Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 227 – 233

ISBN 978-83-60810-22-4
ISSN 1896-7094

Developing A Persian Chunker Using a Hybrid Approach

Soheila Kiani
NLP Research Lab.,
Elecrical & Computer
Engineering Dept., Shahid
Beheashti university,
Tehran, Iran
Email: soha_kn@yahoo.com

Tara Akhavan
NLP Research Lab.,
Elecrical & Computer
Engineering Dept., Shahid
Beheashti university,
Tehran, Iran
Email:t_a_8564@yahoo.com

Mehrnoush Shamsfard
NLP Research Lab.,
Elecrical & Computer
Engineering Dept., Shahid
Beheashti university,
Tehran, Iran
Email: m-shams@sbu.ac.ir

***Abstract*—Text segmentation is the process of recognizing boundaries of text constituents, such as sentences, phrases and words. This paper focuses on phrase segmentation also known as chunking. This task has different problems in various natural languages depending on linguistic features and prescribed form of writing. In this paper, we will discuss the problems and solutions especially for the Persian language and present our system for Persian phrase segmentation. Our system exploits a hybrid method for automatic chunking of Persian texts. The method at first exploits a rule-based approach to create a tagged corpus for training a neural network and then uses a multilayer perceptron neural network and Fuzzy C-Means Clustering to chunk new sentences. Experimental results show the average precision of %85.7 for the chunking result.**

I. Introduction

SEGMENTATION is one of the primary activities in natural language processing and includes fields of text segmentation, phrase segmentation and word segmentation. Text segmentation is a general term related to any activity which determines the boundaries of text constituents. It may detect paragraphs, sentences, phrases and words.

Phrase segmentation is related to determining the boundaries of groups and phrases in a sentence (sentence constituents). A segmenter which detects syntactic constituents (such as noun phrase and verb phrase) is called a chunker. Correct chunking may facilitate text processing activities in NLP applications such as machine translation, information retrieval, question answering, thematic role extraction and etc. These phrases are non-overlapping, i.e., a word can only be a member of one chunk. It provides a key feature that helps on more elaborated NLP tasks such as parsing and Information Extraction [1].

Phrase extraction from sentences is one of the most important parts of text analysis. Various chunking systems have been developed for different languages. Persian language due its special features such as omission of Ezafe marker[1] and various writing prescriptions is a language with challenging chunking. There is neither a Persian corpus with phrase tags (e.g. IOB tags) nor a reliable Persian chunker available. To solve these problems, in this paper, an automatic phrase chunking system is presented for Persian. In this system, chunks' information in a sentence is represented by IOB tags.

In our work, at the first stage, an IOB tagged corpora is constructed using a rule based approach, exploiting some rules manually extracted for Persian language chunking. Then a multilayer perceptron neural network with 2 hidden layers is used to train the system. To improve the results, a fuzzy C-means clustering method is applied on neural network output data. The proposed method is the first learning approach which is used for the Persian language and improves the results obtained from rule based method presented for IOB tagging in [2, 7].

In the rest of the paper we will first describe some related works. Then after discussing some effective factors on phrase chunking in various languages, we will present our proposed chunking system in detail. The implemented tool is described in section V. Section VI illustrates the experimental results. The last section concludes the implemented method and its results.

II. Related Works

Various approaches are used in phrase segmentation. Each of them has some advantages and disadvantages in comparison with the others. Many systems utilize combinations of two or more approaches to increment the accuracy. The most popular approaches are discussed here.

A. Rule based methods

Rule based methods need linguistic knowledge consisting of both semantic and syntactic elements. The rules may be defined by human or extracted from linguistic resources

[1] Ezafe Marker is the sign of Ezafe construction. It is used to relate nouns to their adjective and noun modifiers. It is a short vowel which is usually pronounced but not written.

such as tagged corpora using a learning procedure. Accuracy incrementation is hard in these approaches [13].

Shamsfard and SadrMousavi [2], [7] presented a rule-based semantic role labeling system for Persian sentences. The system exploits a two-phase architecture to (1) identify the arguments and (2) label them for each predicate. For the first phase a rule based shallow parser is developed to chunk Persian sentences and for the second phase a knowledge-based system is developed to assign 16 selected thematic roles to the chunks. The main restriction of this work is that the Ezafe markers should be written explicitly in the sentences. As there is no tagged corpus for Persian, evaluation of the results is done manually.

B. Statistical approaches

Unlike rule based methods, these approaches do not need linguistic knowledge and their success highly depends on the resources. Statistical approaches are more portable than rule based approaches. They are shallow but cover more comprehensive width. The other advantage of statistical approaches is that they are not language specific and can be applied on languages with common features.

These methods should extract statistical information from processed corpus, web documents, search engine outputs, etc. The extracted statistical information consists of high frequently phrases, their frequency of occurrence, phrase occurrence and co-occurrence probability, etc.

As an example we can mention the work done by Diab and her colleagues [3]. They presented a Support Vector Machine (SVM) based approach to automatically do tokenization, part-of-speech (POS) tagging and annotating base phrases (BPs) in Arabic texts. Their system was trained with the Arabic TreeBank corpus.

C. Learning methods

In these methods, systems learn required segmentation information from input sources. This information can be linguistic models, semantic and syntactic rules or statistical information. In other words learning methods may combine with each of approaches that are mentioned above. Learning resources are generally lexicons and corpora. Segmented syntactically tagged corpora are one of the most appropriate linguistic resources for segmentation learning.

However, these methods properly handle new cases, but the lack of appropriate tagged corpora makes use of these methods difficult and inefficient.

Yousif and colleagues [4] have implemented a 2-layer Multilayer perceptron (MLP) method with one hidden layer to train a system to POS tag Arabic texts.

In references [5], [6] a hybrid method for tagging Arabic texts was presented, It firstly used a rule-based method and then a memory-based method to tag. The memory-based method was based on K-NN method.

Milidiu et. al. applied the Entropy Guided Transformation Learning (ETL) to four phrase chunking tasks: Portuguese noun phrase chunking, English base noun phrase chunking, English text chunking and Hindi text chunking and in all four tasks got better results than Decision Trees [1]. ETL is a new machine learning strategy that combines the advantages of decision trees (DT) and Transformation Based Learning (TBL) and only requires the training set and no handcrafted templates. ETL also simplifies the incorporation of new input features, such as capitalization information, which are successfully used in the ETL based systems. This model (ETL) is also used in [14] for three Portuguese Language Processing tasks: Part-of-Speech Tagging, Noun Phrase Chunking and Named Entity Recognition.

As it can be seen there are some taggers and chunkers developed for the Arabic language with high accuracies. Although, the Arabic and Persian Alphabets are so similar, the grammar and structure is completely different. They come from different language families[2]. Therefore in Persian we can not reuse Arabic chunking systems / algorithms and a special tagging system for Persian is needed.

Many other chunking algorithms available for other languages can not be used for Persian too, as there is no tagged corpus. So the best solution is either to use rule based methods which do not need a corpus or creating a chunking tagged corpus and then use it to chunk sentences. The first approach has problems with Ezafe construction. So it seems that the best approach is the second one.

III. Effective Factors on Phrase Chunking

Punctuation marks, grammatical rules, verbs, function words, prepositions and vowels are some of the linguistic information which affect on phrase chunking. We explain these resources briefly in the following subsections.

A. Punctuation marks

Punctuation marks determine sentence boundaries and can help text segmentation. Punctuation marks play an important role in text to speech synthesis; because these marks make differences in pronunciation of phrases and sentences. For example a sentence that ends with a stop or exclamation, will be pronounced in different ways. One method of determining phonological phrases is considering the presence of punctuations that assign an intonation contour to each group of words which falls between punctuation marks. This method is less satisfactory for sentences which contain little or no punctuation.

In many languages like English, punctuation marks exist but in some languages like Vietnamese they do not [8].

In Persian, the exclamation and the interrogation marks are unambiguous boundary. On the other hand, the stop is an ambiguous boundary indicator as, it marks a sentence boundary, and may also appear in the formation of abbreviations or acronyms. Apart from the slash (/), which is used in numbers, and the dash, which could be used to separate compound words, the other punctuation marks unambiguously indicate word boundaries. These include the comma, quotes, brackets and colon [10][3].

[2] Arabic is the largest member of the Semitic branch of the Afro-Asiatic language family (classification: South Central Semitic) and is closely related to Hebrew and Aramaic but Persian is an Indo-European language; it is part of the Iranian branch of the Indo-Iranian language family.

[3] In this paper we do not consider the problem of tokenization which focuses on detecting the boundaries of words. Tokenization is a challenging problem in Persian as space is not a deterministic word boundary and there are

B. Grammatical rules

Grammatical rules may be used to parse a sentence and extract its phrases. Parsing can be performed in two ways: shallow parsing or deep parsing. Deep parsing can find a complete syntactic structure by means of a grammar and parsing algorithm, but it has high complexity and needs a predefined complete set of grammatical rules. In shallow parsing a low level syntactic structure is assigned to the input sentence and extracts constituents related to specific parts of speech.

In Persian language, exploiting shallow parsing is preferred for two important reasons. First, there is no general and standard computational grammar which covers all sentences and second, it has free word order behavior.

C. Verbs

Verbs are key elements in a sentence. The number of arguments of a verb can be used in phrase chunking. Determining the number of essential arguments is itself one of the complex problems in language processing. Before determining the number of arguments, the verb in the sentence should be detected. In case of complex verbs or different derivational forms, segmentation and morphology analysis will be required to find the verb.

In many cases the number of verb arguments is used for shallow parsing disambiguation [9]. For example, consider the following sentences:

"نامه علی به حسن گم شد" -'Name-ye Ali beh Hassan gom shod'
(The letter of Ali to Hassan was lost) and

"نامه علی به حسن رسید" 'Name-ye Ali beh Hassan Resid'
(The letter of Ali Arrived to Hassan)

They contain the same words except their verbs. At the first sentence (The letter of Ali to Hassan) is a noun phrase and in the second one (The letter of Ali) is a noun phrase and (to Hassan) is a prepositional phrase. The verb just has one essential argument in the first sentence while in the second one it may have more than one argument and so chunking of noun phrase and prepositional phrase is correct.

If the verbs of the input sentences can be reliably detected then a Case Grammar/Thematic relations approach could be incorporated in the parser. This approach would focus on the verb as the central element in the sentence and look for the roles associated with the verb, such as Beneficiary, Object, and Location [8].

D.Function words

Most languages contain words that can be placed only at the first or end of a phrase. These words generally are determiners, prepositions, conjunctions, disjunctions; and personal, possessive, and interrogative pronouns.

Function words especially prepositions play the most effective role in Persian phrase boundary detections.

As in structure of some languages like English, Persian and Spanish, prepositions are usually placed at the first of a phrase; it is possible to use them to chunk phrases.

In most cases prepositions can be used to not only determine phrase boundaries but also to determine the phrase type and its semantic role in a sentence.

There are some exceptions in prepositions. For example the preposition de 'of/from' which is the most frequent function word in Spanish does not lead the general rule and may be seen inside a noun phrase [11]. Such circumstances may appear in Persian too, especially for proposition "az" (to) [9].

E. Vowels

Existence of various phonetics is one of the major challenges of segmentation and chunking in Asian languages in comparison with Indo- Europian languages.

The lack of representation of Ezafe construction in the Persian texts creates ambiguity in chunking and phrase pronunciation. For example in sentence "مرد دانشمند را دید" there is no written Ezafeh sign; so it may be pronounced and chunked as 'mard daneshmand ra did' (The man saw the scientist) or be processed as 'mard –e- daneshmand ra did' (He/she saw the scientist man). Ezafeh is a vowel added (pronounced –e but not written) to join parts of a noun phrase. It does not change the semantic of its surrounding words but it has its own semantic and changes the head and boundaries of noun phrases [7]. It may be equivalent to "'s" as in "کتاب علی", 'ketab –e Ali' (Ali's book), equivalent to 'of' as in "در باغ", 'dar-e bagh' (door of the garden) or has no equivalent in English as in "کیف مدرسه", 'kif-e madreseh' (school bag).

Attachment of Ezafeh causes addition of enclitic when the words end with some special characters (Such as ا، ه، و). Although these explicit enclitics may facilitate the Ezafeh detection and consequently eases chunking but they have again some different form of writing which need some processing to recognize. For example "خانه علی", 'khane –e Ali' (Ali's House) may be also written as "خانه ی علی" or "خانـة علـی" while pronouncing the same. All these writing forms may vary by adding or ignoring (short) spaces between parts.

An analogous problem exists in the Arabic script and "e" vowel is not usually represented in newspapers and books (except for religious texts or elementary education). In the modified Arabic script of Kurdish, a new letter has been added to the alphabet for representing the "e" vowel. Vowels such as "a" and "o" also has a corresponding letter in the modified alphabet in Kurdish Arabic-script. "a" and "o" vowels do not play a crucial disambiguation grammatical role in Kurdish or Persian. But in Arabic "a" (fatha) and "o" (zamma) are used for grammatical marking of object and subject. This kind of grammatical ambiguity can be captured by a language model and using semantics in the parsing stage [12].

IV. Our Proposed Chunking Method

The proposed method uses a combination of a multilayer perceptron neural network and fuzzy C-Means clustering to perform automatic phrase chunking.

several writing prescriptions with different spacing rules. Here we assume that the text is tokenized. For more details on Persian tokenization readers can refer to [].

Phrase chunks in a sentence are represented by IOB tags. As it was mentioned before, a tagged corpus is needed to train the network. As there was not a tagged corpus for Persian which can be used in this work, an IOB tagged corpus is firstly constructed using a rule based system.

A. IOB tagging

Chunk information in a sentence can be represented by means of tags. The bracket style and IOB tag set are the two common tagging styles. Bracket style is the simplest case in which the start and end of phrases are limited within brackets. The following sentence is marked using brackets.

[in ketab NP] [bist safheh NP] [darad VP].

[This book NP] [has VP] [twenty pages NP].

IOB tagging is used to represent phrase chunks in this paper. Each token is tagged with one of three specific chunk tags, I (inside), O (outside), or B (begin) in this tagging style. If a token marks the beginning of a chunk, it will be tagged as B. Subsequent tokens within the chunk are tagged as I and the remaining tokens are tagged as O.

We preferred IOB tagging over bracket notation as its outputs can be efficiently applied in the different machine learning techniques

B. Constituent ordering and rule extraction

To identify phrase boundaries about 60 patterns (rules) for constituent ordering of Phrases in Persian were used, as well as a description of their structure. These patterns show how the lexical information presented in the sentence could be used in determining the boundaries of the phrases. Since there is not a character to distinguish the last token of a phrase from other inner constituents in IOB form, the start of a new phrase will be the end of current phrase.

The exploited rules can be divided into two groups. The first are the rules that determine if the token is inside the phrase and the next are those that determine the beginning token of a phrase regardless of the phrase type. In other words, these rules can be considered as two classes, the rules that mark the token as I or the rules that mark it as B.

A sample rule which is extracted from PP's constituent ordering is introduced as follow:

Persian prepositional phrases are easily recognized and can be used to mark phrasal boundaries in the sentence. The following structure describes the constituent ordering of PP.

- PP: preposition + NP

The headword of a PP is a preposition, which is always followed by an NP. This structure shows that detecting the start of a PP is not difficult. The following rule shows that if the current token is a preposition the next token certainly will be in the PP structure.

- IF POS (X) = P then IOB-tag (X+1) = I

But the preposition itself is not always the beginner of the phrase. There may be an identifier before it in the prepositional phrase (e.g. 'hatta dorost dar khiaban' (even right in the street)). Following is a sample rule to handle such cases.

- IF POS(X) =P and X-1 _generalID then IOB-tag(X) =I

In some cases with ambiguities, assigning the IOB tag is not easy. For example in cases which a preposition occurs in an NP not a PP we may have some ambiguities. As an example in sentence: '*nameh-ye Ali beh Hassan resid*.' (Ali's letter to Hassan arrived) or (Ali's letter arrived to Hassan) we have two interpretations. For such cases we have developed some disambiguation modules which find the correct chunking regarding statistical and the semantic information about the verb (coded in the verb lexicon) and the constituent. For instance if the verb was '*gom shod'* (was lost) instead of '*resid*' (arrived) in the above example which leads to sentence '*nameh-ye Ali beh Hassan gom shod*.' (Ali's letter to Hassan was lost), as the verb – was lost – accepts one argument and has no prepositional phrase in its argument structure, we could choose the first interpretation (considering the preposition inside a noun phrase) easily. However according to some theories, in these cases we can postpone the ambiguity resolution to the next steps and do not consider the embedded chunks, otherwise we have to use the statistical information about the probability of initiating an argument (a role) by a preposition (ex. using preposition '*beh*' (to) for denoting destination role) to disambiguate the chunking process too.

In our work we have used some rules to cover a part of embedded structures as well. A detailed description of the rules we used can be found in [15]. These rules are applied on the POS tagged corpus and determine IOB tag of each token of the corpus. The accuracy of the available rule based method is about 70%. So after rule base IOB tagging of the corpus, a linguistic expert is needed to check the assigned IOB tags and corrects them manually.

The next phase which applies a supervised learning method for chunking is developed to overcome the shortcomings of the rule based method and increase the performance of chunking. Table I shows a part of IOB tagged corpus.

TABLE I.
A SENTENCE IN THE IOB TAGGED CORPUS

ID	Token (Transliteration)	Translation of Token	POS Tag	IOB Tag
1	در ('Dar')	In	P	B
2	چند ('Chand')	Seve ral	N	I
3	دهه ('Daheh-ye')	Decade	N	I
4	گذشته ('Gozashte')	P ast	ADJ	I
5	کار ('Kar-e')	Work	N	B
6	چشمگیری ('Chashm-giri')	Important (Valuable)	ADJ	I
7	در ('Dar')	In	P	B
8	زمینه ('Zamineh-ye')	Field	N	I
9	تاسیس ('Tasis-e')	Establishing	N	I
10	کتابخانه های ('ketab-khaneh-haye')	Libraries	N	I
11	آموزشگاهی ('Amouzeshgahi')	Teaching Institute (School)	ADJ	I
12	انجام ('Anjam')	Have not	N	B
13	نشده ('Nashodeh')	Been	ADJ	I

14	است (‘Ast’)	Done	V	I
15	.	.	DELM	O

I. Multilayer perceptron neural network

Different machine learning approaches are applied to the chunking problem for various languages.

Such problems can be considered as a classification problem where, given a number of extracted features from a predefined linguistic context which are acquired during learning and that the task is to predict the class of a new case.

In the proposed method, a multilayer perceptron neural network is used. A multilayer Perceptron is a feed forward artificial neural network model which maps sets of input data onto a set of appropriate output. It uses three or more layers of neurons with nonlinear activation functions and can distinguish data that is not linearly separable.

Multilayer Perceptron using a back propagation algorithm are the standard algorithm for any supervised-learning. They are useful in research in terms of their ability to solve problems stochastically, which often allows one to get approximate solutions for extremely complex problems.

a) The Tag Set

There are about 1000 POS tags available in Bijankhan Persian corpus among which we selected 17 ones for our system. It is obvious that having a large number of tags requires more input tokens for training the system. Also there are some extra tags in the corpus that using them does not make any difference in our IOB tagging. These tags consist of those which determine the person, tense or aspect of the verbs. For instance words “آمد ” (he/she came) and “آمدم” (I came) have different POS tags but both are IOB tagged as ‘I’ (means inside a phrase) in our chunker. Also words “آمد” (came) and “می آمد” (was coming) that are different in aspect and so have different POS tags, both have the same IOB tag: ‘I”.On the other hand,it should be considered that some POS tags are essential for chunking and omitting them from the tagset causes incorrect IOB tagging. As it is obvious, choosing a suitable number of POS tags to train the Neural Net with, is very important. We obtained this suitable number, for our available input tokens, with trial and error method.

We reduced the 1000 initial tags in an iterative process to its smallest size which (1) reserves the essential information for chunking and (2) reduces the size of training set as much as possible. After some experiments under supervision of a linguistic expert we found that we could reduce the POS tags to 17 and got the best results.

To create numeric input for the neural network, some numbers are assigned to POS tags which are called POS-Numbers. POS tags and their corresponding POS-Numbers are shown in Table II.

TABLE II I.
POS TAGS AND POS-NUMBERS

POS tag	Description	POS-Number
N	Noun	17
V	Verb	16
ADJ	Adjective	15
ADV	Adverb	14
SPEC		13
QUA	Quantifier	12
DET		11
P	Preposition	10
IF		9
PRO	Pronoun	8
RA		7
CON	Conjunction	6
AR	Arabic	5
DELM	Delimiter	4
Alpha-per		3
MS		2
SUBJ	Subject	1

b) Input and output data representation

The POS tag set consists of 17 basic tags each corresponding to a POS number.

The input of the neural net is a sequence of POS-Numbers which are divided with maximum POS-Number (17) as the neural network input should be within [0, 1] . A window of -2/+2 tokens centered at the focused token is used to make the appropriate input data for the system to work with.

The IOB tag of each token is determined according to the POS tag of the previous and next words and the word itself. So the input can be considered as (t(n-2) t(n-1) t(n) t(n+1) t(n+2)), in which t(i)is the POS tag of the ith token and the nth token is the focused token for which we are computing the IOB tag.

The output of the neural net determines one of the three classes I, O and B.

Table III shows the neural network input and output for the part of corpus shown in table I.

TABLE III..
NEURAL NETWORK INPUT AND OUTPUT

ID	Input	Output
1	(0,0,10,17,17) / 17	B
2	(0,10,17,17,15) / 17	I
3	(10,17,17,15,17) / 17	I
4	(17,17,15,17,15) / 17	I
5	(17,15,1715,10) / 17	B
6	(15,17,15,10,17) / 17	I
7	(17,15,1017,17) / 17	B
8	(17,10,17,17,17) / 17	I
9	(10,17,17,17,15) / 17	I
10	(17,17,17,15,17) / 17	I
11	(17,17,15,17,15) / 17	I
12	(17,15,17,15,16) / 17	B
13	(15,17,15,16,4) / 17	I
14	(17,15,16,4,0) / 17	I
15	(15,16,4,0,0) / 17	O

a) Training

A multilayer perceptron with two hidden layers is designed. These layers have 10 and 3 neurons respectively. As it was mentioned in the previous section, the output layer has 2 neurons. Activation functions of hidden layers are ‘tansig’ and ‘logsig’ respectively. The activation function of the output layer is ‘purelin’. The number of neurons in hidden layers and the type of activation functions are obtained by trial and error. The goal of the network is to obtain the minimum mean square error (MSE). 80 epochs are used to train the network.

In the proposed method, the Fuzzy C-Means (FCM) clustering is used because it is fast and precise enough. In FCM, each data may belong to two or more classes by a Membership value.

The fuzzy C-Means clustering is used to classify output data into three classes. Class I, O and B are determined according to the optimal centers of the clusters which calculated using FCM. Each data belongs to the class with the highest membership value. It is shown in experimental results that using FCM has more accuracy than a fixed threshold.

It should be mentioned that theSVM method was also tested for training the system but as we will explain, it did not result in as good as the MLP method. In our opinion there are some reasons that in this project, MLP results are better than SVM. The most important reason is the large size of the training set. As MATLAB's toolbox SVM is not optimum and takes a lot of memory, with this large amount of input data a normal PC with 2G RAM gives an "out of memory" error and needs more memory. We also tried the SVM Light module which is more optimal than MATLAB's, but with our possible resources, it also could not get more than 6000 tokens for the training part. The maximum amount of input data that SVM results with is about 6000 tokens. With the same number of input tokens MLP and SVM lead to very similar results. The comparison of SVM and MLP methods' results, for different number of input tokens for the training part, is given in Table IV. As in all rows of the table and for different amounts of input data, MLP results better, we believe that with more training data, it also results better, therefore we chose MLP (with the help of FCM) to be our base method of training. It should be mentioned that all the results shown in tables IV and V, are obtained by comparing the IOB tagged file created by our chunker with a golden standard created by an expert. So we can say that the system evaluation is done manually by this comparison.

TABLE IV
COMPARISON OF MLP AND SVM RESULTS FOR DIFFERENT NUMBER OF INPUT TOKENS FOR TRAINING

Number of tokens	MLP result	SVM result
4000	68.7%	65.8%
5000	69%	67.2%
6000	73%	7 0%
6500	76%	-

V. THE IMPLEMENTED TOOL

The proposed chunking system was implemented using Visual Studio 2008 and the C# language.

The MLP neural network which is the main part of the system was developed by Matlab. We selected Matlab due to its ease of data manipulations, implementation of algorithms and the interoperability with other programming languages.

Figure 1 shows the main form of the chunker system. As it shows, the path of input text should be firstly selected. The input format is shown in Table I.

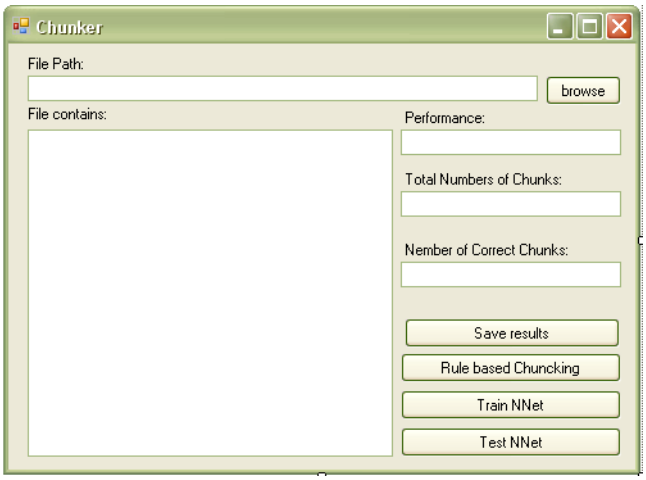

Fig. 1 The proposed chunker interface

The rule based chunker is applied on the selected input file by pressing the "Rule based Chunking" button. The results will be shown in the textbox labeled "File contains". Since there are some incorrect chunks and its performance is not 100%, an expert is needed in order to improve IOB tags. The results will be saved to use as neural network input.

The designed MLP neural network is trained afterward. The MLP neural network is developed using Matlab, and by pressing the "Train NNet" button, the constructed module will be called. Figure 2 shows its interface.

To test the system, the test file should be browsed and by clicking the "Test NNet" button, the output, the system performance, total number of chunks and number of corrected chunks will be shown.

VI. EXPERIMENTAL RESULTS

Here the proposed system results are firstly explained and then a comparison with a previous rule based method [2], [7] is done. The tagged corpus has 11600 tokens. A subset of 8000 tokens was used to train the MLP neural network and the remaining 3600 tokens were used to test.

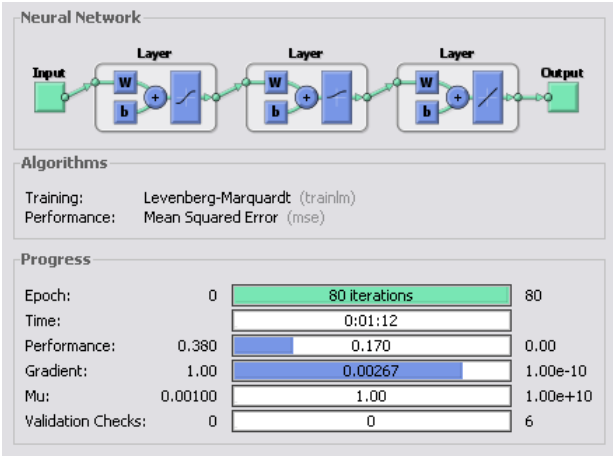

Fig. 2 MLP neural network design

The training algorithm is iterated over 80 epochs and the last Mean Square Error (MSE) was 0.17. The decreasing of MSE in training process can be seen in Figure 3. Also as said in section 4-3.2, the design of the network is a three layered with two hidden layers of 10 and 3 perceptrons which is shown in Figure 2.

The proposed system tagged 3085 tokens corpus correctly that concludes to the result of 85.7%. As it was mentioned before, the system evaluation is done by comparing the outputs with a golden standard created by human.

It should be considered that the best result of using a fixed threshold instead of Fuzzy C-Means (FCM) clustering was about 82%. This was predictable as the Fuzzy C-Means (FCM) clustering method results in a more flexible way of defining a threshold. Results of the system output for different fixed threshold values and FCM methods are shown in Table V.

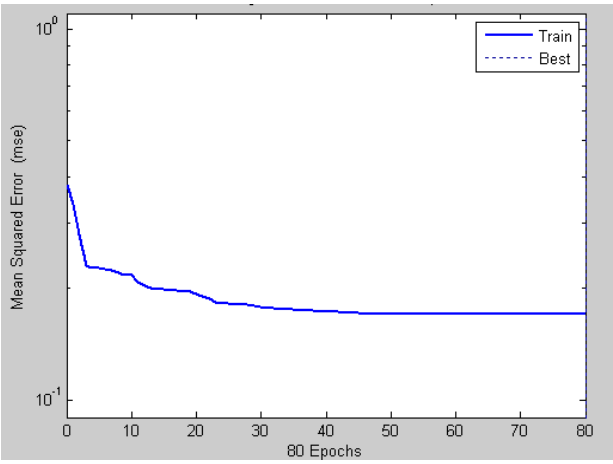

Fig. 3 Decreasing of MSE during the epochs

The only rule-based method available for Persian IOB tagging has the accuracy of about 70%, so our system gains a better performance and also it has the advantage of being automated.

TABLE V.
NEURAL NETWORK INPUT AND OUTPUT

Threshold value	Precision
0.4	80.8%
0.5	81.4%
0.6	83.1%
0.7	81.9%
FCM	85.7%

VII. CONCLUSION

Phrase chunking has an important role in natural language processing fields like machine translation, text to speech synthesis information retrieval, summarization, etc. Existence or non-existence of applicable corpora in natural language processing that have different tags or annotations, complexity and spread of semantic/syntactic rules have an important role in selecting an approach. If we do not have suitable corpus, we can not use learning or statistical approaches. Complexity or spread of semantic and syntactic rules, make it hard to use rule based approaches. A new automatic phrase chunking for Persian language has been presented in this paper. The corpus used to train and test the MLP is constructed by a rule

based approach for Persian IOB tagging. The proposed method uses a combination of a multilayer perceptron neural network and fuzzy C-Means clustering. Chunk information is represented using IOB tagging method. Fuzzy C-Means Clustering is applied on neural network outputs instead of fixed threshold to get more accuracy.

REFERENCES

[1] Milidiu, R. L., Santos, C., and Duarte, J. C., *"Phrase chunking using entropy guided transformation",* in Proc. of ACL-08: HLT, pages 647-655, USA, June 2008.

[2] Shamsfard M., Sadr Mousavi M., "Thematic role extraction using shallow parsing", *in International Journal of Computational, Intelligence* , 2008.

[3] Diab, M., K. Hacioglu and D. Jurafsky, *"Automatic tagging of Arabic text: from raw text to base phrase chunks", in Proc. of HLT-NAACL-04* , 2004.

[4] Yousif, J., Tangku, and Sembok, T., "Design and implement an automatic neural tagger based Arabic language for NLP applications", *in Asian Journal of Information Technology* , 2006.

[5] Tili-Guiassa, Y., "Hybrid method for tagging Arabic text", *in Journal of Computer science* , 2006.

[6] Tili-Guiassa, Y., "Tagging by combining rules-Based method and memory-based learning", *in Proc. of World academy of Science, Engineering and Technology* , 2005.

[7] Shamsfard M., Sadr Mousavi M., "A rule-based semantic role labeling approach for Persian sentences", *in Proc. of 2nd Computational Approach to Arabic Script Language* ,USA,2007.

[8] Thanh V. Nguyen , Hoang K. Tran , Thanh T. T. Nguyen, Hung Nguyen , "Word segmentation for Vietnamese text categorization : an online corpus approach" *,in the 4rd IEEE International Conference in Computer Science* , Hochiminh, Vietnam, 2/2006.

[9] Virongrong Tesprasit, Parison Charenpornsawat and Virach Sornlertlamvanich,"Learning phrase break detection in Thai text-to-speech", *in Proceeding of 8th European Conference on Speech Communication and Technology(EuroSpeech)* , Geneva Switzerland, 2003.

[10] Megerdoomian k., Zajac R., Processing *"Persian Text: tokenization in the Shiraz project",* technical report, NMSU,CRL , Memoranda in Computer and Cognetive Sience(MCCS-00-322),2000.

[11] E.karn Helen, "Design and Evaluation of a phonological phrase parser for Spanish text-to-speech", *in proceeding of 4th international Conference on Spoken Language* , vol.3 pages 1696-1699, Philadelphia, USA, Oct. 1996.

[12] Rezaei. S., "Tokenizing an Arabic script language", *NLP workshop at ACL/EACL* , France, 2001.

[13] Wang. Xin-Jing, Liu Wen,Qin Yong, "A search-based Chinese word segmentation method", *16th international Conference on World Wide Web* , 2007.

[14] Ruy Luiz Milidiú, Cícero Nogueira dos Santos, Julio Cesar Duarte, "Portuguese corpus-based learning using ETL", *in Journal of the Brazilian Computer Society, J.Braz.Comp.Soc* . vol.14 no.4 Campinas Dec. 2008.

[15] Sadrmousavi, Maryam, Shamsfard, Mehrnoush, 2006, *"Identifying Persian constituents by shallow parsing",* Technical report, NLP lab, Electrical & Computer Engineering Dept, Shahid Beheshti University, Tehran, Iran.