International Journal of Information Science and Management
Vol. 18, No. 1, 2020, 93-108

Persian Semantic Role Labeling Based on Dependency Tree

Soghra Lazemi
Ph D. candidate in Artificial Intelligence, Department of Computer Engineering, Faculty of Electrical & Computer Engineering, University of Kashan, Kashan, Iran, Corresponding Author
Soghra.lazemi@gmail.com

Hossein Ebrahimpour-Komleh
Assistant Prof. Department of Computer Engineering, Faculty of Electrical & Computer Engineering, University of Kashan, Kashan, Iran, Ebrahimpour@kashanu.ac.ir

Nasser Noroozi
Assistant Prof. Department of Computer science
Faculty of Mathematics. University of Kashan, Kashan, Iran
Noroozi@kashanu.ac.ir

Abstract

Semantic role labeling is the task of attaching semantic tags to the words according to the event represented by the sentence. Persian semantic role labeling is a challenging task and most methods proposed so far depend on a huge number of manually extracted features and are applied on feature engineering to attain high performance. On the other hand, considering the Free-Word-Order and Subject-Object-Verb-Order characteristics of Persian, the arguments of the verbal predicate are often distant and create long-range dependencies. The long-range dependencies can hardly be modeled by these methods. Our goal is to achieve a better performance only with minimal feature engineering and also to capture long-range dependencies in a sentence. To these ends, in this paper a deep model for semantic role labeling is developed with the help of dependency tree for Persian. In our proposed method, for each verbal predicate, the potential arguments are identified by dependency relations, and then the dependency path for each pair of predicate and its candidate argument is embedded using the information in the dependency trees. In the next step, we employed a bi-directional recurrent neural network with long short-term memory units to transform word features into semantic role scores. Experiments have been done on the *First Semantic Role Corpus in Persian Language* and the corpus provided by the authors. The achieved Macro-average F_1-measure is 80.01 for the first corpus and 82.48 for the second one.

Keywords: Semantic Role Labeling, Full-Syntactic Parsing, Shallow Syntactic Parsing, Dependency Tree, Phrase-structure Tree, Persian language.

Introduction

The goal of natural language processing with machine, as an interdisciplinary science between the language science and computer science, is to understand human language and, in a higher level, to generate it in the world of intelligent computing. The first step for meaning-understanding is semantic role labeling (SRL) or shallow semantic parsing (SSP). SRL is done at the sentence level with the goal of grouping a sequence of words and classifying them with appropriate semantic labels according to the event represented by the sentence. Semantic roles are attached in response to questions such as *who, what, how, what time, where etc*. In the other words, the words or phrases in one sentence, take on semantic roles such as *agent, patient, theme, source, goal etc.* The predicate-argument structure is the most common semantic representation that is considered as one of the most basic and important representations in linguistics (Hacioglu, 2004). In recent years, this kind of representation has become very popular and a key issue in many applications such as question-answering systems, SRL systems and information retrieval systems (Yoshino, Mori, & Kawahara, 2013). In this structure, a word is considered as predicate and a number of words or phrases are considered as its arguments. Arguments are assigned to different semantic groups depending on the role they play according to the predicate. Semantic roles represent the meaning of the components of the predicate-argument structure and can be defined as domain-dependent (with limited semantic roles) or domain-independent (general). There are a variety of semantic roles and there is no consensus on the set of them (in terms of type and number), because it is the goal that defines a set of roles. For example, for a travel agency that wants to create a question-answering system to help its users know about the time of flights, the required semantic roles will be *patient* (which flight), *time* (when), *source* (from where), *goal* (to where). An example of SRL is shown in Figure 1.

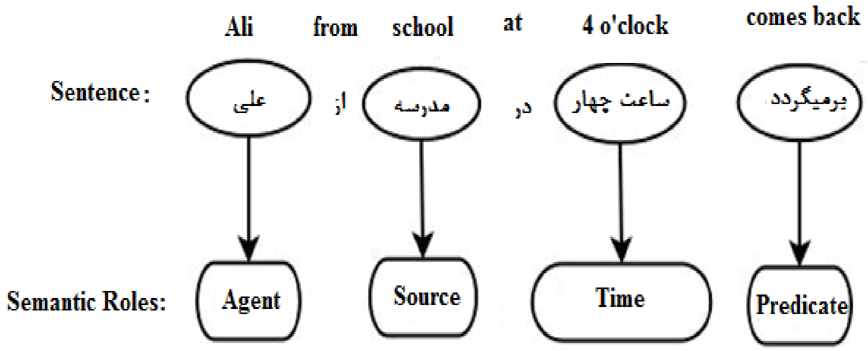


Figure 1. An example of semantic role labeling.

The prerequisite for determining the semantic roles is syntactic parsing (He, Lee, Lewis, & Zettlemoyer, 2017). The information obtained from syntactic parsing plays a crucial role in determining the semantic role label of words. Considering the type of syntactic parsing, SRL can be done phrase-by-phrase (P-by-P), word-by-word (W-by-W) or constituent-by-constituent (C-by-C) (Le-Hong, Pham, Pham, Nguyen, Nguyen & Nguyen, 2017). In P-by-P method, expressions are entered in system as input and phrase-structure-based full syntactic parsing is used. In W-by-W method, labeler system receives single words as input and uses dependency-based full syntactic parsing. In C-by-C method, constituents are entered in system as input and shallow syntactic parsing is used. In Figure 2, samples of phrase-structure-based full syntactic parsing, dependency-based full syntactic parsing and shallow syntactic parsing are shown.

Comparing shallow and full syntactic parsing, researches show that the use of the former increases the accuracy of the system and the use of the latter increases its speed (Johansson & Nugues, 2008), and comparing phrase-structure parsing and dependency parsing, recent researches show better performance of the latter. The advantages of dependency trees over phrase-structure trees can be summarized as follows (Choi & Palmer, 2011; Falavarjani & Ghassem-Sani, 2015):

- Construction time: dependency parsing is performed in a shorter time.
- Structure: dependency structure is more similar to the structure of the predicate-argument.
- Language: for free-word-order languages (such as Persian), dependency parsing has desirable results.
- Type of search: in terms of search method, dependency parser is more similar to semantic role labeling; both of them try to find relations between pairs of words. The major difference between them is in search range; dependency parser tries to find a relation between all the pairs of words while the search of the semantic role labeler is limited to the relation between predicate and arguments.

In general, SRL is accomplished in two ways (Yang & Zong, 2016):

- Rule-based approach: this method analyses semantic roles of sentence components using the rules defined by linguists for a variety of sentences with different grammatical structures. This approach is not efficient because of the variety and complexity of natural language and is used only for specific applications.
- Statistical approach: in this approach, the problem of semantic labeling is considered as a classification problem or a sequence labeling problem. Therefore, the main purpose is to extract features of sentences and to train classifiers to determine semantic labels of sentence components. This approach requires annotated corpus.

In recent years, researchers would like to use deep-learning-based methods that have less dependence on defined features (Furstenau & Lapata, 2009). The main advantage of these methods is their ability to learn features by the network during training phase (Tan, Wang, Xio, Chen, & Shi, 2017).

In this paper, we focus on Persian semantic role labeling with the help of recurrent neural networks and dependency trees. The paper is organized as follows. In the second part, the conducted research works in the field of SRL are discussed. The proposed method is presented in the third section and the fourth section includes experiments and results. Finally, in the fifth section, a summary of the paper is presented with some future works.

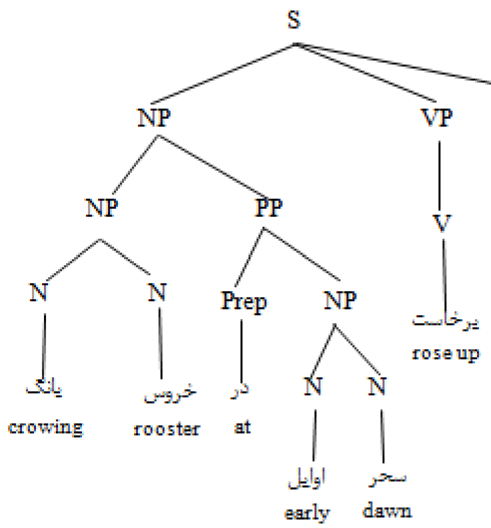
Related works

SRL was first proposed by Gildea and Jurafsky (Gildea & Jurafsky, 2002) for English based on predicate-argument structure. Subsequently, extensive researches were carried out for different languages (such as Chinese, Japanese, etc.). In spite of the fact that SRL has a long history and diverse research in other languages (especially English), it is somewhat new in Persian and limited to a number of recent researches based on the traditional methods (rule-based and machine-learning-based methods). The main reason for this is that Persian belongs to the low-resource languages group.

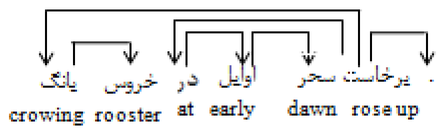
SRL has attracted many researchers to work in the English at the initial years. Gildea & Jurafsky (2002) considered the problem of SRL as a classification problem and sought to find the effective features. After that, the features defined by Xue and Palmer (2004) and Johansson and Nugues (2008) were most welcomed. Among recent studies, Collobert Weston, Bottou, Karlen, Kavukcuoglu & Kuksa (2011) used feedforward neural network, which uses the convolutional function on a window of words. The authors did not use syntactic information to determine the boundaries of constituents. Zhou, Jie, Xu & Wei (2015) used the long-short-term memory (LSTM) recurrent neural network (RNN) for the semantic labeling of English sentences. They designed a multi-layer LSTM network that received text information as input. They took only original word sequence with four simple features (predicate, argument, predicate context and region mark) as input, without using any explicit syntactic knowledge. Henderson and others (Henderson, Merlo, Titov, & Musillo, 2013) used a latent variable model of parsing, the incremental sigmoid belief network architecture. This architecture induces the latent feature representations of derivations which are used to discover correlations both within and between two derivations. Foland and Martin (Foland & Martin, 2015) expanded the proposed model in Collobert et al. (2011) using some features derived from syntactic dependency parse trees to reduce the use of manually extracted features and to make use of unsupervised techniques. They considered role labeling and sense identification as two separate tasks. For each predicate in a given sentence, role-subsystem outputted the list of predicted role tags for all words, and sense-subsystem outputted the sense tag of the predicate. FitzGerald et al. (2015) used the neural network that learns to embed both inputs and outputs in the same vector space. Their model performed separately for each marked predicate in a sentence in which arguments and semantic roles are jointly embedded in a shared vector space for a given predicate. They considered both local and structured training methods for the network parameters from supervised SRL data. Roth and Lapata (Roth & Lapata, 2016) used a logistic regression classifier with a feature set including the predicate word form, Part of Speech (POS) tag and the dependency relationship of all of its children to identify and eliminate predicate ambiguity. To identify and classify arguments, they used a LSTM neural network trained with a dependency parsing tree path and with the features introduced by Xue and Palmer (2004), i.e. lexical-syntactic features, local textual features, a sequence of all POS tags of words between predicate and the argument, and argument position relative to predicate.

For Chinese language, Sun and Jurafsky (Sun & Jurafsky, 2004) did a primitive research without having a large annotated corpus and it had promising results. Yang and Zong (2014) provided multi-predicate SRL method for both English and Chinese languages. They used phrase-structure tree with the features defined by Xue and Palmer (2004) and with a number of features defined by themselves. In the classification step they used a Maximum Entropy (MaxEnt) classifier. Wang and others (Wang, Jiang, Chang, & Sui, 2015) presented a sequence labeling method. Initially, a representation was used for input sentence arguments. The representation was based on the following features: current word, current word POS tag, predicate, its right and left words, its right and left words POS tags, and distance to predicate. The resulted representation had only local features of the word, so the contextual features were added to it in the next step. To do this, nonlinear transmission, bidirectional LSTM RNN, was designed to combine their word information and contextual information into both directions. Finally, for each argument, a k-dimensional vector was obtained each dimension of which represents the corresponding value for each semantic role labels.

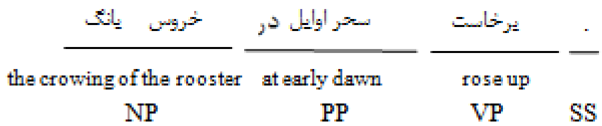
Le-Hong and others (Le-Hong et al., 2017) presented statistical method for Vietnamese language, with regard to its complexity. They used a phrase-structure tree and two categories of feature sets: the first category included the features provided by (Gildea & Jurafsky, 2002). The second one was features defined by authors themselves. SVM and MaxEnt classifiers were used in classification step.



a- Phrase-structure-based full syntactic parsing.



b- Dependency-based full syntactic parsing.



c- Shallow syntactic parsing.

Figure 2. An example of syntactic parsing.

Among the first researches for the Persian, we can mention (Ghalibaf & Rahati, 2009). Their proposed method was based on shallow syntactic parsing. The authors have provided statistical method by examining some of the syntactic and structural features presented for English. To evaluate their method, the authors prepared a corpus consisting of 1300 sentences. The achieved accuracy is reported about 87%. Shamsfard and Mousavi (Shamsfard & Mousavi, 2008) provided a method based on shallow syntactic parsing. The authors firstly specified the range of constituents by using shallow parsing and then used the X-bar theory and constituent ordering patterns to identify the arguments. Some rules were defined by the authors to classify the arguments. The used corpus to perform the experiments was not mentioned. The reported precision is 81.6%. Jafarinejad and Shamsfard (2012) tried to extract the main roles of actor and undergoer. Their method is based on shallow syntactic parsing. Initially, sentence phrases were specified by using shallow parsing. Then the candidate noun phrases and their head were specified in the sentence. Finally, the roles of actor and undergoer were identified by defining a number of rules, based on the sentence structure and verb features. Due to the lack of an annotated corpus, experiments were conducted on the small corpus provided by the authors. The reported precision is 72% and the reported recall is 80%. Saeedi and Faili (Saeedi & Faili, 2012) did SRL by using shallow syntactic parsing and memory-based method. Their goal was to create a suitable feature set for achieving high accuracy. For this purpose, authors defined three categories of features: chunk content features, verb features, link features. Finally, the semantic roles of the arguments were determined with an instance-based classifier. For training and testing the classifier, a part of the Bijankhan corpus (Bijankhan, 2004) was annotated semantically by the authors manually. The F_1-measure was reported 66.12%. (Rezaei Sharifabadi & Khosravizadeh, 2016) is the first research that used a dependency parser to label semantic roles. Dependency parsing was performed by using the Maltparser tool and semantic label of words was determined by using the Naïve Bayes (NB) and MaxEnt classifier. The used features were obtained reviewing the features introduced in (Johansoon & Nugues, 2008) and (Gildea & Jurafsky, 2002). The required corpus for experiments was prepared by the first author and contains 1,000 sentences with their dependency trees and 50 frequent verbs. Accuracy reports are 76% and 84% by using the standard dependency tree and 63% and 69% by using automatic dependency tree for NB and MaxEnt classifiers respectively.

Proposed method

In this paper, we develop a deep learning model that can be applied to the task of Persian SRL. The main objective of this model is to use dependency trees only with minimal feature engineering. To this end, for each word of a given sentence, at first we transform it into continuous vector representations by using word embedding method. Then, for each word, the lexical feature vector is computed. At the next layer, the LSTM network extracts dependency path feature vector from each pair of predicate and its potential argument. The extracted features are concatenated and passed into Bi-LSTM. Finally we construct our neural network model by feeding the output vectors of Bi-LSTM into a fully connected network and a softmax layer. Figure 3 illustrates the architecture of our network in detail.

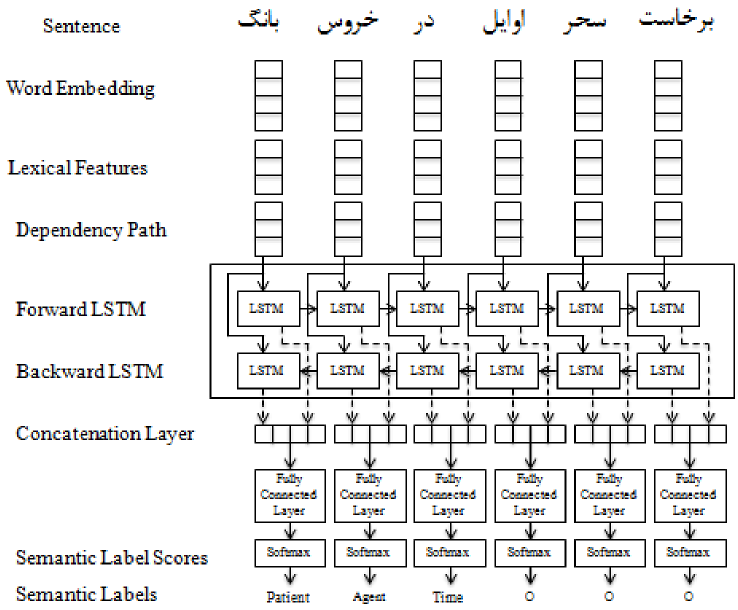


Figure 3. An example of Persian SRL using our proposed model.

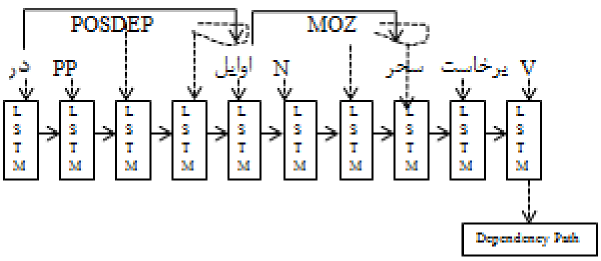


Figure 4. An example of dependency path feature extraction using LSTM network of pair of predicate and its candidate argument (در، برخاست) from sentence shown in Figure 2-b.

Candidate argument identification

As we know, not all the words of a sentence are related to the verbal predicate, or, in other words, a limited number of words in a sentence are the verbal predicate's arguments. The purpose of this step is to determine the arguments and their range. To achieve this goal, we have used dependency tree, which has led to word-to-word SRL; therefore, the range of arguments is limited to words.

In the dependency parsing, the dependency structure is determined for a sentence. Dependency parsing provides important information about the relationship of words. In fact, dependency trees carry the dependency connection between sentence words. For example, as shown in Figure 2-b, in the dependency tree, the words associated with each word are identified. We have used dependency relations to identify candidate arguments. So, we consider heads and dependents of verbal predicate as candidate arguments with respect to its dependency tree.

3.2. Core features

Word embedding and lexical features

In this step, for the sentence $S = \{x_1, x_2, \dots, x_n\}$ with the arguments $A = \{a_1, a_2, \dots, a_m\}$ and the semantic roles $R = \{r_1, r_2, \dots, r_m\}$, each word x_i is mapped to a lookup table by applying word embedding process. At this point, we use the word2vec algorithm (Mikolov, Chen, Corrado, & Dean, 2013). This algorithm is provided by Google in 2013 to convert words into word vectors.

This algorithm maintains the semantic similarity of words. With this algorithm, we can transmit text to a k-dimensional space and calculate the semantic similarity of words.

In addition, each word also is represented as a vector containing lexical features. Lexical features are low-level features that are obtained without any need to sentence parsing. The lexical features used in this paper are inspired by the works of Hacioglu (Hacioglu, 2004) Gildea and Jurafsky (Gildea & Jurafsky, 2002) and include:

- **Position:** indicates the position of the word relative to the verbal predicate whether it is located before or after it. The word position contains useful information to determine its semantic role. For example, the role of the *agent* usually occurs before the verb. However, Persian is a verb-final language; at the same time is a free-word-order language. Therefore this feature will be useful in the exceptions (the subject-object-verb order is ignored).
- **Distance:** indicates the number of the components between the predicate and the argument. Usually, the distance between the *agent* and the predicate is greater than that between the *concerning* and the predicate.
- **Voice:** indicates the verb to be active or passive. The verb form can determine the presence or absence of some arguments. On the other hand, the verb form can be very useful in recognizing some semantic roles. For example, the object of the active form of a verb has the same semantic role as the subject of its passive form.

Dependency path

In this step, the dependency path vector is computed for each pair of predicate (x_i) and its candidate argument (x_j). For each predicate/candidate-argument we employ a forward LSTM network that takes a sequence from candidate argument to predicate ($X=x_j, x_{j+1}, \dots, x_i$) as input, with each input step representing a binary indicator for an edge direction to the next word (x_{j+1}), an edge label (syntactic role) to the next word (x_{j+1}), word POS tag (x_j^{pos}), word embedding (x_j^{form}), next word embedding (x_{j+1}^{form}), and next word POS tag (x_{j+1}^{pos}), and finally outputs one embedding state called dependency path vector. The next word (x_{j+1}) must be a head (or dependent) of candidate argument or a head (or dependent) of the current word (x_j) if and only if the current word (x_j) is the head (or dependent) of candidate argument. This feature is only computed if the word is one of the candidate arguments. Figure 4 shows an example of the LSTM network we use to extract dependency path representation of a pair of words.

The main reason of using the mentioned features (edge direction to the next word, edge label to the next word, POS tag and word form) is that they are useful to distinguish the semantic roles. The edge label (type of dependency relation), edge direction and word category determine which roles are possible and what kinds of path are to be expected. For example, the probability that the argument in prepositional dependency relationship be *source* and *goal* is much more than that be *agent*.

Semantic Role labeling

In this layer, the extracted features of each word are concatenated and passed into forward and backward LSTM networks. The LSTM network learns long dependencies and allows the use of information far from the current word for labeling (Zhou et al., 2015). Due to the fact that in SRL both previous and subsequent dependencies are important, access to the right and left side information of the word is required. The Bi-directional LSTM (Bi-LSTM) network allows us to do this. Bi-LSTM propagates in two directions using parallel layer of forward and backward LSTM. The output of forward and backward layers is concatenated to form the final output.

Finally, the output of each Bi-LSTM network at each time step is decoded by a fully connected network and a softmax layer into probabilities for each semantic category.

Experiments

Corpora

For the experiments two corpora have been used: 1. *The First Semantic Role Corpus in Persian Language* (Mirzaei & Moloodi, 2014), 2. A small corpus prepared by the authors (Lazemi, Ebrahimpour-Komleh, & Norozi, 2018b). The first corpus contains 29983 sentences in contemporary Persian language, which have manually annotated *based on the concept of thematic roles of Fillmore*, with 27 semantic roles in three stages. This corpus has added a semantic layer to the Persian Syntactic Dependency Treebank (which is a collection of sentences with their corresponding dependency trees, including 44 types of syntactic roles and 49 types of POS tags) (Rasooli, Kouhestani, & Moloodi, 2013). The used semantic roles include two groups of *thematic roles and functional tags*, such as: *agent, patient, theme, experiencer, instrument, location, source, goal, time, condition, cause,* and *productive*. The second corpus has added a semantic layer to the Uppsala Persian Dependency Treebank (UPDT, which is a collection of sentences with their corresponding dependency trees, including 48 types of syntactic roles and 31 types of POS tags) (Seraji, Ginter, & Nivre, 2016). This corpus contains verbs classified according to their syntactic and semantic capacities, provided by Ghalibaf and Rahati (Ghalibaf & Rahati, 2009). For each verb, 20 simple sentences are extracted from the UPDT and manually annotated with Palmer's (Palmer, Gildea, & Xue, 2010) proposed semantic roles (including *agent, patient, theme, experiencer, instrument, location, source* and *goal*).

The statistical information of the corpora and the frequency of the semantic roles are reported in Table 1 and Table 2, respectively.

The corpora are individually split into train, development and test sets by using of 10-fold cross validation method. 80% of the data is used for training, 10% is used for developing and 10% is used for testing. In both corpora, we focus on labeling arguments with *agent, patient, theme, experiencer, instrument, location, source* and *goal* tags.

Network training and Hyper-parameter initialization

We implement the neural network using the UKPLab[1]. Training is performed with mini-batch stochastic gradient descent (SGD) with a fixed learning rate. Also, we explored AdaGrad, AdaDelta, RMSProp, Adam and Nadam optimization algorithms, but they did not improve upon SGD.

Table 1

Statistical properties of the used corpora.

	First Corpus	Second Corpus
Number of sentences	29982	940
Number of distinct verbs	62889	17
Number of verbs	9200	940
Average sentence length	16.61	9.23
Coarse-grained POS tags	17	13
Fine-grained POS tags	32	18

Table 2.

Frequency of semantic labels in used corpora.

	Frequency%	
Semantic role labels	First Corpus	Second Corpus
Agent	18	30.7
Patient	25.4	10.58
Theme	11.9	2.92
Experiencer	4.4	0.72
Instrument	-	1.67
Location	-	7.14
Source	-	5.47
Goal	4.6	9.89

In order to reduce overfitting, we apply the dropout method (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

We use the word2vec algorithm with Skip-gram and CBOW models to create each vector. In order to train CBOW model, we use the corpus introduced in our previous work (Lazemi, Ebrahimpour-Komleh, & Noroozi, 2018a), but Skip-gram model has a pre-trained form (ibid).

Hyper-parameters are selected by using the development sets by random search. We evaluate 100 hyper-parameter settings. Table 3 summarizes the chosen hyper-parameters for all experiments and Table 4 shows the development set performance of the best setting.

Evaluation metrics

To evaluate the performance of proposed method, precision, recall and F_1-measure metrics are used and defined as Formulas 1, 2 and 3.

$$Precision = \frac{number\ of\ correctly\ labelled\ arguments}{number\ of\ detected\ arguments} \quad (1)$$

$$Recall = \frac{number\ of\ correctly\ labelled\ arguments}{number\ of\ arguments} \quad (2)$$

$$F_1 - measure = \frac{2 \times\ Precision \times Recall}{Precision + Recall} \quad (3)$$

Results and discussion

Table 4, compares the performance of different word embeddings in our model. According to Table 4, the Skip-gram vector representation has better performance than CBOW.

Tables 5 and 6 demonstrate the results obtained from applying the proposed method with Skip-gram model for different role labels, for the first and second corpora respectively. The obtained poor results for some of the labels can be considered as the result of the inadequacy of training data belonging to the corresponding class.

In order to analyze and evaluate the impact of each type of features on the performance of our method, we test our system without specific types of features. The results are shown in Tables 7 and 8. Table 7 confirms that the combination of features increases the accuracy of the system by about 12% of F1 score. As shown in Table 8, the use of dependency path feature doesn’t show a large and significant improvement. On the other hand, according to Tables 6 and 8, the combination of features doesn’t improve significantly the overall results.

Table 3
Hyper-parameters of our model.

Parameter		First Corpus	Second Corpus
	Range	Final	
Word embeddings	-	50	50
POS tags	-	17	13
Syntactic label	-	44	48
LSTM layers	-	2	2
LSTM state size	[100,400]	150	150
Learning rate	$[10^{-3},10^{-1}]$	0.041	0.05
Dropout rate	[0,1]	0.67	0.52
Mini-batch size	[5,12]	10	6
Neuron	[50,150]	100	100

Table 4
Development set performance of the best parameter settings.

		Precision	Recall	F1
First corpus	CBOW	80.86	83.94	82.37
	Skip-gram	79.27	86.56	82.75
Second corpus	CBOW	86.15	85.85	85.99
	Skip-gram	89.12	84.36	86.67

Table 5
Results on the first corpus.

Semantic role labels	Precision	Recall	F1
Agent	86.53	84.72	85.61
Patient	83.30	82.55	82.92
Theme	81.82	87.96	84.77
Experiencer	74.25	67.37	70.64
Instrument	73.43	74.08	73.75
Location	83.52	85.72	84.60
Source	81.63	85.38	83.46
Goal	77.00	71.88	74.35

Table 6
Results on the second corpus.

Semantic role labels	Precision	Recall	F1
Agent	94.02	93.39	93.70
Patient	89.04	91.35	90.18
Theme	78.28	76.61	77.43
Experiencer	65.95	60.37	63.03
Instrument	69.18	68.28	68.72
Location	92.28	94.03	93.14
Source	88.10	86.98	87.53
Goal	86.98	85.33	86.14

Figures 5 and 6 demonstrate the impact of dependency path on the sentences with varying length, for the first and second corpora respectively. As shown in Figure 5, it doesn't give good improvement for short sentences, but it can be observed that the differences between results are increasing by adding the number of words in the sentences, and it gives good performance in the long sentences using dependency path. The reason for this can be searched in the handling of complex structure of long sentences by dependency path.

In Figure 6, the dependency path feature also helps increase slightly the F1 score. This is mainly due to simple and short sentences in second corpus.

Comparing Figures 5 and 6, we can see that complex structure of long sentences is handled by using dependency path. Therefore dependency path can be used as an effective feature in Persian SRL.

Table 7
Accuracy of feature sets on the first corpus.

	Precision	Recall	F1
Emb	69.69	67.85	68.75
Emb+Lex	77.20	68.54	72.61
emb+lex+Dep Path	80.18	79.95	80.01

Table 8
Accuracy of feature sets on the second corpus.

	Precision	Recall	F1
Emb	81.93	82.78	82.35
Emb+Lex	83.04	79.48	81.22
Emb+Lex+Dep Path	82.97	82.04	82.48

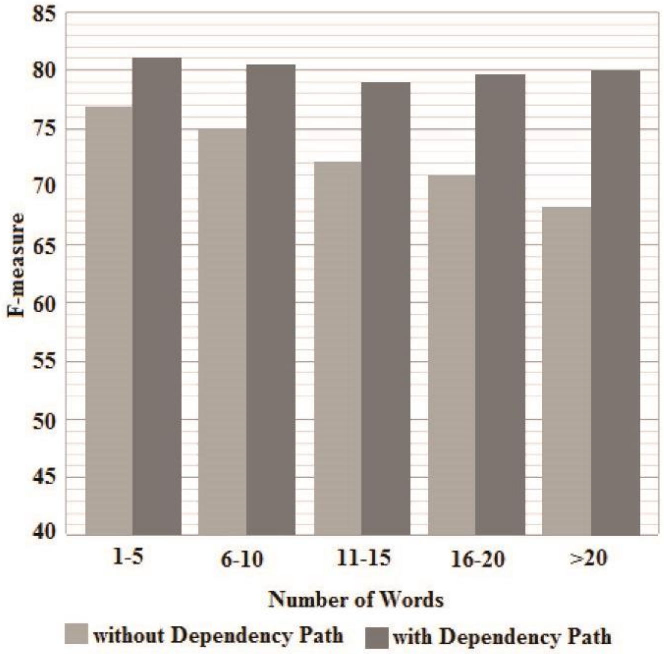


Figure 5. Results on first corpus by sentence length**.**

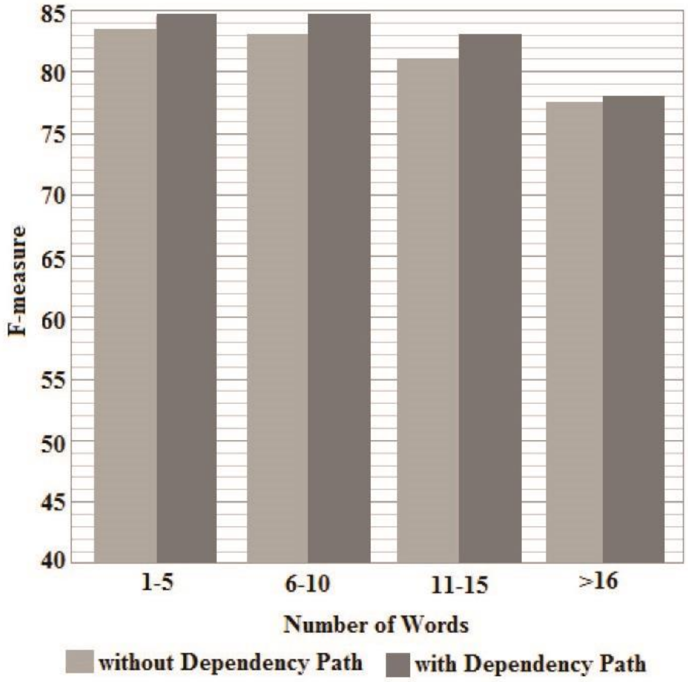


Figure 6. Results on second corpus by sentence length.

Conclusion

In this paper, a deep-learning-based method for the semantic role labeling of Persian sentences was developed. Existing methods consider the SRL of Persian sentences as a classification problem. Therefore, their models are in need of manual feature extraction of syntactic parse tree of sentence. On the other hand, their proposed models ignore the dependencies between words in a sentence, despite the fact that they are related to each other. Our proposed method classified arguments verbal predicate by using a hierarchy of feature extraction in a deep structure neural network. It tried to improve the results with the help of dependency trees and also to capture long-range dependencies in a sentence with little feature engineering.

Experiments showed that the injection of dependency path handled complex structure of long sentences and improved the performance of proposed method too. But, because there is no published corpus except that we have used (which has been recently published), and because other researchers have used their own unpublished corpora, so there is no possibility of comparison between our research and the others.

Unfortunately, there is a very big gap between researches done for Persian and other languages in the domain of SRL; the main reason for this is the previous lack of a labeled semantic corpora. With the availability of the first Persian semantic corpus, it is hoped that this area will attract many scholars.

Tags in sequence labeling methods are allocated not only based on local information but also on long-distance dependencies; therefore, considering the nature of the Persian (a lot of flexibility in the position of the words), it is strongly recommended to consider the problem of semantic role labeling in Persian as a sequence labeling problem. Since in Persian, most of sentences contain several predicates, to pay attention to semantic role labeling in multi-predicate sentences can strongly be suggested as future work. In multi-predicate sentences, a word may have more than one semantic role; so eliminating the ambiguity of these types of words can improve the accuracy of the labeling system. In this paper, we also focused on a limited number of semantic roles; so the extension of our model by using extended semantic role set can lead to better results.

Endnote

1. https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf

References

Bijankhan, M. (2004). The role of the corpus in writing a grammar: An introduction to a software. *Iranian Journal of Linguistics*, 19(2), 48-67. [in Persian]

Choi, J. D. & Palmer, M. (2011). Transition-based semantic role labeling using predicate argument clustering. In Proceedings of the ACL 2011 Workshop on Relational Models of Semantics (pp. 37-45). Association for Computational Linguistics

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12, 2493-2537.

Falavarjani, S.A.M. & Ghassem-Sani, G. (2015). Advantages of dependency parsing for free word order natural languages. In: Italiano G.F., Margaria-Steffen T., Pokorný J., Quisquater J.J., Wattenhofer R. (Eds.) SOFSEM 2015, *Theory and Practice of Computer Science*. SOFSEM 2015. Lecture Notes in Computer Science, vol 8939. Springer, Berlin, Heidelberg

FitzGerald, N., Täckström, O., Ganchev, K. & Das, D. (2015). Semantic role labeling with neural network factors. In Proceedings of *the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 960-970)*.

Foland, W. & Martin, J. (2015). Dependency-based semantic role labeling using convolutional neural networks. In Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (pp. 279-288).

Fürstenau, H. & Lapata, M. (2009). Semi-supervised semantic role labeling. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (pp. 220-228). Association for Computational Linguistics.

Ghalibaf, A. K., Rahati, S. & Estaji, A. (2009). Shallow semantic parsing of persian sentences. In Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Vol. 1, (pp. 13-22).

Gildea, D. & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational linguistics*, 28(3), 245-288.

Hacioglu, K. (2004). Semantic role labeling using dependency trees. In Proceedings of the 20th international conference on Computational Linguistics (pp. 1273-1277). Association for Computational Linguistics.

He, L., Lee, K., Lewis, M. & Zettlemoyer, L. (2017). Deep semantic role labeling: What works and what's next. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 473-483).

Henderson, J., Merlo, P., Titov, I. & Musillo, G. (2013). Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational linguistics*, 39(4), 949-998.

Jafarinejad, F. & Shamsfard, M. (2012). Extracting generalized semantic roles from corpus. *International Journal of Computer Science,* 9(2), 200-206.

Johansson, R. & Nugues, P. (2008). The effect of syntactic representation on semantic role labeling. In Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1 (pp. 393-400). Association for Computational Linguistics.

Lazemi, S., Ebrahimpour-Komleh, H. & Noroozi, N. (2018a). Persian plagirisim detection using CNN. In *the 8th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 171-175). IEEE.*

Lazemi, S., Ebrahimpour-Komleh, H. & Norozi, N. (2018b). Shallow semantic parsing of Persian sentences via phrase-structure parsing. *In Proceedings of the 4th National Conference on Computational Linguistics* (pp.83-104). Masoud Qayyum & Azin Shahriari Fard (Eds). Tehran: Neviseh Parsi

Le-Hong, P., Pham, T. H., Pham, X. K., Nguyen, T. H., Nguyen, T. L. & Nguyen, M. H. (2017). Vietnamese semantic role labelling. *VNU Journal of Science: Computer Science and Communication Engineering,* 33(2), 39-58.

Mikolov, T., Chen, K., Corrado, G. & Dean, J.. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:*1301.3781.

Mirzaei, A., & Moloodi, A. S. (2014). The first semantic role corpus in Persian language. *Language Science,* 2(3), 29-48. [in Persian]

Palmer, M., Gildea, D. & Xue, N. (2010). Semantic role labeling. *Synthesis Lectures on Human Language Technologies,* 3(1), 1-103.

Rasooli, M. S., Kouhestani, M, & Moloodi, A. (2013). 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 (pp. 306-314).

Rezaei Sharifabadi, M. & Khosravizadeh, P. (2016). Automatic labeling of semantic roles in Persian sentences using dependency trees. *Signal and Data Processing*, 13(1), 27-38.

Roth, M. & Lapata, M. (2016). Neural semantic role labeling with dependency path embeddings. *arXiv preprint arXiv:1605.07515*.

Saeedi, P. & Faili, H. (2012). Feature engineering using shallow parsing in argument classification of Persian verbs. In The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012) (pp. 333-338). IEEE.

Seraji, M., Ginter, F. & Nivre, J. (2016). Universal dependencies for persian. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (pp. 2361-2365).

Shamsfard, M. & Mousavi, M. (2008). Thematic role extraction using shallow parsing. *International Journal of Computational Intelligence*, 4(2), 126-132.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.

Sun, H. & Jurafsky, D. (2004). Shallow semantc parsing of Chinese. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004 (pp. 249-256).

Tan, Z., Wang, M., Xie, J., Chen, Y. & Shi, X. (2018). Deep semantic role labeling with self-attention. In Thirty-Second AAAI Conference on Artificial Intelligence.

Wang, Z., Jiang, T., Chang, B. & Sui, Z. (2015). Chinese semantic role labeling with bidirectional recurrent neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1626-1631).

Xue, N. & Palmer, M. (2004). Calibrating features for semantic role labeling. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (pp. 88-94).

Yang, H., & Zong, C. (2014). Multi-predicate semantic role labeling. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 363-373).

Yoshino, K., Mori, S. & Kawahara, T. (2013). Predicate argument structure analysis using partially annotated corpora. In Proceedings of the Sixth International Joint Conference on Natural Language Processing (pp. 957-961).

Zhou, J. & Xu, W. (2015). End-to-end learning of semantic role labeling using recurrent neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 1127-1137).