Proceedings of the 10th WSEAS International Conference on COMPUTERS, Vouliagmeni, Athens, Greece, July 13-15, 2006 (pp637-641)

# An Application of Neural Network for Extracting Arabic Word Roots

*HASAN M. ALSERHAN & ALADDIN S. AYESH*
*Centre for Computational Intelligence (CCI)*
*School of Computing*
*Faculty of Computing Sciences and Engineering*
*De Montfort University, Leicester*
*UNITED KINGDOM*

*Abstract:-* A good Arabic stemming algorithm is needed in many applications such as: natural language processing, computerized language translation, compression of data, spells checking and in information retrieval. Arabic language is considered to be one of the world's most complicated languages due to the complexity of its morphological structure, so it has huge morphological variations and rules. Majority of the existing Arabic stemming algorithms use a large set of rules and many algorithms also refer to existing pattern and root files. This requires large storage and access time. In this paper, a novel numerical approach for stemming Arabic words is described. The present approach attempts to exploit numerical relations between characters by using backpropagation neural network. The empirical positive results show that the stemming problem can be solved by neural network.

*Key-Words:-* Backpropagation Neural Networks, Stemming, Natural Language Processing (NLP), Arabic Language.

## 1 Introduction

Stemming refers to the process of reducing tokens to root/stem form of words to recognize morphological variation. This done by removing word affixes (prefixes and suffixes). For example, the words *computing*, *computer*, *computation*, *computerize*, and *computers* could be mapped to a common stem, *comput-*. In text search, this permits a search for "computers" to find documents containing all words with the stem *comput-*

Affixes in Arabic, unlike in English, are *prefixes*, *suffixes* and *infixes*. *Prefixes* are attached at the beginning of a word, where *suffixes* are attached at the end of the word, and *infixes* are found in the middle of the word. For example, the Arabic word "المعلومات" *"alma'lwmat"*, which means *"information"*, consists of the following parts: "ال" represent ***prefix-part***, "ات" represent ***suffix-part***, two "و", which is in the middle, represents ***infix-part***, and the remaining letters "علم" represent the ***root-part***.

Stemming can be used in many applications such as in data compression, in spell checking, and in information retrieval (IR) systems where many studies showed that using roots as an index word in IR gives much better results than using full words. It also can be used in text generation to generate different part-of-speech of a given word by attaching a certain affixes to the root verb. In spite of the rapid research conducted in other languages, Arabic language still suffers from the shortages of the researchers and

development.

## 2 Related Works

### 2.1 Arabic Language Overview

Arabic is the language spoken by over 200 million people throughout the Middle East and North Africa. Plus, at least another 35 million speak Arabic as a second language. It is one of the six official languages of the United Nations, the language of Islam and its holy book the Qur'aan, and the language in which some of the world's greatest works of literature, science and history have been written.

Arabic language is considered to be a member of a highly sophisticated category of natural languages, which has a very rich morphology. In this category one root can generate several different words that have different meaning.

Like other Semitic languages, Arabic words are generally based on a root that uses three consonants to define the underlying meaning of the word. Each Arabic word can be reduced to, or stems from, a root. 90% of Arabic roots are 3 letters, few are 4 letters and fewer still are 5 letters. Various vowels, prefixes and suffixes are used with the root letters to create the desired inflection of meaning. For example, table 1 shows some words based on the root "كتب" "ktb" which means "write", and each word has its own meaning.

| Arabic Form | Pronunciation | Meaning |
|---|---|---|
| كاتب | *kaatib* | *a writer* |
| الكتاب | *alketab* | *the book* |
| مكتبة | *maktab* | *an office* |
| يكتب | *yaktubu* | *he writes* |
| أكتب | *'aktaba* | *he dictated* |

Table 1: *Some of derived forms of the root "كتب"*

The vowels and the non-root letters (affixes) give the word its specific meaning. The pattern of vowels and non-root letters itself carries meaning. For example, the pattern "$mC_1C_2aC_3$" (where $C$ is a root letter) means "place", so mktab "مكتب" means "place of writing", malعab "ملعب" (root "لعب" "lعb" "play") means "playground", and matbakh "مطبخ"' (root "طبخ" "tbkh" "cook") means "kitchen".

### 2.2 Stemming Techniques

Stemmers are commonly designed for each specific language. Stemmers design requires some linguistic expertise in the language itself. For English language, stemming is well understood, with techniques such as those of Lovin [5] and Porter [6]. Many Stemmers have been implemented for many languages. Due to its non-concatenative nature, Arabic is very difficult to stem[4], so regrettably there are only few stemmers in Arabic.

In literature, stemmers can be classified into table lookup, linguistic, and combinational approaches.

Table lookup approach [3] utilize huge list that stores all valid Arabic words along their morphological decompositions. This method does not use stemming process. For a given Arabic word, it access the list and retrieve the associated root/stem. Consequently, the stems obtained are guaranteed to be highly accurate. However, the availability of such a table that should include all the language words is practically impossible.

Linguistic approach[1], attempts to simulate the behavior of a linguist by considering Arabic morphological system and thoroughly analyzing Arabic words according to their morphological components. In such approach, prefix and suffix of a given word are removed by comparing leading and trailing letters with a given list of affixes. In literature, most of the published works were mainly linguistic-based.

Finally, in Combinational approach[1], a given word is used to generate all combinations of letters. These combinations are compared against predefined lists of Arabic roots. If

matched, stem and patterns are extracted.

### 2.3 Backpropagation Neural Network (BPNN)

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, such as character recognition[2].

Typically, BPNN architecture consists of three layers[2]: input, hidden, and output layers. Figure 1 shows BPNN architecture in which 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 serves to provide the internal relations between input and output layers.

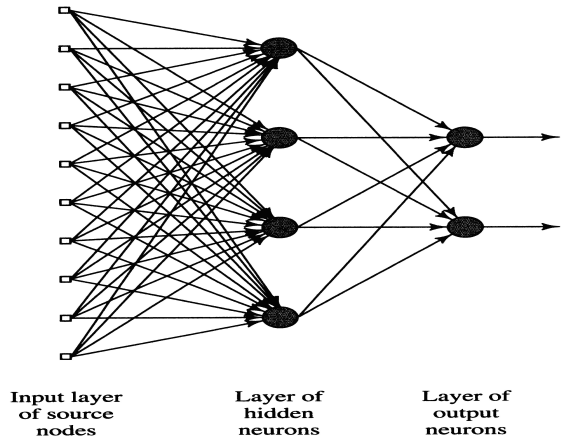


Figure 1: *Backpropagation Neural Network*

## 3 Methodology

The purpose of this study is to show numerical relations between characters can be extracted for the purpose of fast accurate stemming. This is inspired by linguistic studies that identify the likely redundant characters in words. A neural network approach for extracting Arabic roots without referring to root or pattern files is proposed here and proven to work.

Backpropagation neural network architecture is used. The model was trained to accept an encoded Arabic word as an input and to generate the correct root of that word as an output.

### 3.1 Data

Most Arabic words are derived from verb forms which extend or modify the meaning of the root form of the verb, giving many shades of meaning. The simple or root form of the verb is called the "stripped" verb, while the derived form is said to be the "increased" verb. Several hundreds of Arabic words can be derived from a single root by adding affixes. Arabic linguists have identified the commonly used letters in these affixes and presented them in the following set: {س,أ,ل,ت,م,و,ن,ي,هـ,ا}. The Arabic linguistics experts classified these letters according to their frequency in appearing as affix letters [7]. The highest frequently used are {ا,و,ي}, then {ت,ن,م,أ}, and the lowest are {ل,هـ,س}. Depending on this, we classify all Arabic letters into four classes and assign a numeric value to each class as shown in table 2. These values are used in the network as inputs.

| Class | Decimal Code | Binary code |
|---|---|---|
| ا,و,ي | 4 | 100 |
| أ,م,ن,ت | 3 | 011 |
| س,هـ,ل | 2 | 010 |
| باقي الحروف | 1 | 001 |

Table 2: *Arabic Letter Classification*

Any letter in the input word is encoded as three binary digits, as shown in table 2. Limiting an input word to length of four letters, gives us an input matrix of 4×3, where each row represented contains three binary digits code for the letter.

The output matrix will also 4×3. If the input letter in any given word is in the root word, we presented it by three ones "111", other wise it presented by three zeros "000".

### 3.2 Network Architecture

The architecture of the model is summarized in Figure 2. The network has an input layer of 12 neurons (4 letters and each letter represented by 3 bits), followed by one hidden layer consists of 12 *tansig* neurons. The hidden layer is fully connected to the input layer, output layer. The number of hidden neurons was estimated experimently. The output layer has 12 *purelin* neurons.

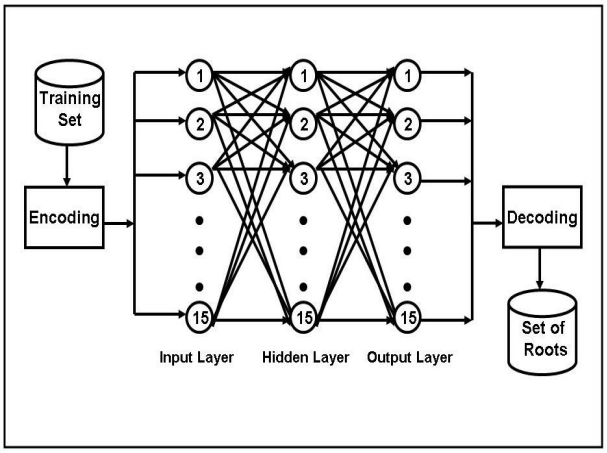


Figure 2: *The Proposed Model Architecture*

The network was trained by *trainlm* backpropagation learning algorithm with learning rate $\eta = 0.5$. Training was performed until an acceptable convergence was found for 515 epochs with mean squared error less than 0.01.

The procedures and functions of the software package "MATLAB" were used to design, train, and test the proposed neural network.

## 4 Experiments and Results

The data for this study were taken from a generated lexicon. A shell program was implemented by the first author. The program generated a set of words from a given set of roots. The lexicon contains about 9000 Arabic words and its correct word roots. A set of 1000 randomly chosen words was selected from this lexicon, restricted to words of length four letters. The correct output Arabic root for each word was available to the model during training.

The trained model has been tested using a set of 250 randomly chosen words from the generated lexicon. The word length was four letters. Comparing the output from the trained network with the actual roots, it shows that the network produces 209 correct roots from the given test set. The proposed technique has proved a positive behavior with accuracy rate 84%. Table 3

| Word | Input Matrix | Root | Output Matrix |
|---|---|---|---|
| كاتب | 001100011001 | كتب | 111000111111 |
| ملعب | 011010001001 | لعب | 000111111111 |
| لاعب | 010100001001 | لعب | 111000111111 |
| سعيد | 010001100001 | سعد | 111111000111 |
| لعبة | 010001001100 | لعب | 111111111000 |
| يدرس | 100001001010 | درس | 000111111111 |

Table 3: *A Sample of Tests Results*

shows a sample of tests results. While, figure 3 shows how the network extracts the Arabic root "سعد" from the input Arabic word "سعيد", as an example.

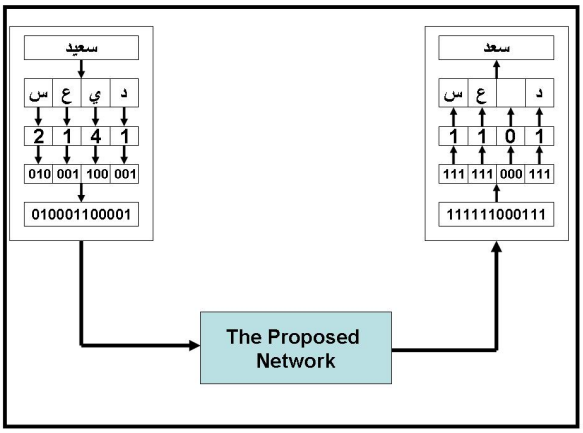


Figure 3: *Extraction of Root "سعد" from the word "سعيد"*

## 5 Conclusion

In this paper, we presented a backpropagation neural network approach to the problem of stem-

ming the Arabic words.

A neural network can be used to solve a variety of problems. However, the successful implementation depends on the nature of the problem. Stemming Arabic words is not a straight forward kind of problem, which appears at first time to be not quite suitable for neural networks. However, the positive results that we have obtained through our experiments, show that a neural network can be used for stemming Arabic words.

Currently, the process is limited to triliteral roots and four letters Arabic words length. Future work will concentrate on extending our approach to cover all length of Arabic words and roots.

## References

[1] Ibrahim A. Al-Kharashi and Imad A. Al-Sughaiyer. Rule merging in a rule-based arabic stemmer. In *COLING*, 2002.

[2] Laurene V. Fausett. *Fundamentals of Neural Networks*. Prentice Hall, 1994.

[3] Jerey Coombs Kazem Taghva, Rania Elkhoury. Arabic stemming without a root dictionary. *International Conference on Information Technology: Coding and Computing (ITCC'05)*, 1:152–157, 2005.

[4] Larkey, Leah S., Lisa Ballesteros, Connell, and Margaret E. Improving stemming for arabic information retrieval: light stemming and co-occurrence analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Arabic Information Retrieval, pages 275–282, 2002.

[5] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1-2):22–31, 1968.

[6] Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[7] Fagher Aldeen Qabaweh. *Tasreef Al Asma' wa Al Afa'l*. Al Maerf Library, 1988.