RePersian
A Fast Relation Extraction Tool in Persian

Raana Saheb-Nassagh, Majid Asgari, Behrouz Minaei-Bidgoli
Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
r_sahebnassagh@comp.iust.ac.ir, majid.asgari@live.com, b_minaei@iust.ac.ir

Received: 2020/04/08 Revised: 2020/05/29 Accepted: 2020/07/04

***Abstract*— The task of extracting semantic relations from raw data is called relation extraction. One of the most important fields in open information extraction is the automatically extraction of relations in any domain, especially in web mining. There are many works and approaches for relation extraction in English and other languages. Some of these approaches are based on parsing trees. Dependency parsing in the Persian language is difficult and time-consuming, since Persian is a low resource language and has also a dependency grammar and lexical structure, which affects also the speed of relations extraction in Persian. In this paper we will introduce a fast relation extraction method in Persian called RePersian. RePersian is dependent on part-of-speech (POS) tags of a sentence and special relation patterns, which are extracted by analyzing sentence structures in Persian. For finding relation patterns, RePersian searches through POS-tags that are given in regular expression forms. By matching the correct POS pattern to a relation pattern, RePersian extracts the semantic relations in a sentence. We appraise RePersian in two different scenarios on the Dadegan Persian dependency tree dataset. RePersian had on average the precisions 78.05%, 80.4% and 54.85% in finding the first argument on a relation, the second argument and the right relation between them.**

Keywords—Relation Extraction; Persian Language; Regex; POS Tag;

1. INTRODUCTION

The task of extracting semantic information from raw data is called relation extraction. One of the most important points in open information extraction (OIE) systems is to automate the relation extraction process. There are many analysis and researches which have tried to automate the relation extraction task. Some of them like TextRunner depend on shallow parsing and normalized noun phrases [1]. TextRunner has to main advantages: They are indexed and give fast accessibility. Others like WOE [2] utilize enormous sets of dependency-based extraction patterns. This gives a more accurate result but is slower than other OIE systems. Simple syntactic and lexical constraints can make the relation extracting task more profitable. ReVerb [3, 4] is one of the researches done in English that has considered syntactic constraints on relations expressed by verbs. Thusly ReVerb has proposed a faster and more capable method for finding semantic relations.

The above works, above all ReVerb, motivated us to look for a faster and greedier system for extracting relation in the Persian language. So we propose RePersian (Relation of Persian sentences) which has the same objective as ReVerb but uses other syntactic and constraints for relation extracting in Persian: POS-tags are the main structure of RePersian. By inspecting different writings in Persian, we conclude that each relation can be written as a special POS-pattern. So the relation extraction task can be achieved by converting the POS-Patterns regular expression (regex) syntax. Along these lines RePersian scans through the POS-tags for finding the relation patterns, which are given in regex-form and find semantic relations by matching the correct POS-patterns. For the evaluation part of this paper we examined our method with the Persian dependency tree dataset [5] in different cases.

This paper is organized as follows. In section 2 we present the basic knowledge of relation extraction and regex syntax. In section 3, we will discuss the focal points and drawbacks of the related works. Section 4 describes the model itself. Finally we test RePersian in section 5 on the Persian dependency tree dataset. Topics for future work are described in section 6.

2. BACKGROUND

In this section, we will present the basic knowledge that is related to the proposed method.

2-1. Open Information Extraction

Finding the semantic relations between entities is a major key in many NLP-applications [6]. Open Information Extraction opens a new way for extracting sets of relational tuples automatically without requiring hand-tags or any human input which were utilized in the conventional information extraction (IE) for each new domain [1].

Relation extraction, which is a subtopic of IE is used for extracting a set of rational tuples and also gives structured information of open information systems [7]. The task of extracting relations between entities in a text is called relation extraction. These relations can happen between at least two entities (for example Person, Organization, and Location). With each relation a semantic relation between the entities (e.g. come from, go to, interested in) is showed. Finally relations between entities in a text can help to have a better understanding of the text.

Table 1 mentions some examples of relation extraction in Persian with the English translation.

In Section 3, we will discuss different relation extraction methods for the Persian language and also other new techniques in other languages.

2-2. Persian Part of Speech (POS) tags

Part-of-speech tagging (POS tagging) is the task of tagging a word in a text. In other words a collection of words with the same grammatical properties is called the POS tagging. Noun, verb, adjective, adverb, pronoun, preposition and conjunction are the most routine English POS tagging. RePersian is based on part-of-speech (POS) tags, because of that we present here some of the POS-tags which are utilized in the Persian language.

In Table 2 we introduce POS tagging which are used in this work [5]. Table 2 shows also some of the POS-tags like ADJe and Ne that are explicit for the Persian language, since the 'e' represents the Persian diacritical mark "Kasre". Kasre is a short –e sound which is utilized for possession of nouns, adjectival phrases and naming. Like other Persian vowels is Kasre most of the time not written but read. In Table 3 we have shown some of the usages of Kasre with examples in English and Persian.

2-3. Dependency tree relation

The grammatical relations and structure of a sentence can be shown with the help of a dependency tree [8]. One of the advantages of this representation is its simple structure. In this work we assess our outcomes with the Persian dependency tree dataset [5] in separate scenarios. In the evaluation part we check if the founded entities in the extracted relations are one of the main grammatical Persian phrases, in other words SBJ-OBJ-PP-MOS because entities in Persian have mostly one of these four syntactic structures in the sentence [9]. We also evaluate if the extracted relation for the entities speaks for a verb.

For the evaluation of the founded entities and verbs (relations) we use the main grammatical phrases (dependency relations) in the Persian language which are shown in Table 4.

It is notable to say that in the evaluation process, we only tested the central syntactic part of the phrases (the center of each phrase) and did not focus on other verb or noun-dependencies.

2-4. Regex

A regular expression, regex or regexp (also called a rational expression) is an array of characters that proclaims a search pattern [10,11,12]. Each character in a regex has either a special meaning or a literal meaning. For example, in the regex "a^{+}", "a" has a literal meaning which shows only the character 'a', while '$^{+}$' has a special meaning and signifies at least one occurrence of one character. In other words the regex "a^{+}" matches with "a", "aa", "aaa" and so on [13]. Regular expressions help us to search for a specific pattern in a context in an easier way.

In this work, for each grammatical phrase in the Persian language, SBJ-OBJ-PP-MOS-V, a specific POS pattern, is used. These POS pattern are written for the ease of use with the regex syntax. For implementing the regexes we used PyRegex (regex in python) [14]. Table 5 shows the regex syntax used in this work. In section 4 we introduce for each POS tag the regex expression.

In this section, we discuss about the most relevant researches to this work. To have a better understanding we separate this section into two subsections:

1. Relation extraction works in Persian
2. State-of-the-art relation extraction works in other languages.

3. Related work

3-1. Relation extraction in Persian

There are many different relation extraction methods in the Persian language. One of the mentionable works which is

TABLE 1. EXAMPLES OF RELATIONS IN PERSIAN

Persian sentence	Extracted relation	Translated sentence	Translated relation
من غذا را خوردم.	من- خوردم - غذا	I eat the food.	I – eat – the food
او به پدرش سلام کرد.	او - سلام کرد- به پدرش	He greeted his father.	He – greeted – his father
خیابان شلوغ است.	هوا - است - بسیار گرم	The street is crowded.	The street – is - crowded

TABLE 2. POS TAGGING USED IN THIS WORK

POS tag	Meaning
CONJ	Coordinating Conjunction
POSTP	Postposition
PR	Pronoun
ADJ	Adjective
ADJe	Adjective with Kasre
N	Noun
Ne	Noun with Kasre
PREP	Preposition
PRENUM	Numeral
PREM	Determiner

TABLE 3. USAGES OF KASRE

Situation	Explanation	Example in Persian	Translated example
Possession	Kasre is the connection between the proposed noun with its owner	- کیف من - در خانه	- My bag - The door of the house
Adjectival phrase	Kasre is the conncetion between the noun and its adjective	- هوای گرم - کتاب قشنگ	- A hot day - Nice book
Namings	Kasre is the connection between names and titles, places or seasons.	- خانم کیانی - خیابان تجریش - فصل بهار	- Miss Kiani -Tajrish Street - Spring season

TABLE 4. DEPENDENCY RELATIONS USED IN THE EVALUATION PART

Dependency Relation	Meaning
SBJ	Subject
OBJ	Object
PP	Preposition
MOS	Mosnad
V	Verb

TABLE 5. REGEX SYNTAX USED IN THIS WORK

Regex Syntax	Explanation
^ ?!	Not (at the beginning)
?!	Not
|	Or
?	Zero or one occurence
+	One or more occurence
*****	Zero or more occurence

based on verbal extraction, is the method presented by Bagherbeygi and Shamsfard. They suggested an automatically method for extracting verbal concepts. One of the notable features of this work is the combination of noun/adjective and Persian verbs as a compound verb concept [15].

Bagherbeygi and Shamsfard also introduced another work that uses POS tagging for the extraction of compound verbs and their relations. This approach achieved an efficient accuracy among the other works done in this field. It is notable to say that this work considered only verbs. Other phrases weren't considered [16].

Shamsfard also introduces another similar work that searches for relation patterns in the Persian language. In this work linguistic patterns for semantic categories and phrases have been extracted. One of the advantages of this work is its accurate result for finding phrases in text. On the other side it did not present a well-defined mechanism for relation extraction with the extracted patterns and also used too many features from the text like lexemes, POS tags, syntactic categories, semantic similarities and constraints [17].

There are also works that process relation learning based on a special structure like the Wikipedia structure. One of the first systems which used Wikipedia structure in Persian is the work of Fadaei and Shamsfard. They used in their approach a combination of statistics, structure-based and pattern-based methods for learning the relations in Persian [18].

Momtazi and Moradiannasab presented a statistical approach for relation extraction. Recording to a bootstrapping approach based on the n-gram* model, they searched for relations between named entities in Persian. One of the advantages of this work is that it doesn't need any background knowledge of the target language. On the other side it only covers relations in subject-object-verb formats, which can be a limitation of this work. [19].

There have been also other ideas for finding triples in Persian: Sajadi and Minaei presented a Persian knowledge graph with more than 500K of entities and 7 million relations. This work showed in experiments that more than 94% of the triples were labeled correctly. But it is notable to mention that knowledge graphs need a high process of extraction and a widely storage capacity [20].

By exploring the mentioned works for relation extraction in Persian, we were inspired to search for a more rule-based way with simple patterns for relation extraction that do not need much data about the content and furthermore can work with just having the POS tags of the words.

3-2. Relation extraction in other languages

Naderi Golshan et al. studied the recent works in information extraction and relation extraction [21]. Golshan et al. introduced the work of Socher et al. as one of the earliest works that use the RNN method for relation extraction. Socher et al. built a system called "MV-RNN" based on the IMDB dataset and got accuracy about 80% on the test set [22]. Besides RNN, CNN was another method used for relation extraction [23, 24] and also for named entity recognition [24, 25]. In spite of the fact that these works had precision over 70% in relation extraction, they did not have such a high recall in this area.

Wang et al. also used CNN, but differently. They employ a CNN with entity-aware attention to extract high level features from tokens that were extracted with the BERT architecture. In this work the model achieves about 90% F1-score [26]. Other studies like the work of Meishan et al. discussed the performance of CNN-models with different pooling operations and regularization parameters for relation extraction [27].

In another system called "IExM" Distant-Supervised algorithm was used for relation extraction. The data for this study was accumulated from film articles and could so get a momentous precision but again not so convincing recall [28].

As indicated by Naderi Golshan et al. a Rule-Based framework called Card Pyramid got the highest precision, recall and F1-score in relation extraction among all referenced works that are RNN, CNN and IExM [29].

By studying the related researches in different languages, we got motivated to look for a more persuading rule-based method indicated for the Persian language which is quicker than parsing-based techniques, CNN and RNN, which need a tedious period for learning an immense amount of information. We built a method for relation extraction in Persian dependent on specific patterns in the POS tagging, which is described in the following section.

4. THE PROPOSED METHOD

This section presents the overall mechanism of RePersian. Our main idea is to define a fast relation extraction method based on the POS tags of a sentence in Persian. To do so, we first have to find the dependency relation of the sentences, because the finally extracted relations, in other words, the triples of argument-relation-argument, will represent a relation between two entities or nouns.

Entities can be found by finding the four grammatical structures: SBJ-OBJ-PP-MOS, because entities in Persian can have only one of these four types [9].

Since the extracted relations represent a relationship between the subjects of the sentences with other entities, the relation of the triples can be found by discovering the verb of the sentence.

In other words the first argument of a relation and the relation itself can be extracted by finding the subject of a sentence and the verb dependent on it. The second argument mostly represents one of the three other noun clauses: OBJ, PP or MOS that is also dependent on the same verb of the sentence [9].

Thus, relation extraction changes into identifying the main phrases, SBJ-OBJ-PP-MOS-V, of the sentence. Rather than using a time-consuming parse tree, a faster and simpler method is employed. The unavailability of the parsed dependency tree of a sentence necessitates finding another method. A simple available feature of words in a sentence is the POS tags. By searching in the POS tags of the four main phrases in sentences, a repeated POS structure is found for each phrase. For example, the POS tag of the SBJ phrase follows a particular pattern. When the pattern is available for the SBJ,

this pattern can be searched through the POS tags of the words in a sentence, finding the subject of the sentence.

This study examines the structure of each phrase separately in Persian and finds a particular pattern of POS tags for each phrase. Such patterns have been written in a regex form to make the searching part easier. Although it may seem to be impossible to find a pattern for each sentence phrase, an efficient system was constructed by concatenating several possible regexes for each sentence phrase in one regex. The system searches all possible regex forms for a particular phrase, such as SBJ, and matches, at last, the longest possible match.

For a better overview of the system flow, the main phrases of the RePersian are presented below:

1) For each sentence phrase, SBJ-OBJ-PP-MOS-V, a particular regex pattern was found, which will be described later.

2) For relation extraction from a sentence, the POS tag of each word in the sentence is first found.

3) Then, the SBJ, OBJ, PP, MOS, and V of the sentence are searched for with the help of their regex pattern. For example, to find the subject of the sentence, the POS tags of the sentence are searched, trying to match the longest possible pattern in the POS tags with the SBJ-regex. In case there are more than one answer, the longest possible match is chosen

4) By finding the main phrases of a sentence, the main entities and relation of the sentence are found. In the last step, it is seen which relations can be extracted. Since Persian sentences have a limited structure, the found relation and entities concerning Table 6 are presented [9]. For example, if only SBJ, OBJ, and V are found, there is a relation between the SBJ and OBJ with the related V according to Table 6. In other words, the first argument of this relation is the subject of the sentence. The second argument is the object of the sentence, and the relation is the verb of the sentence.

To better understand this study, a more accurate explanation of each main sentence phrases and its regex is provided in the next subsections.

4-2. Subject

The subject of a sentence in Persian is often a group of nouns coming at the beginning of sentences.

To find the SBJ-pattern, it is required to find the POS pattern of a noun clause. When the noun pattern is available, the first noun clause can be chosen heuristically as the subject.

Table 7 shows different existing noun clauses in Persian and their POS tags and possible regexes. The final regex for the SBJ phrase is a concatenation of the entire possible regexes. It is worth mentioning that the SBJ-regex also has constraints: The SBJ-regex cannot begin with a preposition, considering that prepositions are used in PP phrases, and cannot end with a postposition since postpositions are used in the OBJ phrase of a sentence. Thus, the final PyRegex does not include PREP and POSTP, which is shown with the PyRegex-syntax.

The final SBJ-regex which is mixture of all regexes and other constraints is showed in (1):

r“ ^(?! **PREP**) (**PREM** | **PRENUM** (**PREP** (**ADJ**)?)?)? ((**N** (**ADJ** | **N**$^{+}$)* (**ADJ** | **N** | **PR**)) | (**N** (**ADJ**)?) | **PR**) (?! **POSTP**) (**CONJ**)? " (1)

and with Kasre in (2):

r" ^(?! **PREP**) (**PREM** | **PRENUM** (**PREP** (**ADJ**)?)?)? ((**Ne** (**ADJe** | **Ne**$^{+}$)* (**ADJ** | **N** | **PR**)) | (**N** (**ADJ**)?) | **PR**) (?! **POSTP**) (**CONJ**)? " (2)

4-3. Object

The object of a sentence in Persian is often a set of nouns followed by the postposition “RA”. Knowing this fact the object of a sentence has the same pattern as the SBJ pattern with the difference that the noun clause must follow a postposition and cannot begin with a preposition.

So the PyRegex of the object is similar to that of the subject but needs the POSTP at the end. This method does not work when the object of a sentence is not followed by the postposition sign.

The final OBJ-regex has this syntax shown in (3):

r" ^(?! **PREP**) (**PREM** | **PRENUM** (**PREP** (**ADJ**)?)?)? ((**N** (**ADJ** | **N**$^{+}$)* (**ADJ** | **N** | **PR**)) | (**N** (**ADJ**)?) | **PR**) (**POSTP**) " (3)

and with Kasre in (4):

r" ^(?! **PREP**) (**PREM** | **PRENUM** (**PREP** (**ADJ**)?)?)? ((**Ne** (**ADJe** | **Ne**$^{+}$)* (**ADJ** | **N** | **PR**)) | (**N** (**ADJ**)?) | **PR**) (**POSTP**) " (4)

TABLE 6. THE RELATION DEPENDENCY IN PERSIAN

Persian sentence	relation	Translated sentence	Translated relation	Relation dependency of the Persian rel.
من کتاب را خریدم.	من – خریدم – کتاب را	I bought the book.	I – bought – the book.	SBJ – V - OBJ
من به مدرسه رفتم.	من – رفتم– به مدرسه	I went to school.	I – go – to school	SBJ – V - PP
من خوشحال هستم.	من – هستم - خوشحال	I am happy.	I – am – happy.	SBJ - V - MOS
من داستان را به او تعریف کردم.	من - تعریف کردم – داستان را– به او -	I told him the story.	I – told – him –the story	SBJ – V – OBJ - PP
این را خوردم.	خوردم – این را	(I) eat it	eat - it	V – OBJ
به مدرسه رفتم.	رفتم – به مدرسه	(I) went to school	went – to school	V - PP

TABLE 7. REGEX OF NOUN CLAUSES

Persian sentence	Translated sentence	Explanation	POS tag of the noun clause	PyRegex form of the noun clause	PyRegex form of the noun clause (with Kasre)
آشپز آمد.	The cook comes.	The noun clause can be just a noun.	N	N	N
یک آشپز آمد.	One cook comes.	The noun clause can have a numeral.	PRENUM-N	PRENUM?N	PRENUM?N
او آمد.	He comes.	The noun clause can be a pronoun.	PR	PR	PR
آشپز و گارسون آمدند.	The cook and the waiter come.	The noun clause can be connected to another noun or pronoun with the connector "and"	N-CONJ-N	N(CONJ)?	N (CONJ) ?
آن آشپز آمد.	That cook comes.	The noun clause can have a determiner.	PREM-N	PREM?N	PREM?N
آشپز زیبا آمد.	The beautiful cook comes.	The noun clause can be followed by one or more adjectives.	N-ADJ	N(ADJ)*	N (ADJe)* ADJ
بهترین آشپز آمد.	The best cook comes.	The adjective of the noun clause can come before the noun.	ADJ-N	(ADJ)*N	(ADJ)* N
آشپز رستوران آمد.	The cook of the restaurant comes.	Noun clauses can also have possessive nouns.	N-N	N(N)*	N(Ne)* N
آشپز من آمد.	My cook comes.	Noun clauses can have come with a possessive pronoun.	N-PR	N(PR)?	Ne(PR)?
آشپز زیبای رستوران معروف آمد.	The beautiful cook of the famous restaurant comes.	Noun clauses can be a mix of adjectives and possessive nouns.	N-ADJ-N-ADJ	(ADJ)?N((ADJ)|(N)+)*	(ADJ)? N((ADJe)|(Ne)+)*(ADJ|N)?
یکی از بهترین آشپزها آمد.	One of the best cooks comes.	Noun clauses can also come with the form "one of the + superlative."	PRENUM-PREP-ADJ-N	(PRENUM)(PREP)(ADJ)N	(PRENUM)(PREP)(ADJ)N

4-4. Prepositional phrase (PP)

The main preposition clause of a sentence in Persian is often a noun clause that begins with a preposition. Thus, the PyRegex of the PP is the same as the SBJ-regex but also has the PREP sign at the beginning. Since adverbs with preposition can be detected as the primary preposition of the verb in this method, some false positives will probably exist in this part, but we prefer for now the fastness of this work than a more complicated but accurate one.

The final syntax of the PP-regex is shown in (5):

r" (**PREP**) (**PREM** | **PRENUM** (**PREP** (**ADJ**)?)?)? ((**N** (**ADJ** | **N**$^{+}$)* (**ADJ** | **N** | **PR**)) | (**N** (**ADJ**)?) | **PR**) (?! **POSTP**) (**CONJ**)? " (5)

and with Kasre in 6:

r" (**PREP**) (**PREM** | **PRENUM** (**PREP** (**ADJ**)?)?)? ((**Ne** (**ADJe** | **Ne**$^{+}$)* (**ADJ** | **N** | **PR**)) | (**N** (**ADJ**)?) | **PR**) (?! **POSTP**) (**CONJ**)? " (6)

4-5. Mosnad (MOS)

The Mosnad of a sentence in Persian appears merely in sentences with special Mosnad-Verbs (to be, to become, etc.). Thus, Mosnad is looked for in sentences with the Mosnad-Verbs. The Mosnad is also a noun clause. It is searched for with the SBJ-regex in sentences with Mosnad-Verbs. Since the first noun clause of the sentence is heuristically chosen as the subject, the second noun clause is also chosen heuristically as the Mosnad of the sentence.

Thus, the MOS-regex has the same regex as the SBJ which has been shown in (1) and (2).

4-6. Verb (V)

The verb of a sentence in Persian can be found easily by the POS tag "V". The only problem can occur with compound verbs. In Persian, some verbs are constructed of a verb and a non-verbal element, such as a noun, adjective or preposition. Such Verbs are known as compound verbs since they are composed of several components. Thus, a compound Verb cannot be found easily by the POS tag "V". This study solved the problem by merely focusing on simple verbs. Compound verbs can be addressed in future work.

The V-regex in show in (7):

r" **V*** **V** " (7)

Finally, the PyRegex of a phrase can be used to find the main phrases of the sentence and the relations between phrases.

5. EVALUATIONS AND COMPARISONS

This section evaluates the method on the Persian dependency tree dataset. This dataset was evaluated in two different tests:

Test 1. The raw Persian dependency tree was used without changing any parts of it.

Test 2. The Persian dependency tree was used along with the addition of Kasre to the POS-tags.

In this section, RePersian was employed to find a relationship between two or more entities in a sentence. The entities in Persian relations can include these phrases in a sentence: SBJ, OBJ, PP and MOS. The relation must be a verb (V). Table 6 shows the dependency relation of a few simple Persian sentences.

To evaluate the model, Persian relation were extracted from the Persian dependency tree dataset with RePersian according Persian phrases. In other words, the phrases described in Table 6 were found in sentences and compared to the real dependency relation of the Persian dependency tree dataset. By identifying the correct relation dependency of each word in the sentence, the correct relations between the entities were also found. Thus, for each case study in this section, the accuracy of each of the founded phrases was evaluated (e.g. SBJ, OBJ, PP, MOS and V) in a relation.

5-1. Test 1: The Persian dependency tree dataset

The evaluation part of the relations is dependent on the correct detection of the phrase in a sentence. Figure 1 illustrates precision, recall and f1-score of each of the five phrases, i.e., SBJ, OBJ, PP, MOS and V, in the sentences.

According to Figure 1:

- The SBJ relation has a convincing precision. An explanation can be the fact that most subjects in Persian sentences come at the beginning of sentences. The regex for subjects also searches for the first match in a sentence.
- The OBJ relation is highly precise. An explanation can be the fact that objects in Persian come with the postposition "RA" (Persian: را), leading to a smaller searching space for the objects. The low recall arises from objects that are not succeeded by "RA."
- The PP relation is highly precise. An explanation can be the fact that PPs in Persian come with a preposition (such as to, at, by), leading to a smaller searching space for the prepositions. The low recall arises from sentences with adverbs and more than one preposition. RePersian finds the first match. The main preposition clause of a Persian sentence is often a noun clause that begins with a proposition related to the verb. Since adverbs with preposition can also be detected as the primary preposition of the verb in this method, some false positives will probably exist in this part.
- The MOS relation has low precision and recall. An explanation can be the fact that MOS in Persian has no signs or fix places in the sentence. In this method, a heuristic way was chosen for finding MOS, which causes such results in this part and can be approved by identifying a better solution.
- The V relation has precision and recall of approximately 50%. An explanation can be the fact that many verbs in Persian have more than just one part. So RePersian does not always find all the parts of a verb.

5-2. Test 2: The Persian dependency tree dataset with Kasre

Figure 2 shows the precision, recall and f1-score of each of the five phrases, SBJ, OBJ, PP, MOS and V, in the sentences.

According to Figure 2:

- The SBJ, OBJ , PP and V relation has a lower precision than the first test. Maybe the added Kasre has brought some complexities to the regexes.
- The MOS relation has been again not been extracted right and needs a more specific way to be found.

6. Conclusions and Future Work

This paper presents a rapid relation extraction method known as RePersian based on POS tags for Persian. RePersian searches the POS-tags of the sentences for particular relation patterns that are written for each grammatical phrase in the Regex-form.

As a result, RePersian finds semantic relations by matching the correct POS Pattern to a relation pattern.

To evaluate the approach, the Persian dependency tree dataset with two different POS-tags was employed.

The approach had a mean precision of 78.05% for finding the first argument of a relation, a precision of 80.4% for finding the second argument and a precision of 54.85% for finding the correct relations between the entities.

As a future work, we intend to identify better heuristic ways to match a pattern to its right phrase. We also look for other regex formats. Although PyRegex is a popular library for regex implementations, still other ways that may have a better approach then PyRegex such as the Standford Token Regex exist.

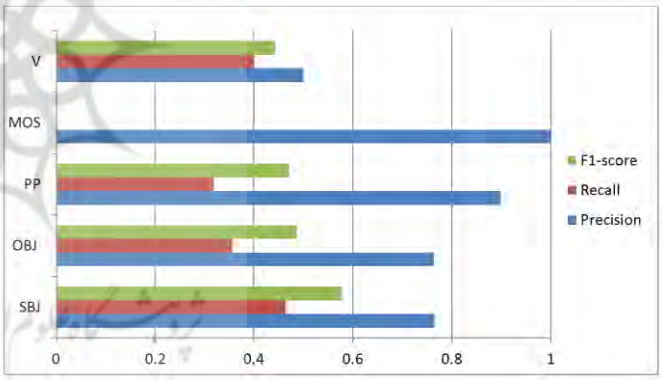

Fig. 1. Summary of test-results 1

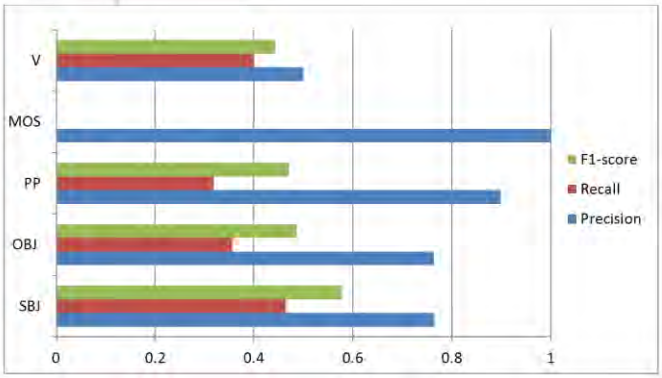

Fig. 2. Summary of test- results 2

Another open problem is compound verbs, which were not considered in this work and can be an important research topic for improving the results of RePersian.

Also, the MOS phrase could not be found by RePersian. Since MOS has no signs in Persian, it can be easily mistaken with other noun phrases. By finding MOS in future researches, arguments of relations can be found more precisely.

The proposed approach can also be easily extended by adding more regexes or even changing some regexes, which can lead to a higher precision and recall.

ACKNOWLEDMENTS

The project is contrived by the IUST-NLP lab. We thank all the participants of our lab for moving the idea of this research forward.

REFERENCES

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web. *InIjcai*, 2007 Jan 6,Vol. 7, pp. 2670-2676).

[2] F. Wu, D. S. Weld, "Open information extraction using Wikipedia". In *Proceedings of the 48th annual meeting of the association for computational linguistics*, Uppsala, Sweden. Association for Computational Linguistics. 2010 Jul 11, pp. 118-127.

[3] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction". In *Proceedings of the conference on empirical methods in natural language processing* ,Association for Computational Linguistics, 2011 Jul 27, pp. 1535-1545.

[4] O. Etzioni, A. Fader, J. Christensen, and S. Soderland, "Open information extraction: The second generation". In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011 Jun 28.

[5] M. S. Rasooli, M. Kouhestani, and A. Moloodi, "Development of a Persian syntactic dependency treebank", *In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013 Jun, pp. 306-314.

[6] N. Bach, and S. Badaskar, "A review of relation extraction". *Literature review for Language and Statistics II*. Language Technolgies Institute, Carnegie Mellon University, 2007.

[7] N. Konstantinova, "Review of relation extraction methods: What is new out there?". In *International Conference on Analysis of Images, Social Networks and Texts*, Springer, Cham, 2014 Apr 10, pp. 15-28.

[8] A. Culotta, and J. Sorensen, "Dependency tree kernels for relation extraction". In *Proceedings of the 42nd annual meeting on association for computational linguistics* Association for Computational Linguistics, 2004 Jul 21, p. 423.

[9] A. M. Haghshenas, A. Samie Gilani, T. Vahidian Kamyar, H. Davodi, H. Zolfaghari, M. R. Sangari, G. R. Omrani, H. Ghasempour Moghaddam, S. A. Mirjafari. *Persian Language 2* (written in Persian). Tehran: Iran Textbook Publishing Company, 2015

[10] Regular-Expressions.Info, "Regular Expression Tutorial - Learn How to Use Regular Expressions", URL: https://www.regular-expressions.info, Access Date: 17 July 2013.

[11] R. Mitkov, editor. *The Oxford handbook of computational linguistics*. Oxford University Press; 2004.

[12] M. V. Lawson, *Finite Automata*. New York: Chapman and Hall/CRC, 2003 Sep 17.

[13] A. Watt, *Beginning Regular Expressions*. Indiana: John Wiley & Sons, 2005 Feb 4.

[14] "PyRegex", URL: http://www.pyregex.com/, Access Date: 8 May 2012.

[15] M. Shamsfard, A. Hesabi, H. Fadaei, N. Mansoory, A. Famian, S. Bagherbeigi, E. Fekri, M. Monshizadeh, and S. M. Assi, "Semi automatic development of farsnet; the persian wordnet". In *Proceedings of 5th global WordNet conference*, Mumbai, India 2010 ,Vol. 29.

[16] S. Bagherbeygi, M. Shamsfard, Corpus based Semi-Automatic Extraction of Persian Compound Verbs and their Relations. In *LREC* 2012 May, pp. 2863-2867.

[17] M. Shamsfard, "Lexico-syntactic and Semantic Patterns for Extracting Knowledge from Persian Texts". *International journal on Computer Sciences and Engineering*, vol. 2, no. 06, pp. 2190-6, 2010.

[18] H. Fadaei, and M. Shamsfard, "Extracting conceptual relations from Persian resources". In *2010 Seventh International Conference on Information Technology: New Generations*, IEEE, 2010 Apr 12, pp. 244-248.

[19] S. Momtazi, and O. Moradiannasab, "A statistical approach to knowledge discovery: Bootstrap analysis of language models for knowledge base population from unstructured text". *Scientia Iranica*, 26-39, 2019.

[20] M. B. Sajadi, and B. Minaei Bidgoli, "The Architecture of Farsi Knowledge Graph System". *Iranian Journal of Information processing and Management*., p. 587, 2019.

[21] P. N. Golshan, H. R Dashti, S. Azizi, and L. Safari, "A Study of Recent Contributions on Information Extraction". arXiv preprint arXiv:1803.05667, 2018 Mar 15.

[22] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces". In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, Association for Computational Linguistics, 2012 Jul 12, pp. 1201-1211.

[23] T. H. Nguyen, and R. Grishman, "Relation extraction: Perspective from convolutional neural networks". In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015 Jun, pp. 39-48.

[24] S. K. Sahu, A. Anand, K. Oruganty, and M. Gattu, "Relation extraction from clinical texts using domain invariant convolutional neural network". arXiv preprint arXiv:1606.09370. 2016 Jun 30.

[25] P. Li, and H. Huang Clinical information extraction via convolutional neural network. arXiv preprint arXiv:1603.09381. 2016 Mar 30.

[26] Y. Wang, X. Xin, and P. Guo, "Relation Extraction via Attention-Based CNNs using Token-Level Representations". In *2019 15th International Conference on Computational Intelligence and Security (CIS)*, IEEE, 2019 Dec 13, pp. 113-117.

[27] F. Meishan, L. Zhi, W. Hao, Q. Wen, C. Fei, and L. MingYu, "Evaluation of Pooling Operations and Regularization Parameters in Neural Networks for Drug-drug Interaction Extraction". In *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence* ,2019 Dec 6, pp. 112-117.

[28] P. Y. Chen, Y. H. Lee, Y. H. Wu, and W. Y. Ma, "IExM: Information Extraction System for Movies". In *Proceedings of the 26th International Conference on World Wide Web Companion*, International World Wide Web Conferences Steering Committee, 2017 Apr 3, pp. 189-193..

[29] R. J. Kate, and R. J. Mooney, "Joint entity and relation extraction using card-pyramid parsing". In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2010 Jul 15, pp. 203-212.

Raana SahebNassagh- Master Student in artificial intelligence at the Iran University of Science and Technology. Experienced in Software Anti Patterns.

Majid Asgari- PhD Student in artificial intelligence at the Iran University of Science and Technology. Experienced in NLP and knowledge graphs.

Behrouz Minaei-Bidgoli- Associate Professor and director of research in the Computer Engineering Department at Iran University of Science and Technology. Leading the Data Mining Lab (DML) that does research on various areas in artificial intelligence and data mining, including text mining, web information extraction, and natural language processing.