Journal of Computing and Security

October 2016, Volume 3, Number 4 (pp. 233–241)
http://www.jcomsec.org

Persian Texts Part of Speech Tagging Using Artificial Neural Networks

Zahra Hosseini Pozveh [a], Amirhassan Monadjemi [b,*] Ali Ahmadi [c,d]

[a] *Department of Computer, Science and Research Branch, Islamic Azad University, Tehran, Iran.*
[b] *Computer Engineering Department, University of Isfahan, Isfahan, Iran.*
[c] *Faculty of Computer Engineering, K.N.Toosi University of Technology, Tehran, Iran.*
[d] *School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: 28 July 2015
Revised: 14 October 2016
Accepted: 16 November 2017
Published Online: 01 April 2018

Keywords:
POS Tagging, Neural Networks, Persian.

ABSTRACT

Part of speech tagging (POS) is a basic task in natural language processing applications such as morphological parsing, information retrieval, machine translation and question answering. POS Tagging is the task of giving a word its part of speech (*e.g.* noun or verb). It is followed by a lot of challenging steps, in particular, disambiguation, named entity recognition and compound verb detection. Most of tagging approaches for Persian language are focused on the hidden Markov models (HMMs) and rule based models. Since Persian is a free word order language, those models cannot cope with all the complexity of this language for POS tagging, named entity, word sense disambiguation and other related tasks. In this paper, artificial neural networks (ANNs) are used for POS tagging due to their ability to learn complex patterns. In the first study ANN is fed with raw data and in the second phase, data are clustered and multiple ANNs are trained separately for each cluster. The accuracy rates of 95.7% and 96.17% were received respectively. Comparing the results with the other approaches makes it clear that neural networks can do POS tagging and named entity recognition more precise than other methods.

1 Introduction

Part of speech tagging is a fundamental step in most of natural language processing (NLP) applications, like machine translation (MT), question answering (QA) and text summarization(TS) [1, 2]. POS tagging is the process of assigning a word its part of speech, such as adjective or verb [3]. The task faces a lot of challenges including, ambiguity of words, named entity recognition (NER) and compound verb detection [4, 5].

Persian language is a free word order language. This means that its base structure frequently changes and words can place in different positions in a sentence without losing the veracity. This freeness makes POS tagging harder than other languages. On the other hand, there is a concept of 'Ezafe Kasreh' or simply Ezafe, which connects two words, mostly nouns. Ezafe is an unstressed vowel that does not have any writing symbol and it is pronounced 'YE or E'. Sometimes 'He' or 'Ye' hyponyms, are used to identify the case, but it is grammatically incorrect. Ezafe plays the role of " 's " or " of" in English (see Figure 1). Parsing this vowel in a sentence is essential to NER in Persian. Figure 1 clarifies the case.

* Corresponding author.
Email addresses: z.hossaini@gmail.com (Z. H. Pozveh), monadjemi@eng.ui.ac.ir (A. Monadjemi), ahmadi@eetd.kntu.ac.ir (A. Ahmadi)
ISSN: 2322-4460

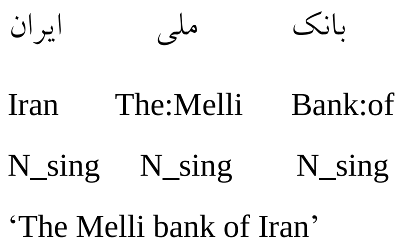

Figure 1. The Melli Bank of Iran '/banke Mellie Iran/ ' Is a Named Entity Which Is Built By Ezafat. N-Sing Stands for Noun.

NER is a subtask of information extraction. It tries to find predefined elements such as names of persons, organizations, locations, expressions of times and so on, in a text [6]. Sometimes named entities have more than one part, for example, 'AT and T Co'. Most of the time a named entity contains a proper noun. Proper nouns do not start with capitals in Persian, therefore, they are hard to detect. Also, Ezafe makes complex named entities in Persian language and since Ezafe does not show up in a text, named entities are not easy to find (see Figure 1). Considering all the above problems, a learning algorithm can facilitate the annotation process of named entities.

In Figure 1 /banke mellie Iran/ will be considered as N N N without Ezafe concept, but since this lexeme is a named entity it should be annotated as a single noun. Sometimes an Ezafe happens after another Ezafe, which is called /tataaboe Ezafat/ or 'multi Ezafe'. Locating those kinds of relations cannot be modeled effectively employing conventional POS tagging algorithms, but recurrent neural networks are suitable for named entity recognition and other subtasks.

A multiword expression (MWE) may also contain Ezafe. A MWE is a lexeme that has more than 1 word or lexeme (2 or more) [7, 8]. MWEs may have different properties from their components (*e.g.* "all at once" is a MWE which means suddenly. A named entity can be regarded as a special case of MWEs (noun compounds). Both multi-word expressions and named entities need special treatments since they affect part of speech tagging.

Compound verb detection is the next problem of POS tagging. A compound verb contains one verb which carries tense, aspect and mode. It is often accompanied by other verbs, nouns, prefixes and so on (depending on language rules). Persian language has a lot of compound verbs that should be addressed properly, but they are not the main concerns of this research.

Ambiguity is another main concerns of POS tagging. Ambiguity refers the fact that a word has more than one grammatical role or interpretation (depending on the task). For example,"interest" can be a noun or a verb. Persian language also faces the ambiguity problem. For instance, the word /Shirin/ can be a proper noun (name) as well as an adjective which means "Sweet".

There are two main methods of POS Tagging, Rule Based [3], and Stochastic [9] tagging which mostly employs hidden Markov model (HMM). A third model which is called transformation based tagging [10] or Brill model was introduced too.

During the past decade, intelligent tools like Artificial Neural Networks (ANN) [5, 11, 12], Support Vector Machine (SVM) [13], Evolutionary algorithms [13, 14], and Conditional Random Fields(CRF) [15] are used to Tag words in English and some other languages and those approaches mostly due to their learning ability have achieved remarkable results in tagging. POS tagging for Persian language has been studied since 2000[4], however most of the few approaches which are studied are rule-based or stochastic-based. In contrary the applications of intelligent models are too limited. In this paper we employ the artificial Neural Networks abilities for robust Persian language POS tagging.

This paper is organized as below, after the introduction and literature, Sections 2 and 3 describe the related works in Persian and methodology respectively. Section 4 discusses the results and Section 5 concludes.

2 Related Works

Persian is a free word order language and is a hard language to process. Due to rapid changes on different sciences, it receives a lot of new words from other languages. Those words mostly fall in the ambiguous class and harden NLP applications. As it mentioned earlier, the first study to annotate Persian documents was rule based. It was performed by Assi and Abdolhosseinin on FLDB corpus [4]. They employed Persian grammar to find tags of unknown words. This model however, is not flexible enough to support all the language changes. Bijankhan introduced a comprehensive corpus, which is named after its creator, the Bijankhan corpus [9]. A corpus is a huge set of documents, papers, newspapers and other texts that are pretagged [10]. A statistical method was firstly used by Raja [16] for Persian POS tagging. Three types of information were gathered by him, namely, structural, known words and unknown words. Arab and Azimazadeh [17], used HMMs to predict the tags of unknown words. 3-grams were used in their model and they tried to solve the ambiguity problem. Okhovvat and Minaiee [18] applied HMMs for POS tagging.

They trained a model by both homogenous and heterogeneous corpora. They determined the sentence boundaries to make the model more precise [18]. The next research was benefited from semantic analysis of sentences in order to find the missed tags and reduce the structural dependency [19]. Seraji also employed statistical methods for POS tagging and obtained high accuracy rate [20]. Shamsfard and Forsati [11] examined Bees colony algorithm to find the most probable tag for a word, the method employed stochastic information as its fitness function. Another study was designing a dependency parser for Persian language and discovering the linguistic dependencies to ease NLP tasks [21]. Kardan and Imani [22] used maximum entropy as a classifier for POS tagging. They chose those types of features that can show the most important characteristics of a word [22]. The next successful research that used dependency grammar, was on Ezafe detection and improving its precision rate [23]. Pakzad and Minaee [24] also used dependency grammar and joint probability for Persian and English annotation. In this paper we employ the artificial neural networks abilities for robust Persian language POS tagging and NER. In this approach; firstly, raw tag sequences are fed to neural network. Then, the documents are clustered and distinct ANNs are trained for each category. Clusters are used to reduce the error rate. Finally, the trained ANNs were used to tag the test set and the results were evaluated. Neural networks as a soft computing method have been chosen due to their learning ability to annotate words in other languages. Section three will discuss the methodology and in section four experimental results will be shown.

3 Methodology

3.1 Corpus

Bijankhan introduced a corpus of more than two million and half of tagged words, in 2004 [9]. Those words are from papers, daily newspapers and web sites which belong to 4300 news groups. The corpus has got 40 main tags of Persian language. BijanKhan has more than 880 POS tags, but we use only 50 most frequent of them.

3.2 The Proposed Method

In order to tag unknown or ambiguous words, Elman neural network is used. Then documents are clustered based on their subjects by a Fuzzy C-means algorithm. Next, we annotate a document using statistics of tag sequences in its cluster. This will reduce ANN parameters and learning time. Clustering also is a proper way to employ semantic information to improve the disambiguation process.

L_{i-m}	…	L_{i-1}	R_{i+1}	…	R_{i+m}	Prob1	Prob2

Figure 2. Input Vector for Neural Network, M Varies From 1 to 6, Prob 1,2 Are Probability of Ezafe Occurrence in the Adjacent Words.

3.2.1 Elman ANN

Having a word to annotate, the input vector contains L left and R right tags, if any available. The algorithm works as following:

(1) Make a vector of L left tags and R right tags, plus probability of Ezafe happening in the direct surrounding words. (For words at the beginning or at the end of each sentence, use only subsequent or preceding tags.)
(2) Initialize weights randomly (wij).
(3) Train the neural network using (1) and (2)
(4) The output will be the tag number and indicators of Ezafe happening.

Sentences in Persian end with '.', similar to most of other languages. Input vector for the ith word is in form of Figure 2. m is the neighborhood range that varies from 1 to 6 and can be more, but it only increases the network complexity without significant increase in the accuracy rate. In fact, the tests have shown that 3 and 4 are almost the optimum values for m. Numerical values are given to tags to make them more suitable for ANN. For example, 15, 2, 9, 3, representing the sequence 'DET N-PL ADJ V-PL' for the input sentence /een ketabha quob boudand/ or 'the books were fine'.

The number of output neurons is 50 which is the number of tags in the selected corpus. For an arbitrary unit in a recurrent Elman network, its activation at the time t is [25]:

$$y_i(t) = f_i(net_i(t-1))$$
$$y_i(t+1) = y_i(t) + logsig(\sum_{j=1}^{n}(w_{ij} - \delta_j)) + Input(t) \tag{1}$$

Elman receives feedback from hidden layers, and can learn sequential and time varying patterns (see Figure 3). Once feedback connections are allowed, the network topology becomes very free to learn the varying rules of free order languages.

While computing activations, it is required to know the activations of all units in the posterior set of nodes (P_i). For computing the final errors (δ_j), errors of all units in their anterior set of nodes (A_i) should be available [25].

Figure 3 shows an Elman neural network [25], where w_{ij} represents the learning weight of the *ith* neuron

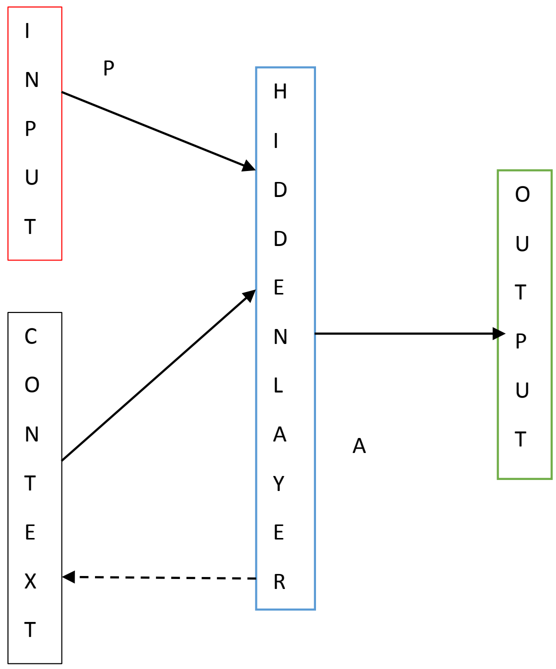

Figure 3. Elman ANN

and y(t) is the output of ANN at time t.

Hidden and output layers both employ the logsig activation function as it is shown in Equation(2).

$$Logsig(n) = \frac{1}{1 + exp(-n)} \tag{2}$$

Considering the Elman properties, it is suitable for learning complex pattern of Persian sentences and POS tagging.

Following the POS tagging stage, in order to detect Ezafe and Ezafat the ANN should be trained with proper inputs. Some of the main forms of Ezafe in a sentence are shown in Table 1. Passing POS tagging and Ezafe detection phases, it is time to perform more advanced processes, NER for instance.

NER is highly related with Ezafe detection. Since Persian language contains a lot of named entities, it is very important to detect them correctly. Named entities (NE) may have three parts, beginning-of-NE, middle-of-NE, end-of-NE. Most of the time, middle-of-NE is a person, place, location or time, but they can perch in other places as well. Bijankhan has a tag for proper nouns which is perfect for training phase. For the test step, a dictionary is needed to define persons, places, locations and so on. Thanks to FarsNet it is an easy step [26]. Then Elman learning rule is performed.

3.2.2 Clustering Approach

In order to cluster the documents, a Fuzzy C-Means algorithm is used. Firstly, unrelated words such as propositions (*e.g.*\aaz \ which means 'of '), conjunctions (*e.g.* \ke \which means 'that ') and pronouns (e.g \maan \that means 'I ') are eliminated from the incoming text, using a data dictionary. Then the algorithm computes word frequencies (WF) and normalizes them (nWF), dividing by the length of the text, as indicated in Equation (3).

$$nWF = WF/length(text) \tag{3}$$

Using a data dictionary for common names, Inverse Document Frequency (IDF) is computed. IDF is a cross-document normalization that puts less weight on the common terms and more weight on rare terms.

Then the fuzzy C-means algorithm is used to cluster the documents by minimizing Equation (4) [27].

$$arg_c min \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij}^m |x_i - c_j|^2 \tag{4}$$

where n, is the number of incoming documents; k, is the number of final clusters and m, is a value greater than 1 that determines the degree of cluster fuzziness. Larger m values result in smaller membership values, u_{ij}, and hence, fuzzier clusters.

X is a collection of n input documents, $\{x_1, x_2, \ldots, x_n\}$

C is the resulted cluster centers, $\{c_1, c_2, \ldots, c_k\}$

x_i is the ith input document.

c_j is the jth cluster center.

u_{ij} determines the membership degree of the document i in the cluster j, as it is stated in Equation (5).

$$u_{ij} = \frac{1}{\sum_{j=1}^{k} \frac{|x_i - c_j|}{|x_i - c_k|}^{\frac{2}{m-1}}} \tag{5}$$

where u_{ij} is between 0 to 1.

Cluster centers update offline. After clustering, for each category, a distinct Elman neural networks is trained separately. It reduces the training time and increases the tagging accuracy rate for the intra-clusters vocabulary. When there is an untagged word in a document, the neural network of the document's cluster, fires. Actually the input vector is (tags, markers, genre) in which, genre is a document's cluster.

4 Results

To evaluate the ANN based tagger, the *Peykare* corpus is used, which contains about 10 million tagged words. The results are compared to those of various the state

Table 1. Some Base Structures of a Sentence That May Contain Ezafe

Pattern	Example	Translation to English
N-N-V	در اتاق باز شد	The door of the room is openned.
N-N-ADJ-V	شهر اصفهان زیباست	The Isfahan city is beautiful.
N-N-ADJ-ADJ-V	شهر اصفهان بسیار زیباست	The Isfahan city is very beautiful.
N-ADJ-V	چهره زیبای مادر	The beautiful face of mother.

Table 2. Comparison of Four Different POS Taggers for Persian, on *Ezafe* Detection, Named Entity Recognition and Total Precision and Recall(R).

Task	NN-tagger	(R)	HMM-tagger	(R)	RBT	(R)	Pose	(R)
Ezafe Detection	90.1	90.1	84.3	84.12	79.5	79.23	92.12	92.18
NER	91.4	91.33	88.2	87.62	83.7	80.97	89.53	89.53
Unknown Word Det.	85.6	85.51	63.7	63.59	70.5	70.44	80.43	80.37
Known Word Det.	96.8	96.8	98.5	98.5	96.3	96.25	97.51	97.50
Total Precision	94.7		93.9		91.5		94.17	

of art algorithms in persian.

Precision and recall are used to evaluate the accuracy rate [12].

If an input document has T words and accordingly T tags, after annotation process the POS tagger produces M answers (the rest are considered as unknown). If C out of M tags are correct then:

$$Precision = C/M \tag{6}$$

$$Precision = C/T \tag{7}$$

4.1 POS Tagging Using Elman

The results are compared to various algorithm in Persian. The following algorithms are used for evaluation: HMM-tagger [18], rule based tagger [4] (RBT) and a combination of CRF and Ezafe analyzing which is called POSe by its designers [23]. Table 2 shows those comparisons. 70 percent of data were used for training, 20 percent for test and 10 percent for evaluation, Matlab is used for simulations.

As it is clear from Table 2, the ANN tagger (Elman) acts better than all taggers in named entity recognition and unknown words tagging. Unknown words annotation is one of the biggest challenges of POS tagging in all languages and its precision rate is almost low in Persian. Here, by an unknown word we mean, a new coming word in the language or a word with low frequency in the corpus.

POSe acts better in Ezafe detection, since it uses a gold Ezafe tag-set to improve its results [23]. The neural network reaches almost the same precision rate without any pre or post processes and needs no additional information. In overall, Elman ANN has better outcomes than other methods. Adding semantic information and additional processes may enhance the outcomes of Elman neural network.

Figures 4 and 5 show the error rate and the regression of ANN tagger for known words detection, where the regression is very close to 1. It is confirming that the ability of ANN in Persian POS tagging is high. The mean square error (MSE) of our recurrent neural network is between 0.01 and 0.1, for known words and between 0.2 to 1.1 for other subtasks. Those indicate that Elman ANNs are powerful enough to be employed in different NLP applications. More details are provided in Table 2. The Elman network with 12 to 20 neurons in its hidden layer has the best results. The input neurons vary from 3 to 6. With 4 neurons the shortest training time and also higher precision rate is received. More than 7200 samples of named entities were extracted from Bijankhan to train corresponding ANN. Input layer has maximum of 6 neurons, hidden layer contains 12 neurons. The output neurons may set to 1 or 0, consequent 1s are considered as a named entity.

4.2 Domain Dependent POS Tagging Using Elman

After data clustering, 97 to 124 clusters were found (based on the looser or tighter, outer or inner distances of clusters). 10 categories were selected to test the

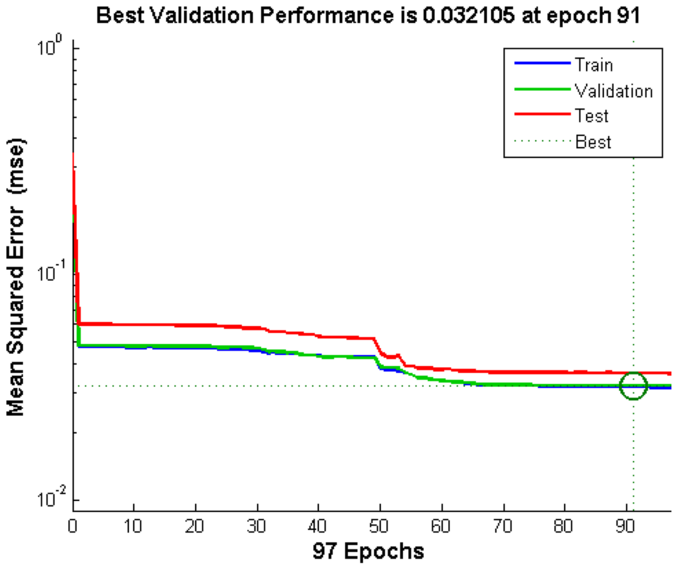

Figure 4. MSE for Training, Test and Validation Data for Known Words

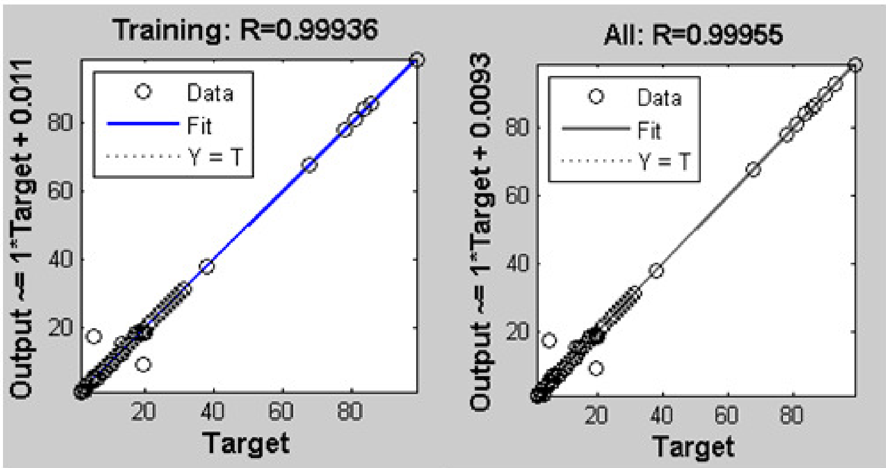

Figure 5. Regression of Training Data and Overall Regression on Test, Training and Validation. Regression is close to 1 and shows the ANN efficiency in POS tagging

results, which contains more than 300,000 words.

When a document comes for tagging, it passes the clustering phase. When a sentence with an unknown word or an ambiguous one is found, a 3-gram or a 4-gram (generally an N-gram) is made and it is sent to the related ANN to find its proper tag. Table 3 and Figure 5 show the results. As it is clear from Figure 6, the training iterations reduce significantly while the error rate is still low (for example, comparing Figures 4 and 6, 97 epochs without clustering reaches 55 epochs with clustering).

After clustering, the ANN- tagger shows better performance for unknown and ambiguous words, and almost better than POSe for known data. Clustering does not affect NER considerably.

Table 3. POS Tagging Using Data Cluster for 3 and 4 Grams

Task	NN-tagger (3-grams)	NN-tagger (4-grams)
UNKNOWN WORD DET.	89.9	87.1
KNOWN WORD DET.	97.2	97.8
AVERAGE	96.27	96.15

5 Conclusion

POS tagging is a challenging task in Persian language, since it has complex rules, which are hard to learn. ANN is used for POS tagging in Persian language in this paper. The results show that recurrent ANNs are efficient tools for POS tagging. Their feedbacks create a short-term memory, which can keep a sequence of

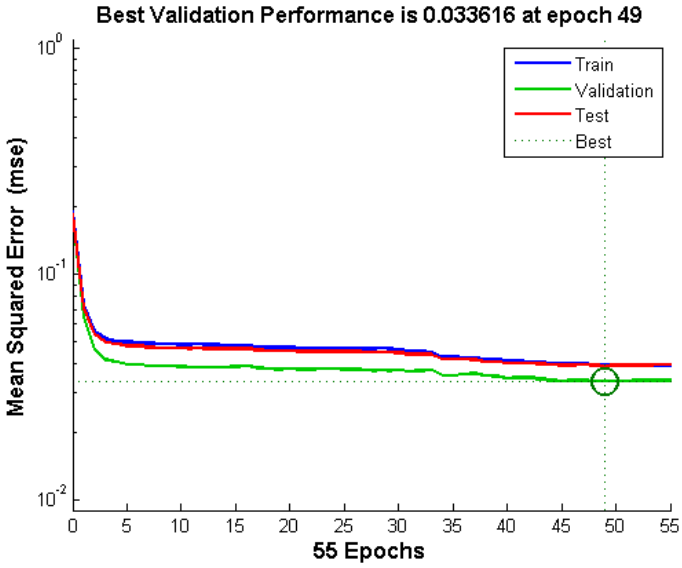

Figure 6. POS Tagging, Train, Test and Validation Error After Clustering for Known Words.

tags and help the network to recognize an unknown word's POS tag in fewer iterations. The results also demonstrate that ANNs are more precise than stochastic and other optimization models. ANN almost acts the same as other algorithms for known words annotation. The precision rate of NER increases significantly using the Elman neural network (more than 4%). Clustering also improves POS tagging more than 5%, since some semantic aspects are added to the model. Semantic analysis in other methods improve Ezafe detection process, but ANN achieves almost the same results in less time. In the future works combining ANN with semantic based models can improve the accuracy rate of all sub-tasks. It is needed for more advanced NLP tasks such as question answering, text summarization and machine translation. ANNs have suitable structure to be mapped on a parallel model, hence, a parallel model (Common Read Exclusive Write Parallel RAM or simply CREW PRAM), can be designed to improve speed up and time complexity.

References

[1] Jarosław Szkoła, Krzysztof Pancerz, and Jan Warchoł. Recurrent Neural Networks in Computer-based Clinical Decision Support for Laryngopathies: An Experimental Study. *Computational Intelligence and Neuroscience*, 2011:1–8, January 2011. ISSN 1687-5265. doi:10.1155/2011/289398.

[2] Jiang-Liang Hou and Yong-Jhih Chen. A conceptual model for text summarisation based on reader requirements. *Journal of Experimental & Theoretical Artificial Intelligence*, 27(3):317–323, sep 2014. doi:10.1080/0952813x.2014.954275.

[3] Steven J. DeRose. Grammatical Category Disambiguation by Statistical Optimization. *Computational linguistics*, 14(1):31–39, January 1988. ISSN 0891-2017.

[4] S. Mostafa Assi. Grammatical Tagging of a Persian Corpus. *International Journal of Corpus Linguistics*, 5(1):69–81, 2000. doi:10.1075/ijcl.5.1.05ass.

[5] R. A. Dam and A. Guessoum. Building a neural network-based English-to-Arabic transfer module from an unrestricted domain. In *2010 International Conference on Machine and Web Intelligence*, pages 94–101, Oct 2010. doi:10.1109/ICMWI.2010.5648157.

[6] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54. Association for Computational Linguistics, 2003. doi:10.3115/1073445.1073462.

[7] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[8] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword Expressions: A Pain in the Neck for NLP. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-45715-2. doi:10.1007/3-540-45715-1_1.

[9] Farhad Oroumchian, Samira Tasharofi, Hadi Amiri, Hossein Hojjat, and Fahime Raja. Creating a feasible corpus for Persian POS tagging. Technical Report UOWD Technical Reports Series Number TR 3/2006, College of IT, University of Wollongong in Dubai, 2006.

[10] E. Brill and M. Pop. *Unsupervised Learning of Disambiguation Rules for Part-of-Speech Tagging*, pages 27–42. Springer Netherlands, 1999. ISBN 978-94-017-2390-9. doi:10.1007/978-94-017-2390-9_3.

[11] R. Forsati and M. Shamsfard. Cooperation of evolutionary and statistical PoS-tagging. In *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, pages 446–451, May 2012. doi:10.1109/AISP.2012.6313789.

[12] Atro Voutilainen. Part-of-speech tagging. *The Oxford handbook of computational linguistics*, pages 219–232, 2003.

[13] J. Pailai, R. Kongkachandra, T. Supnithi, and P. Boonkwan. A comparative study on different techniques for Thai part-of-speech tagging. In *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 1–5, May 2013. doi:10.1109/ECTICon.2013.6559527.

[14] Enrique Alba, Gabriel Luque, and Lourdes Araujo. Natural language tagging with genetic algorithms. *Information Processing Letters*, 100(5):173 – 182, 2006. ISSN 0020-0190. doi:https://doi.org/10.1016/j.ipl.2006.07.002.

[15] A. K. Barman, J. Sarmah, and S. K. Sarma. POS Tagging of Assamese Language and Performance Analysis of CRF++ and fnTBL Approaches. In *2013 UKSim 15th International Conference on Computer Modelling and Simulation*, pages 476–479, April 2013. doi:10.1109/UKSim.2013.91.

[16] Fahimeh Raja, Hadi Amiri, Samira Tasharofi, Mehdi Sarmadi, Hossein Hojjat, and Farhad Oroumchian. Evaluation of part of speech tagging on Persian text. In *Proceedings of the Second Workshop on Computational Approaches to Arabic Script-based Languages, Stanford, California*, pages 120–127, July 2007.

[17] Ali Azimizadeh, Mohammad Mehdi Arab, and Saeid Rahati Quchani. Persian part of speech tagger based on Hidden Markov Model. In *JADT 2008 : 9th international conference on textual data statistical analysis*, pages 121–128, March 2008.

[18] Morteza Okhovvat and Behrouz Minaei Bidgoli. A hidden Markov model for Persian part-of-speech tagging. *Procedia Computer Science*, 3:977 – 981, 2011. ISSN 1877-0509. doi:https://doi.org/10.1016/j.procs.2010.12.160. World Conference on Information Technology.

[19] Hakimeh Fadaei and Mehrnoush Shamsfard. Persian POS tagging using probabilistic morphological analysis. *International Journal of Computer Applications in Technology*, 38(4):264–273, 2010. doi:10.1504/IJCAT.2010.034527.

[20] Mojgan Seraji. A Statistical Part-of-Speech Tagger for Persian. In *Proceedings of the 18th Nordic Conference of Computational Linguistics NODALIDA 2011*, number 11 in NEALT Proceedings Series, pages 340–343, 2011.

[21] Mojgan Seraji, Beáta Megyesi, and Joakim Nivre. Dependency parsers for Persian. In *Proceedings of the 10th Workshop on Asian Language Resources*, pages 35–44, 2012.

[22] A. A. Kardan and M. B. Imani. Improving Persian POS tagging using the maximum entropy model. In *2014 Iranian Conference on Intelligent Systems (ICIS)*, pages 1–5, Feb 2014. doi:10.1109/IranianCIS.2014.6802567.

[23] Alireza Nourian, Mohammad Sadegh Rasooli, Mohsen Imany, and Heshaam Faili. On the importance of ezafe construction in Persian parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 877–882, 2015.

[24] A. Pakzad and B. Minaei Bidgoli. An improved joint model: POS tagging and dependency parsing. *Journal of AI and Data Mining*, 4(1):1–8, 2016. ISSN 2322-5211. doi:10.5829/idosi.JAIDM.2016.04.01.01.

[25] Simon Haykin. *Neural Networks: A Comprehensive Foundation (3rd Edition)*. Prentice-Hall, Inc., 2007. ISBN 0131471392.

[26] Mehrnoush Shamsfard, Akbar Hesabi, Hakimeh Fadaei, Niloofar Mansoory, Ali Famian, Somayeh Bagherbeigi, Elham Fekri, Maliheh Monshizadeh, and S Mostafa Assi. Semi automatic development of farsnet; the persian wordnet. In *Proceedings of 5th global WordNet conference, Mumbai, India*, volume 29, 2010.

[27] Sumit Goswami and Mayank Singh Shishodia. A Fuzzy Based Approach to Text Mining and Document Clustering. *International Journal of Data Mining & Knowledge Management Process*, 3(3): 43–52, may 2013. doi:10.5121/ijdkp.2013.3304.

Zahra Hosseini Pozveh has a B.Sc. in Computer Engineering (software) from Islamic Azad University of Najafabad and M.Sc. In Artificial intelligence form Islamic Azad University of Science and Research, Tehran. She is currently Ph.D. student in Artificial Intelligence (NLP). She is also a faculty member of Islamic Azad University of Isfahan (Khorsagan). She is interested in fields of NLP, big data mining, deep learning and parallel processing.

Amirhassan Monadjemi was born 1968, in Isfahan, Iran. He got his PhD in computer engineering, pattern recognition and image processing, from University of Bristol, Bristol, England, in 2004. He is now working as an associate professor at the Department of AI, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran. His research interests include AI, image processing, artificial neural networks, and physical detection and elimination of viruses.

Ali Ahmadi received his B.Sc. in Electrical Engineering from Amirkabir University, Tehran, Iran and M.Sc. and Ph.D. in Artificial Intelligence and Soft Computing from Osaka Prefecture University, Japan in 2001 and 2004, respectively. He worked as a researcher in Hiroshima University, Japan during 2004-2007. He is currently a faculty member at K.N. Toosi University of Technology. His research interests are interactive learning models, virtual reality and artificial life, semantic data mining and information fusion, image search engines, and Software-hardware system co-design.