Levenberg-Marquardt Learning Neural Network For Part-of-Speech Tagging of Arabic Sentences

HASAN MUAIDI
Computer Science Department
Prince Abdullah Bin Ghazi Faculty of Information Technology
Al-Balqa' Applied University
Salt
JORDAN
alserhan@bau.edu.jo

Abstract: Part-of-speech tagging is usually the first step in linguistic analysis. Also, it is a very important intermediate step to build many natural language processing applications. This paper examines the application of neural networks to the task of tagging Arabic sentences. The network is trained with the help of Levenberg-Marquardt learning algorithm. Corpora of 24,810 words are collected and manually tagged to train the neural networks and to test the performance of the developed POS-Tagger. The developed tagger achieved an accuracy of 98.83% when evaluated on the train set and 90.21% on the test set. The performance of the Levenberg-Marquardt learning algorithm was compared with the performance of the traditional Backpropagation learning algorithm. It was found that the Levenberg-Marquardt Learning neural network is an efficient approach and more effective than the traditional Backpropagation learning algorithm for tagging Arabic words.

Key–Words: Part of Speech Tagging, Arabic Language, Neural Networks, Levenberg-Marquardt Learning Algorithm, Backpropagation Learning Algorithm.

1 Introduction

Part-of-speech (POS-tagging) is considered as a process for automatically assigning the proper grammatical tag to each word of a written text according to its appearance on the text. Thus, the task of POS-tagging is attaching appropriate grammatical or morpho-syntactical category labels to every token, and even to punctuation marks, symbols, abbreviations, . . . etc. in a corpus [1].

POS-tagging is usually the first step in linguistic analysis. Also, it is a very important intermediate step to build many natural language processing applications. It could be used in machine translation, spell checking and correcting, speech recognition, information retrieval, information extraction, corpus analysis, syntactic parsing and text-to-speech synthesis systems [1].

This paper explores the use of artificial neural networks (ANNs) for the problem of partofspeech tagging for Arabic language. The using of ANNs is a new approach in Arabic natural language processing. It had been used and applied successfully in many applications such as extracting the roots and stems for Arabic words [2][3], speech recognition, and part-of-speech prediction [4].

Arabic is considered as a regular language with very few irregular forms [5]. It is a rich language full of vocabulary; and is characterized by its complex morphology and complicated structure of inflection, which in many cases changes the structure of the words, causes high degree of complexity of tagging [5]. The limitations of the current Arabic tagging systems and the modesty of the accuracies of the available systems have induced the author of this paper to investigate a novel approach to build a POS-tagger based on artificial neural networks for Arabic language.

The paper starts with a brief summary of Arabic language. In Section 3, notable previous works are presented. Section 4 describes the principles of artificial neural networks. The experimental results and the causes of the errors are discussed in

Section 5. Section 6 makes a comparison between the proposed tagger and other approach based on standard Backpropagation algorithm. Finally, Section 7 concludes the paper.

2 Arabic Language

Arabic is an eternal language and it is considered as one of the oldest languages in the world. It is ranked the fifth in the widely used these days. Arabic alphabet consists of 28 letters. Arabic words are written as a series of letters, in which the letters of a single word strung together to form it. Unlike English and the Indo-European languages, Arabic text is oriented right to left without the use of capital letters [6].

In addition, Arabic differs from other languages syntactically, morphologically, and semantically that make it one of the most difficult languages for written and spoken language processing [7]. Arabic has been increasingly used in many information retrieval systems and recently on the Internet.

A word in Arabic language is defined as a collection of letters strung together as a single unit has a specific meaning. Arabic grammarians categorized Arabic words into three main part-of-speech classes. These classes are: noun, verb and particle [7].

In Arabic language there are two genders: masculine and feminine. In western languages words are singular or plural, but in Arabic language the words could be singular, dual or plural. The dual represent a total of two of nouns, pronouns, verbs or adjectives. The plural in western languages is happen by adding the letter ”s” to the end of the word, whereas in Arabic language the plural are two types: regular and broken. The diacritics in Arabic language is a characteristic that does not exist in western languages, that makes it more complex for writing and understanding than the other languages, as diacritics make the nouns either nominative, accusative, or genitive [8].

3 Previous Work

In recent years, there has been an enormous body of work done to solve the problem of part-of-speech tagging. In literature, there are two main methodologies for automatic POS-tagging [1]: (a) rule-based methodology; (b) stochastic (probabilistic) methodology. Most of POS-tagging systems have been implemented using these two methodologies. Some of the existing systems combined the two methodologies to produce a hybrid one which uses the both methodologies, and some other systems use other approaches.

Altunyurt and Orhan [9] summarized the commonly approved methods for the POS-taggers. These methods are demonstrated in Figure 1.

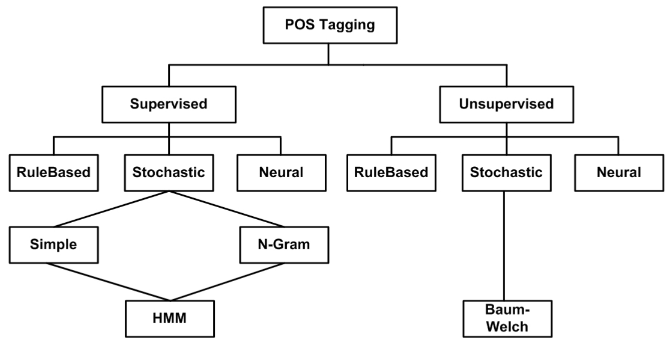

Figure 1: The Common Methods for the POS-Taggers [9]

3.1 Rule-Based Approach

The rule-based approach is the earliest approach was used for automated POS-tagging [10]. The starting using of the rule-based approach goes back to the 1960’s and 1970’s [7].

The rule-based approach tries to use a set of linguistic rules during the tagging process [11][10] and it is based on a core of solid linguistic knowledge [12]. The number of rules that is used in the tagging process could differ from hundreds to thousands. It requires manual work of experts so a huge work and cost is required [10].

Because knowledge representation in the rule-based approach is in the form of rules, it dose not need a huge amount of stored information [10]. This approach is considered to be easy to maintain and provides an accurate systems [10].

Some of a well-known rule-based systems are:

1. POS tagger developed by Harris in 1962.
2. CGC (Computational Grammar Coder) system developed by Klein and Simmons in 1963.

3. TAGGIT system developed by Greene and Rubin in 1971.

4. TBL (Transformation-Based error-driven Learning) system developed by Brill in 1992.

5. Fidditch system developed by Donald Hindle in 1989.

6. ENGCG (English Constraint Grammar) system developed by Voutilainen in 1995.

3.2 Stochastic Approach

The stochastic approach is also known as statistical or probabilistic approach. It based on building a statistical language models (trainable models) and estimating parameters using previously tagged corpus [10]. The statistical language model is build by collecting statistics from existing corpora [10]. Some of these statistics parameters are listed below:

1. **Lexical Probability:** The probability that a certain word appears with a certain tag.

2. **Contextual Probability:** The probability that a tag followed by another.

Stochastic approach requires less work and cost than the rule-based approach. It is considered as the most popular approach of the POS-tagging [13]. It is also considered to be more transporting of the language model to other languages especially when a huge manually tagged corpus is available [10]. Probabilities can be calculated automatically from the corpus. The problem with statistical approach that the tag to the unknown words can not be found [10].

Hidden Marcov Models (HMMs) is an example of statistical approach. HMMs describe a stochastic tagging algorithm that is concerned with modeling a sequence of tags in a sentence [13]. It is called hidden, because the sequence of tags is hidden from the observer of the text [13].

In this approach, a sequence of words that forms a sentence are given and the task is to determine the set of tags these words belong to [13]. Discussing the HMM technique is beyond the scope of this thesis. For further details about HMMs, the reader can refer to related literature such as the book ”Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition” [1].

The following list shows some of the systems that are implemented based on the stochastic approach:

1. WISSYN system developed by Stolz and others in 1965.

2. POS-tagger developed by Bahl and Mercer in 1976.

3. CLAWS (Constituent-Likelihood Automatic Word-Tagging System) system developed by Marshal, Garside, Leesh and Atwell in the period from 1981 to 1983.

4. PARTS system developed by Church in 1988.

5. POS-tagger developed by Cutting in 1992.

6. POS-tagger developed by Kupiec in 1992.

7. POS-tagger developed by Weischedel in 1993.

8. POS-tagger developed by Merialdo in 1994.

3.3 Hybrid Approach

Hybrid taggers approach combines both rule-based and stochastic approach methods and achieved a higher rate of accuracy [14]. In 1994, Tapanainen and Voultilainen developed a tagger for French language that used both techniques separately and achieved an accuracy of 98% [14].

The following list shows some of the systems that are implemented based on the hybrid approach:

1. POS-tagger developed by Chanod and Tapanainen for French language.

2. POS-tagger developed by Kuba and others for Hungarian.

3. POS-tagger developed by Schneider and Volk.

3.4 Other Approaches

These approaches are inspired from the Artificial Intelligence field such as machine learning, memory based and neural networks [7]. The following list shows some of the systems that are implemented based on the neural networks approach:

1. POS-tagger developed by Schmid.

2. POS-tagger developed by Antonio and others.

3.5 Arabic Part-of-Speech Tagging

Different techniques of POS tagging models have been implemented and performed for English language. On the contrary, only a small amount of work has been done for Arabic language [15]. The structure of Arabic language is different than English, so it is not possible to apply available methods directly for Arabic. Few efforts have been done in Arabic part of speech tagging. The following is a list of some POS systems that are implemented for Arabic language:

1. El-Kareh and Al-Ansary implemented a statistical approach [16].
2. Shereen Khoja implemented a hybrid tagger system that uses both morphological rules and statistical techniques in the form of hidden Markov model [7].
3. Andrew Freedman implemented a tagger based on Brill's [17].
4. Mona Diab and others implemented a tagger based on support vector machine [18].
5. Habash and Rambow implemented a tagger based on support vector machine [19].
6. Alshamsi and Guessom implemented a tagger based on hidden Markov model [20].
7. Masri and others implemented a tagger based on memory based approach [21].
8. Zribi and others used a combined approach by combining rule based tagger with trigram hidden Markov model tagger [22].
9. Yousif and others implemented a tagger based on support vector machine [23].
10. El-Hadj implemented a tagger based on combining morphological analysis with statistical approach (hidden Markov model) [24].
11. AlGahtani and others implemented a tagger based on TBL [25].
12. Alqrainy implemented a tagger based on a rule-based approach [10].
13. Ben Ali and Jarray implemented a tagger based on genetic algorithm [26].
14. Abu-Malloh implemented a tagger based on standard Backpropagation neural network [27].

4 Neural Network

Artificial Neural Networks (ANNs) model is an information processing paradigm that is inspired by the way biological nervous systems. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problem. ANNs like people learn by experience not from programming, they are trained (learned) by repeatedly presenting examples (data) to the network which has a training rule and a weighted connection neurons that are adjusted on the basis of data that cannot be altered after the training, these weights help the network to make the decision without the need to use any other resource in decision-making [28][29].

ANNs are fast, tolerant of imperfect data, and do not need formulas or rules. For these reasons, ANNs have been applied to an increasing number of real-world problems of considerable complexity [29] in fact that ANNs are capable of solving complex real-life problems by processing information in their basic neurons in a non-linear, distributed, parallel and local way.

4.1 Backpropagation Neural Network

Backpropagation Neural Network (BPNN) is one of the most common neural network architectures, which has been used in a wide range of machine learning applications [28].

The BPNN structure is illustrated in Figure 2. Typically, BPNN architecture consists of three or more fully interconnected layers of neurons [28]: input, one or more hidden, and output layers. Every layer in the network have a fixed number of nodes (neurons). The input layer receives input data from an external source, the output layer transmit the result of the neural network processing, and the hidden layer provides the internal relations between input and output layers.

Training inputs are applied to the input layer of the network, and desired outputs are compared at the output layer. During the learning process, a forward sweep is made through the network, and the output of each element is computed layer by layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function, and the connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached.

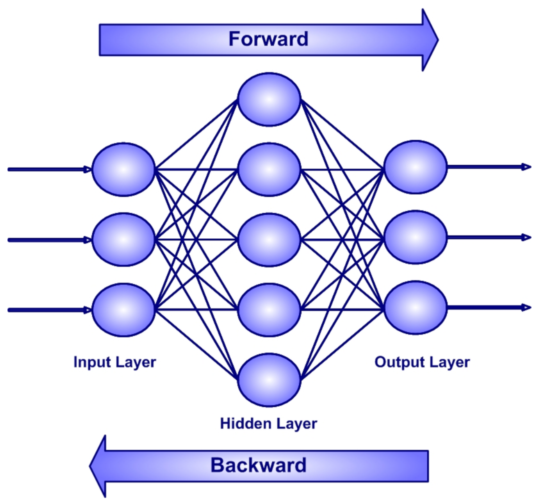

Figure 2: Backpropagation Neural Network

4.2 Levenberg_Marquardt Algorithm

The Levenberg_Marquardt (LM) algorithm is an approximation to the Newton method used for training ANNs. This optimization technique is more powerful than standard Backpropagation Neural Network (BPNN). LM algorithm is very efficient and fast having also a quite good global convergence property. For these reasons, LM algorithm is used in this study. Algorithm 4.1 shows the pseudocode of the LM algorithm [30].

Algorithm 4.1: LEVENBERG MARQUARDT(n)

Initialize Weights
while not StopCriterion

$$\textbf{do} \begin{cases} \text{calculate } e^p(w) \text{ for each pattern} \\ \text{calculate } e1 = \sum_{p=1}^{P} [e^p(w)]^2 \\ \text{calculate } J^p(w) \\ \textbf{repeat} \\ \quad \text{calculate } \Delta w \\ \quad \Delta w = -[\mu I + \sum_{p=1}^{P} J^p(w)^T J^p(w)]^{-1} \ \nabla E(w) \\ \quad e2 = \sum_{p=1}^{P} e^p(w+\Delta w)^T e^p(w+\Delta w) \\ \quad \textbf{if } e1 \le e2 \\ \qquad \textbf{then } \mu = \mu * \beta \\ \textbf{until } (e2 < e1) \\ \mu = \mu/\beta \\ w = w + \Delta w \end{cases}$$

The notations described below were used in algorithm 4.1 above:

$J^p(w)$	Jacobian matrix of derivatives of each error to each weight.
μ	A scaler.
$e^p(w)$	The vector error of pattern p.
I	The identity Matrix.
β	A factor.
Δw	Equation for update ANN weights.

A brief description and specific details on the Levenberg-Marquardt algorithm can be found in [31].

5 Experimental Setup

5.1 Data Used for Experiments

For Arabic language there is no available annotated corpus of sound quality for free, for this reason, a tagged Arabic corpus is needed. The huge spread of the search engines in Arabic language and the use of digital Arabic texts over the Internet in the last decade make it easier to collect digital Arabic texts from different sources to build an Arabic corpus. A shell program is implemented to extract 24,810 distinct words from different Arabic web-sites that are written in Modern Standard Arabic (MSA). The extracted words are not limited to a particular subject, they cover a wide range of subjects. The generated corpus are tagged manually and finalized with a help from an Arabic linguist specialist.

5.2 Designing a Tagset

A tagset is a set of terms (symbols) representing grammatical categories (case, gender, etc.) of word forms [5]. There is no standard tagset that is used by all researchers for all languages. Fortunately the Arabic tagset that has been compiled in this study is an adoption of the work proposed by Alqrainy as part of his PhD study at De Montfort University [32][10]. This tagset is called ARBTAGS, it contains 161 detailed tags and 28 general tags covering Arabic main POS classes and sub-classes. Figure 3 illustrates the main classification of ARBTAGS tagset.

5.3 Representation of the INPUT and OUTPUT data

Both the training and testing datasets consist of tokens. The token here is a series of Arabic letters. In

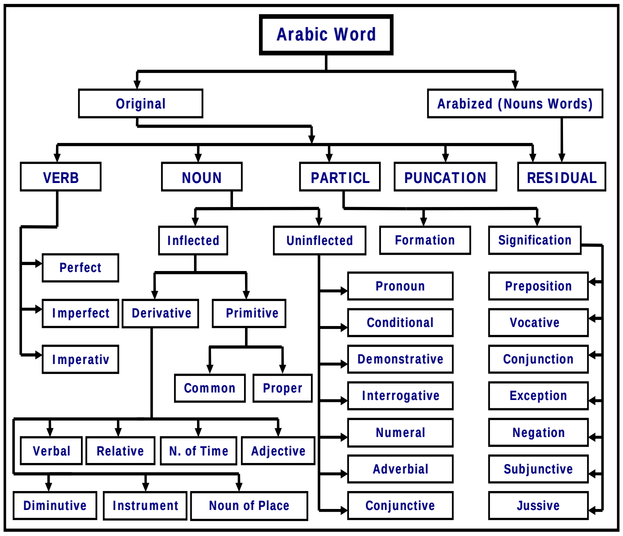

Figure 3: ARBTAGS Tagset Hierarchy

order to let the ANN understand and use these tokens, all letters should be transformed into numeric values. Muaidi [3] has coded the Arabic letters in an efficient method. This method is based on a scientific analysis for the frequency of Arabic affix letters. Table 1 shows Muaidi coding technique for the Arabic letters.

For the coding of the target vector, the binary coded method is used. The idea behind this method is for each class in the tagset there is a vector consists of binary numbers. The bit value **1** in this vector means that this tag is admissible for the given word. While, the bit value **0** in the target vector means that this tag is inadmissible for the given word.

5.4 Experimental Results

After the network architecture has been established and the number of neurons at each layer has been determined, it is necessary to determine which value must be assigned to the different weights in a way that minimize the error rate. To extract such weights, the Levenberg-Marquardt algorithm is used to train the developed network.

Reducing the error rate is achieved by one of the most common methods to determine the connection weights according to Lai and Serra [33]. In this method the input vector $\mathbf{X} = < X_1, X_2, \ldots, X_{12} >$ with its corresponding output $\mathbf{Y} = < Y_1, Y_2, \cdots >$

Table 1: Muaidi Coded Arabic Letters [3]

Code	Letter
20	[ا]
19	[ي]
18	[و]
17	[ت]
16	[ن]
15	[م]
14	[ل]
13	[ه]
12	[ة]
11	[إ]
10	[س]
9	[أ]
8	[ى]
7	[ئ]
6	[ء]
5	[ؤ]
4	[آ]
3	[ف]
2	[ب]
1	[ك]
0	[باقي الحروف]

are presented to the ANN which is considered as experimental data. In this study, the collection of characters that constitute the word represent the input to the network and the tag that is associated to the word represents the desired output.

Using the input vector **X**, the output **O** is calculated, this value differs than the desired output **Y** and it is called the actual output or the net output. The difference between the desired output and the actual output is computed and is called the error.

After that the error is computed using the mean squared error (MSE). The error is propagated backward to change the weights in order to reduce the error rate. This process is repeated for a series of experimental data until the error rate is acceptable.

As mentioned before, the compiled Arabic corpus consists of 24,810 Arabic words with their associated tags. These words are considered as a dataset to train and to test the developed ANN. This dataset is broken up into two distinct sets: training set and testing set.

The training dataset consists of 19,848 Arabic words. This is about 80% of the original dataset. While, the testing dataset consists of 4,962 Arabic words. These words form the remaining words in the original dataset with a ratio 20%. In order to eliminate bias, all the words in the training set and the testing set are chosen randomly.

The developed ANN is tagged successfully 98.83% of the words in the training dataset. This result indicates that the developed ANN is trained well. Table 2 summarizes the evaluation of the results in the training dataset. While, Figure 4 illustrates these results in a bar format.

Table 2: The Evaluation of the Results in Training Dataset

Number of Words	19,848
Words Tagged Correctly	19,615
Words Tagged Wrongly	233
Success Rate S_R	98.83%

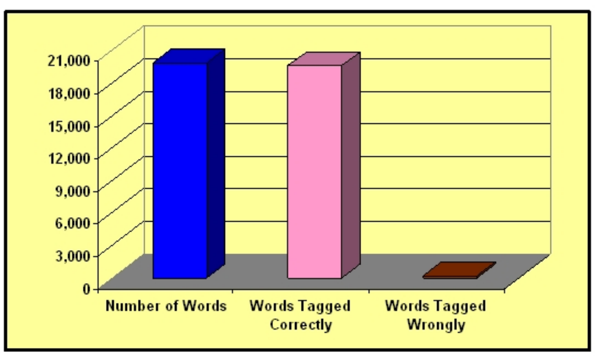

Figure 4: The Evaluation of the Training Dataset

5.5 The Testing Stage

Another experiment is performed to indicate the accuracy of the developed tagger system. This is done by using the testing dataset. As mentioned before, this set consists of 4,962 Arabic words. All of these words are unseen by the developed ANN. In order to check the performance of the developed ANN, the accuracy is calculated using the success rate measure S_R as expressed in Equation 1.

$$S_R = \frac{\text{No. of correctly tagged tokens}}{\text{No. of tested tokens}} \text{x} 100\% \quad (1)$$

The experiment is performed on the developed tagger system using the testing dataset and the success rate is reached to 90.21%. Table 3 summarizes the evaluation of the results in the testing dataset. While, Figure 5 illustrates these results in a bar format.

Table 3: The Evaluation of the Results in Testing Dataset

Number of Words	4,962
Words Tagged Correctly	4,476
Words Tagged Wrongly	486
Success Rate S_R	90.21%

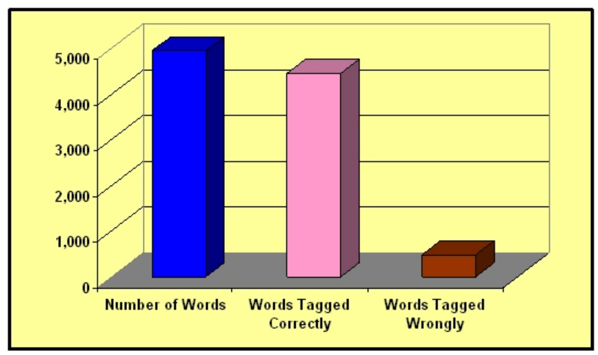

Figure 5: The Evaluation of the Testing Dataset

5.6 Causes of the Errors

The error rate differs with the number of the words in the testing corpus. Because in the training corpus patterns of the words were used and in Arabic language particles have no patterns, the exist of the particles in the testing corpus caused an error. Names of the people in the testing corpus that is derived from non-Arabic language also caused an error in the tagging. As a summary there are several sources of the errors in the proposed system some of these are:

1. The existence of the particles in the testing data.

2. The existence of the Arabic proper names in the testing data.

3. The existence of the words of foreign origins in the testing data.

6 Comparison with other Approach

As mentioned in Section 3.5, there are different approaches to the problem of part-of-speech tagging for Arabic language. In this section our proposed approach is compared with other POS taggers from state of the art namely Abu-Malloh Arabic tagger.

Abu-Malloh [27] has been developed an Arabic POS tagger based on a classical neural network. The network was trained by standard Backpropagation algorithm and implemented on the three-layer network with 8 hidden neurons. The size of the used tag set is 18 tags, where the size of the used corpus is 16,672 distinct words. This corpus was broken up into two distinct sets: training set and testing set. The training dataset consists of 13,337 Arabic words. This is about 80% of the original dataset, while the remaining 20% of the data is used for testing. The overall accuracy of Abu-Malloh developed system reached to 87.02% using the testing dataset.

The comparing process with other developed Arabic taggers is difficult task due to its accuracy relies on different parameters such as tag set size, training data size, testing data size and evaluation metrics used. The performance of our proposed tagger based on Levenberg-Marquardt algorithm and Abu-Malloh system based on standard Backpropagation algorithm are shown in Table 4 for comparison purpose.

Table 4: Comparison Results

	Our Tagger	**Abu-Malloh Tagger[27]**
Learning Alg.	LM	BPNN
Corpus Size	16,672	24,810
Size of Tag Set	18	189
Training Set Size	13,337	19,848
Testing Set Size	3,335	4,962
Success Rate	87.02%	90.21%

The overall results in Table 4 show that our proposed tagger based on Levenberg-Marquardt learning algorithm is more effective and better than the traditional Backpropagation learning algorithm for tagging Arabic words.

7 Conclusion

The main aim of this paper is to design, implement and evaluate a system for tagging Arabic sentences. The limitations of the current Arabic tagging systems and the modesty of the accuracies of the available systems have induced the author to investigate a novel approach to build a POS-tagger for Arabic language.

POS-tagging became very important in natural language processing. It is usually the first step in linguistic analysis. Also, it is a very important intermediate step to build many natural language processing applications.

Tagging Arabic words is a difficult task and a few efforts have been done in Arabic part of speech tagging. In this paper we have presented and examined the application of artificial neural networks to the task of POS-tagging of Arabic sentences. The network is trained with the help of Levenberg-Marquardt algorithm. To ensure the developed tagger accuracy, a corpora of 24,810 distinct words are collected and manually tagged in order to train and to test the performance of the developed POS-Tagger. Two experiments are conducted separately. The first one is performed on the training dateset (19,848 words) and the second is performed on the testing dataset (4,962 words). The developed POS-tagging achieved significant results of 98.83% and 90.21% for the training and testing datasets respectively.

By interpreting the results of the conducted experiments, we can conclude that the Levenberg-Marquardt Learning neural network is an efficient approach and more effective than the traditional Backpropagation learning algorithm for tagging Arabic words.

References:

[1] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An introduction to natural language processing, computational lin-*

guistics, and speech recognition. Prentice Hall, 2 edition, 2008.

[2] Hasan Muaidi and Alaa Ayesh. An application of neural network for extracting arabic word roots. *WSEAS TRANSACTIONS on COMPUTERS 5(11)*, 4:2623–2627, 2006.

[3] Hasan Muaidi. *Extraction Of Arabic Word Roots: An Approach Based on Computational Model and Multi-Backpropagation Neural Networks*. PhD thesis, De Montfort University - UK., 2008.

[4] Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994.

[5] Ahmed Aliwy. *Arabic Morphosyntactic Raw Text Part of Speech Tagging System*. PhD thesis, Institute of Informatics, University of Warsaw, 2013.

[6] N. Awde and P. Samano. *The Arabic Alphabet How to Read and Write it*. Cairo, American University in Cairo Pree, 2000.

[7] Shereen Khoja. *APT: An Automatic Arabic Part-of-Speech Tagger*. PhD thesis, Computer Science Department Lancaster University, July 2003.

[8] H. Wahsheh and M. Al-Kabi. Detecting arabic web spam. In *The 5th International Conference on Information Technology*, pages 813–829, 2011.

[9] Levent Altunyurt and Zihni Orhan. Part-of-speech tagger for turkish. Technical report, Department of Computer Engineering, June 2006.

[10] Shihadeh Alqrainy. *A Morphological - Syntactical Analysis Approach For Arabic Textual Tagging*. PhD thesis, De Montfort University.UK, 2008.

[11] Anthony Mark McEnery. *Computational Linguistics: A Handbook and Tool-box for Natural Language Processing*. Sigma Press, Wilmslow, UK., 1992.

[12] Meryeme Hadni, Said Alaoui Ouatik, Abdelmonaime Lachkar, and Mohammed Meknassi. Improving rule-based method for arabic pos tagging using hmm technique.

[13] Web-Site. http://www.britannica.com/ebchecked/topic-art/259061, Accessed on March, 2014.

[14] Shereen Khoja. APT: Arabic part-of-speech tagger. Technical report, In Proceedings of the Student Workshop at the Second Meeting of (NAACL2001), Carnegie Mellon University, 2001.

[15] Mohammed Albared, Nazlia Omar, and Mohd. Juzaidin AbAziz. Arabic part of speech disambiguation: A survey. *International Review on Computers and Software*, 4(5), 2009.

[16] El-Kareh and Al-Ansary. An arabic interactive multi-feature pos tagger. In *Proceeding of the international conference on Artificial and Computational intelliegence for Decision Control and Automation in engineering and Industrial Application (ACIDCA) conference, Tunisia.*, pages 83–88, 2000.

[17] Andrew David Beale. Lexicon and grammar in probabilistic tagging of written english. In *ACL*, pages 211–216, 1988.

[18] Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. Automatic tagging of arabic text: From raw text to base phrase chunks. In *Proceedings of HLT-NAACL*, 2004.

[19] Nizar Habash and Owen Rambow. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *ACL*. The Association for Computer Linguistics, 2005.

[20] Fatma Al Shamsi and Ahmed Guessoum. A hidden markov model based pos tagger for arabic, 2006.

[21] Erwin Marsi, Antal van den Bosch, and Abdelhadi Soudi. Memory-based morphological analysis generation and part-of-speech tagging of arabic. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Semitic '05, pages 1–8, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[22] Chiraz Ben Othmane Zribi, Aroua Torjmen, and Mohamed Ben Ahmed. An efficient multi-agent system combining pos-taggers for arabic texts. In *CICLing*, volume 3878 of *Lecture Notes*

in Computer Science, pages 121–131. Springer, 2006.

[23] Jabar H. Yousif and T. Sembok. Arabic part-of-speech tagger based support vectors machines. In *Proceedings of International Symposium on Information Technology*, pages 2084–2090. Kuala Lumpur Convention Centre, 2008.

[24] Imad Abdulrahman Al-Sughayeir Yahya Ould Mohamed El Hadj and Abdullah Mahdi Al-Ansari. Arabic part-of-speech tagging using the sentence structure. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April 2009. The MEDAR Consortium.

[25] William Black Shabib AlGahtani and John McNaught. Arabic part-of-speech tagging using transformation-based learning. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April 2009.

[26] Bilel Ben Ali and Fethi Jarray. Genetic approach for arabic part of speech tagging. *International Journal on Natural Language Computing*, 2(3), 2013.

[27] Rabab Abu-Malloh. Arabic part-of-speech tagger: An approach based on neural network modeling. Master's thesis, AlBalqa Applied University, 2010.

[28] Laurene V. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Prentice Hall, 1994.

[29] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall Ptr, 1998.

[30] M. Hagan and M. B. Menhaj. Training feed-forward networks with the marquardt algorithm. *IEEE Transactions On Neural Networks*, 5(6):989–993, 1994.

[31] Lourakis MIA. A brief description of the levenberg-marquardt algorithm implemented by levmar. Technical report, Institute of Computer Science Foundation for Research and Technology-Hellas, 2005.

[32] Shihadeh Alqrainy and Aladdin Ayesh. Developing a tagset for automated pos tagging in arabic. Technical report, Centre for Computational Intelligence (CCI), De Montfort University, UK, 2008.

[33] Oscar Victor M. Antonio and Mark Albert H. Zarco. Neural network prediction of unconfined compressive strength of lime-fly ash stabilized bottom ash. Technical report, University of the Philippines, December 2006.