

- nouns and plural suffixes ها- /-hâ/ , ان- /-ân/, ات- /-ât/, and ین- /-in/
- the suffixes /-i/ ی- or يی- (after long vowel /u:/) when denoting indefiniteness or abstractness, as well as the indefinite suffix ای- (after silent h) and any nouns when forming indefinite nouns or abstract nouns
- nouns and pronominal clitics
- past participle verbs and copula enclitics
- nouns and verbal stems in compound words
- verbal stems and the suffix اک- /-âk/
- verbal stems and the suffixes ار /âr/ or گار /gâr/ when forming nouns of action
- nouns and their adjacent suffixes when forming adjective-adverbs or adjective-nouns
- the negative prefixes نا /nâ/, and -بی (-im, -in, -un, -less) and its adjacent word
- the prefixes -سوء, -عدم, -فرا, and their adjacent words when forming determinative juxtaposed nouns and adjectives.

4.3. Sentence Segmentation and Tokenization: SeTPer

A sentence segmenter and tokenizer for Persian texts called SeTPer has been developed, aiming at segmenting texts based on Persian sentence boundaries, i.e., handling full stops, exclamation marks and question marks, and tokenizing a ZWNJ normalized text. SeTPer uses the modular software platform Uplug, a system designed for the integration of text processing tools (Tiedemann, 2003). The Uplug sentence segmenter and tokenizer is a rule-based program, that can be adapted to various languages by using regular expressions for matching common word and sentence boundaries. SeTPer treats the full stop, the question mark, and the exclamation mark as sentence boundaries.

Token separators are: apostrophe, brackets, colon, semicolon, dash, exclamation mark, question mark, at sign, slash, backslash, percent, asterisk, and tilde. The tokenizer also handles numerical expressions, web URLs, abbreviations, and titles. Acronyms are seldom used in Persian (Ghayoomi et al., 2010) but might exist in SMS, and in social media platforms, and are therefore also taken care of.

To evaluate SeTPer, we applied the software to two different texts; one in standard writing style and another in non-standard writing style. In standard writing style the internal word boundaries are marked on the basis of the official orthography of the Farsi language (Academy, 2005). The results showed that standard texts were successfully tokenized with an accuracy of 100% when evaluated on 100 randomly chosen sentences. On a random sample from non-standard texts, the tokenizer resulted in 98% recall and 95% precision on the character level, due to internal word boundaries not being marked in a consistent way.

4.4. A Statistical Part-of-Speech Tagger for Persian: TagPer

Our goal in creating a tagger for Persian was to develop a robust data-driven part-of-speech tagger to disambiguate ambiguous words (words with more than one tag), and annotate unknown words (not being in the training data). The part-of-speech tagger TagPer has been developed for Persian using the statistical part-of-speech tagger HunPoS (Halácsy et al., 2007), an open source reimplementation of TnT (Brants, 2000) that is based on Hidden Markov Models with trigram language models. The great advantage of HunPoS is that it is open source and freely available compared to other taggers.

To optimize HunPoS for Persian, we ran several experiments to train the tagger with different feature settings and feature combinations. Our experiments resulted in an overall accuracy of 96.9% (Seraji, 2011) for Persian when applied on the Bijankhan corpus which can be treated as a state-of-the-art result, and comparable to the performance of other data-driven part-of-speech taggers developed for Persian. The experimental results reported in (Seraji, 2011) are the best published results for an open source tagger for Persian so far, which means that HunPoS is a good alternative for the annotation of parts of speech in Persian. On the other hand, since we do not exactly know whether our study used the same training-test split as those used for other data-driven taggers explained in Raja et al. (Raja et al., 2007), the scores may not be directly comparable.

Later we performed the same experiment on UPEC, using the same training and test set data splits as on the Bijankhan corpus. The results revealed an overall accuracy of 97.8%. The result shows an improved performance of the tagger which is probably due to more consistent normalization and morpho-syntactic annotation of the corpus. The TagPer, part-of-speech tagger for Persian developed using UPEC is also freely available.

4.5. A Dependency Parser and a Treebank for Persian: ParsPer and UPEDT

The goal in creating a parser for Persian was to develop a data-driven dependency parser. Using machine learning with supervised learning techniques has already proven to be a successful approach in developing large linguistically annotated corpora in a short period of time. Furthermore, data-driven parsers have been shown to be able to efficiently assist the process of developing a treebank, and parser performance can be improved as the size of the treebank grows. In our work, we use MaltParser (Nivre et al., 2006) which is an open source data-driven parser based on dependency structures to syntactically annotate texts in order to build a treebank.

As the basis for treebank data we decided to use UPEC, as it is large open source balanced corpus with validated, morphologically analyzed texts. In order to build the treebank, we extracted 10,000 sentences randomly from our corpus to serve as treebank data with an average sentence length of 19 words per sentence (Seraji et al., 2012).

The annotation scheme is based on dependency structure, where each head and dependent relation is marked and annotated with functional categories, denoting the grammatical function of the dependent to the head. The syntactic annotation is based on the Stanford Typed Dependencies (Marneffe and Manning, 2008) which is a de facto standard for English. 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 adapted successfully to Finnish (Haverinen et al., 2010).

We have annotated 215 sentences from UPEC using the Stanford Typed Dependencies. The relations given by the Stanford scheme were directly applicable to Persian and most constructions could be analyzed. However, two constructions could not be covered, therefore we introduced the following two extensions:

- **Accusative marker:** The relation **acc** is used for the accusative marker of direct objects.
- **Light verb construction:** The relation **lvc** is used for the preverbal noun, adjective or adverbial elements in light verb constructions.

The syntactic relations with the extended Stanford scheme adapted to Persian are listed with explanations in Table 5.

It is worth noting that in our earlier published results (Seraji et al., 2012), the syntactic annotation scheme included two additional relations: **ezafe construction (ez)** and **interjection (int)**. The **ez** dependency label was used to indicate the semantic relation of the nominal elements in a sentence and has been replaced by the Stanford relations *poss* (possession modifier) and *amod* (adjective modifier) because of the similarity of the relational function. The **int** relation has been replaced by the Stanford relation *parataxis* based on the same motivation.

Category	**Description**
acc	*accusative marker*
advcl	adverbial clause modifier
advmod	adverbial modifier
amod	adjectival modifier
appos	appositional modifier
aux	auxiliary
auxpass	passive auxiliary
cc	coordination
ccomp	clausal complement
complm	complementizer
conj	conjunction
cop	copula
det	determiner
dobj	direct object
lvc	*light verb construction*
mark	marker
mwe	multi-word expression
nn	noun compound modifier
npadvmod	noun phrase as adverbial modifier
nsubj	nominal subject
nsubjpass	passive nominal subject
num	numerical structure
number	element of compound number
parataxis	parataxis
pobj	object of a preposition
poss	possession modifier
prep	prepositional modifier
punct	punctuation
quantmod	quantifier phrase modifier
rcmod	relative clause modifier
rel	relative
root	root
tmod	temporal modifier

Table 5: Syntactic relations in UPEDT based on Stanford Typed Dependencies including extensions for Persian.

215 sentences have already been manually corrected and validated to be used as seed training data for the data-driven dependency parser, MaltParser. To annotate and correct our syntactic annotation we used the free TrEd tree editor (Hajic et al., 2001). Figure 2 shows the dependency annotation for a sentence from the seed data set.

To evaluate the performance of MaltParser when trained on the seed data, we carried out an empirical study with 10-fold cross validation using the 215 manually validated sentences. Results from the current stage of the parser revealed a mean labeled attachment score of 62%. The developed parser ParsPer is intended to be optimized in later stages when the size of the treebank has grown.

To increase the size of the treebank, more sentences will

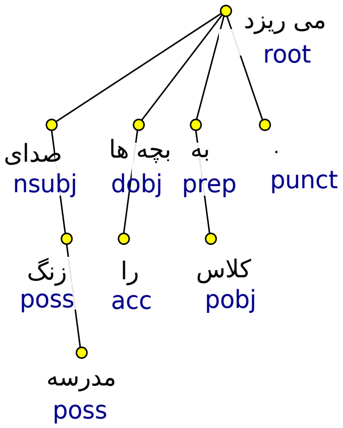

Figure 2: syntactic annotation for the Persian sentence صدای زنگ مدرسه بچه‌ها را به کلاس می‌ریزد. (The school bell brings the kids to the class.)

be parsed using MaltParser, trained on the seed data. The parsed sentences will be corrected and added to the seed data set. The process will continue as the size of the treebank grows.

5. Conclusion and Future Directions

In this paper we presented freely available new tools and resources for Persian. We described fully functional tools for preprocessing, sentence segmentation, tokenization, part-of-speech tagging, and a parser. As for resources, we present a normalized version of Bijankhan corpus with additional sentence segmentation and consistent tokenization modified for more appropriate syntactic annotation and a dependency treebank under development, as our contribution to a more complete BLARK for Persian. As the next step we will improve parsing accuracy and continue with further development of our treebank.

6. Acknowledgements

We are grateful to Jon Dehdari for generously sharing his annotation scheme and a set of annotated sentences as a starting point for our treebank annotation efforts. We would also like to thank Carina Jahani for her valuable comments and discussions concerning Persian grammar. The first author has been partly supported by the Swedish Graduate School in Language Technology (GSLT).

7. References

Iran's Academy. 2005. *Iran's Academy Of Farsi Language and Literature: Official Farsi Orthography*. ISBN: 964-7531-13-3, 3rd edition.

M. S. Assi. 2005. PLDB Persian Linguistics Database Pažuhešgarn (researchers). Technical report, Institute for Humanities and Cultural Studies, Iran.

A. A. Barghi, 2011. *Virastar*. https://github.com/aziz/virastar.

M. Bijankhan. 2004. The Role of the Corpus in Writing a Grammar: An Introduction to a Software. *Iranian Journal of Linguistics*, 19.

T. Brants. 2000. TnT a Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, Washington, USA.

J. Dehdari. 2006. *Crossing dependencies in Persian*. Master thesis, Brigham Young University.

A. A. Dehkhoda. 2006. *Dehkhoda Dictionary, 1931, the digital version of the Dehkhoda Dictionary*. Tehran University Press.

T. Erjavec, C. Krstev, V. Petkevic, K. Simov, M. Tadic, and D. Vitas. 2003. The MULTEXT-East Morphosyntactic Specifications For Slavic Languages. In *Proceedings of The EACL 2003 Workshop on the Morphological Processing of Slavic Languages*.

M. Ghayoomi, S. Momtazi, and M. Bijankhan. 2010. A Study of Corpus Development for Persian. *International Journal on Asian Language Processing*, 20(1):17–33.

M. Ghayoomi. 2012. Bootstrapping the Development of an HPSG-based Treebank for Persian. *Linguistic Issues in Language Technology*, 7:105–114.

J. Hajic, B. Hladk, and P. Pajas. 2001. Prague Dependency Treebank: Annotation Structure and Support. In *Proceeding of the IRCS Workshop on Linguistic Databases, Philadelphia*, pages 105–114.

P. Halácsy, A. Kornai, and C. Oravecz. 2007. HunPoS - an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Companion Volume Proceedings of the Demo and Poster Sessions. Association for Computational Linguistics, Prague, Czech Republic*, pages 209–212.

K. Haverinen, T. Viljanen, V. Laippala, S. Kohonen, F. Ginter, and T. Salakoski. 2010. Treebanking Finnish. In *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT)*, pages 79–90.

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

J. Nivre, J. Hall, and J. Nilsson. 2006. MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 2216–2219.

M. T. Pilevar, H. Faili, and A. H. Pilevar. 2011. Tep: Tehran English-Persian Parallel Corpus. In *Proceedings of 12th International Conference on Intelligent Text Processing and Computational Linguistics*.

B. QasemiZadeh and S. Rahimi. 2006. Persian in MULTEXT-East Framework. In *Advances in natural language processing, 5th International Conference on NLP, FinTAL*, pages 541–551.

F. Raja, H. Amiri, S. Tasharofi, H. Hojjat, and F. Oroumchian. 2007. Evaluation of part-of-speech tagging on Persian text. *The Second Workshop on Computational approaches to Arabic Script-based Languages, Linguistic Institute, Stanford University*.

M. S. Rasooli, M. Kouhestani, A. Moloodi, F. Bakhtiary, P. Dadras, M. Faal-Hamedanchi, S. Ghadrdoost-Nakhchi, M. Mahdavi, A. Mirzaei, N. Poormorteza-Khameneh, M. Rezaei-Sharifabadi, A. Shafie, S. Zamani, S. M. Hoseini, A. Noorian, Y. Souri, M. Hossein-Alizadeh, and M. Behniafar. 2012. Persian dependency treebank, annotation manual and user guide. Technical report, Dadegan.

B. Sagot, G. Walther, P. Faghiri, and P. Samvelian. 2011. A new morphological lexicon and a PoS tagger for the Persian language. In *Proceedings of Fourth International Conference on Iranian Linguistics*.

M. Seraji, B. Megyesi, and J. Nivre. 2012. Bootstrapping a Persian Dependency Treebank. *Linguistic Issues in Language Technology*, 7.

M. Seraji. 2011. A Statistical Part-of-Speech Tagger for Persian. In *Proceedings of the 18th Nordic Conference of Computational Linguistics NODALIDA*, pages 340–343.

M. Shamsfard, S. Kiani, and Y. Shahedi. 2009. STeP-1: Standard Text Preparation for Persian Language. In *Proceedings of the Third Workshop on Computational Approaches to Arabic Script-based Languages (CAASL3)*.

D. Sleator and D. Temperley. 1993. Parsing English with a Link Grammar. In *Third International Workshop on Parsing Technologies*.

D. L. Stilo. 2005. Iranian as buffer zone between the universal typologies of Turkic and Semitic. In Éva Ágnes Csató, Bo Isaksson, and Carina Jahani (eds.), editors, *Linguistic Convergence and Areal Diffusion: Case Studies From Iranian, Semitic, and Turkic*, pages 35–64, London: Routledge Curzon.

J. Tiedemann. 2003. *Recycling Translation - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. ISSN 1652-1366, ISBN 91-554-5815-7, Studia Linguistica Upsaliensia 1.