List of Contents

List of Figures

List of Tables

Chapter I

Introduction

1- Introduction

1-1 Arabic languages

Today, Arabic is one of the few living languages spoken by millions of people, and at world conferences, it is one of the four languages, English, French, German and Arabic, translated simultaneously in lectures and conversations. It is the language spoken by millions of people and the official languages of many countries [1].

The preamble to the UN resolution recognizing Arabic is one of the recognized languages of the United Nations states that the General Assembly recognizes the important role of the Arabic language in the preservation and promotion of human civilization and culture and recognizes that the language of 19 Is officially a member of the General Assembly, but Arabic is now the official language in 22 countries. In 2010, the United Nations celebrated its six official languages and designated December 18 as International Arabic Language Day. The first celebration of World Arabic Day was held by UNESCO in 2012 and called for the official and international promotion of the Arabic language on this day.

Currently, more than 430 million people worldwide, including 22 Arab countries and several provinces or states in Iran, Turkey, Chad, Niger, and Mali, speak Arabic. Also, more than one and a half billion Muslims in the world use this language for rituals and worship, especially for reciting the Holy Quran and getting acquainted with tradition and jurisprudence [1].

Eloquent (Classical) Arabic is, in fact, a dialect that stood out from the rest of the dialects common among the Arabs in the past. The Arabs had several dialects and even languages, so that each tribe had its dialect without being disconnected from the different dialects [1, 2].

Introduction

The variety of dialects and the derivative power of the Arabic language have given Arabic extraordinary word-building power, so much so that the prominent Iraqi scholar Javad Ali in his book "Pre-Islamic Arab History" estimates the number of words in this language at more than 12 million. Arabic can be considered the most prolific language in human history, and there are few living languages in the world today that have not received a word from Arabic [2].

So that Persian, Turkish, Kurdish, Urdu, Hindi, Malay, Baluchi, and Pashto or Patan have more Arabic words and phrases than any other language, and among the major languages of the world, Spanish has been influenced by Arabic more than any other European language. Some words and letters are rooted in Arabic in almost all languages of the world, such as the names of coffee and alcohol, and the names of some stars, and scientific and philosophical terms in most languages are also rooted in Arabic. Scattered countries in the world that speak Arabic, as shown in Figure 1. Table 1 also lists 22 countries in the world based on population and language statistics.

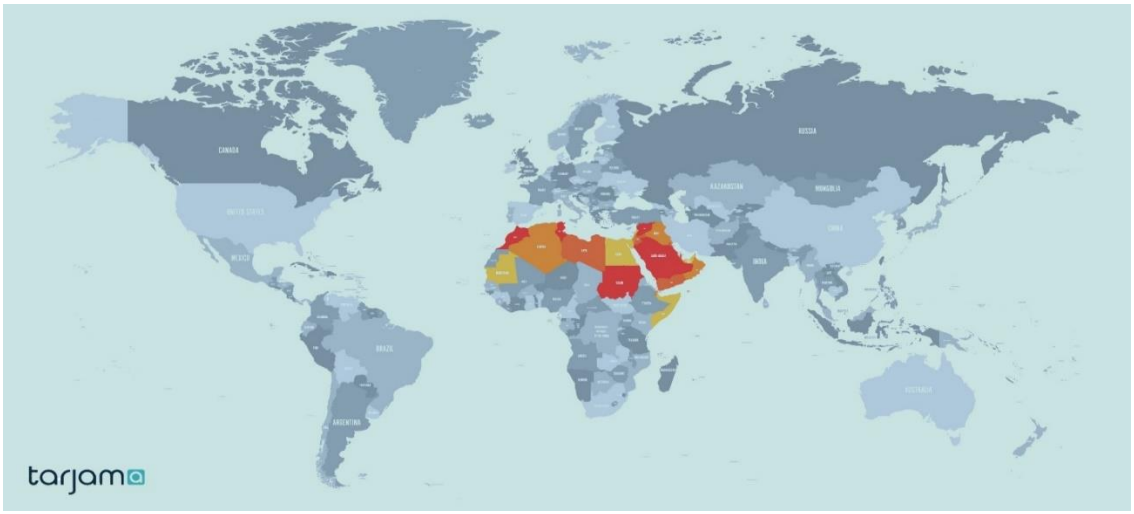

Figure 1 Scattered countries in the world that speak Arabic (tarjama.com)

Table 1 22 Arabic countries in the world based on population and language statistics[1]

No	Country	No. of Arabic Speakers	Population
1	Egypt	82,449,200	100,000,000
2	Algeria	40,100,000	41,701,000
3	Sudan	28,164,500	40,235,000
4	Saudi Arabia	27,178,770	30,770,375
5	Morocco	25,003,930	35,250,000
6	Iraq	22,908,120	36,004,552
7	Syria	17,951,639	20,956,000
8	Yemen	14,671,000	23,833,000
9	Tunisia	10,800,500	10,982,754
10	Jordan	5,083,300	6,655,000
11	Libya	4,526,000	6,244,174
12	Lebanon	4,180,000	4,965,914
13	Somalia	3,788,000	10,428,043
14	Palestine	3,762,076	4,484,000
15	United Arab Emirates	3,607,600	9,346,129
16	Mauritania	3,140,000	3,359,185
17	Oman	2,518,816	4,055,418
18	Kuwait	1,735,000	2,789,000
19	Chad	1,320,000	10,329,208
20	Qatar	1,215,900	2,155,446
21	Bahrain	690,302	1,343,000
22	Eritrea	249,700	6,380,803

Of course, Arabic, as a living language that has been in contact with other cultures and languages throughout history, has also accepted words. Therefore, ten characteristics the of Arabic language can be summarized as follows.

[1] www.tarjama.com

Introduction

- It is one of the oldest Semitic languages and one of the most widespread languages in the world.
- More than 430 million people in the world speak Arabic.
- It is called “anti-language” in Arabic because it is the only language that has this letter and is pronounced.
- It is among the four most widely used languages on the Internet.
- The Arabic alphabet has 28 basic letters, although, in other languages that use Arabic letters, such as Urdu, Persian, Kurdish, and Uighur, more letters have been added to the basic Arabic letters.
- The number of Arabic words without repetition has reached more than 12 million words.
- Arabic was initially written without punctuation, but punctuation developed in the seventh century AD, while the punctuation system was added in the eighth century.
- Maltese, which is one of the European languages, is derived from Arabic.
- The current Arabic language is at least 1700 years old and is related to Nabataean, Aramaic, Assyrian, Hebrew, Chaldean, Amharic, and Tigris languages, respectively.
- More than one and a half billion Muslims in the worldwide and some Christians in the Arab world also use Arabic to perform religious rites.

There are many types of Arabic language, such as, Classical (Qur’anic) Arabic, Modern Standard Arabic, Colloquial Arabic, etc. In this study, we focus on Modern

Standard Arabic (MSA), which contains 28 letters and eight diacritics. Modern Standard Arabic is the classic version of the language known as Modern Standard Arabic and is used throughout the Arab world in particular, and general in the world. However, is generally used in both formal and written contexts: education, television/radio news programs, newspapers, etc., are commonly used. One of the most essential, critical parts of the Arabic language is diacritics. The concept of diacritics is explained in the section [1, 2].

1-2 Diacritics in Arabic

Diacritics in Arabic are the same symbols that are placed on the letters. The Figure 2 below; Shows their location in an Arabic sentence. The Table 2 below lists these eight diacritics [1]. There are eight essential diacritics in Arabic that can be categorized into 3 different categories [3]:

- Three marks of the doubled case endings (Tanween): Tanween Damm that we say Dammatan, Tanween Fath that we say Fathatan, Tanween Kasr that we say Kasratan.
- Three marks of the short vowels: Damma, Fatha, Kasra.
- . Two marks of the syllabification marks: Sukoon and Shadda [3].

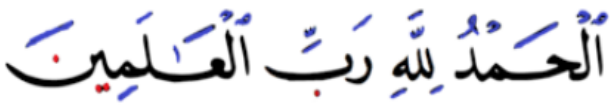

Figure 2 Diacritics are placed on letters in Arabic.

Table2 : Eight diacritics in Arabic

Diacritic	Arabic name	Transliteration
دَ	فَتْحَةٌ	Fatha
دِ	كَسْرَةٌ	Kasra
دً	تَنْوِينُ فَتْحٍ	Tanween Fath
دٍ	تَنْوِينُ كَسْرٍ	Tanween Kasr
دُ	ضَمَّةٌ	Damma
دْ	سُكُونٌ	Sukoon
دٌ	تَنْوِينُ ضَمٍّ	Tanween Damm
دّ	شَدَّةٌ	Shadda

1-3 Problem definition

In this section of chapter 1, we introduce the research problem. The Arabic speaking community has one of the most outstanding growth rates on the Internet and social media use. Therefore, interest in Arabic Natural Language Processing (NLP) has increased over the years. Unfortunately, however, the work on Natural Language Processing of the Arabic language is far behind concerning to the natural languages of other languages, such as English, Chinese and French, due to the weak efforts invested in Natural Language Processing of the Arabic language and the lack of linguistic resources available to researchers and developers, and the difficulty of the Arabic language, which has many problems. diacritics play an important role in translating the Arabic language. A natural, asymptomatic language processing application may run into problems [4].

Today, in most Arabic texts, these signs are removed. Of course, these signs still exist in the purchase of items such as Arabic teaching texts or important documents. The reason for removing these signs from texts is that Arabic speakers

and those whose mother tongue is Arabic, usually read and understand the text without difficulty, but these signs are still an important problem for non-Arabic speakers and intelligent systems based on natural language Processing.

The full diacritization problem includes two sub-problems: morphological diacritization and syntactic diacritization. The first indicates the word's meaning, and the second shows the grammatical case [5].

By definition, a syntactic diacritic mark is the last diacritical mark on the last character of the word. A word can have different dependent diacritics, which is defined based on the decomposition tree of a sentence. Figure 3 illustrates this well.

The word "الانسان" can have different syntactic diacritic marks depending on the given sentence and its parsing tree based on Arabic grammar. this problem can be formulated as a word-level tagging approach. This approach needs the diacritics marks for any word in the corpus.

For example, the Arabic sentence "علم الانسان ما لم يعلم" consists of 5 words, and we need one diacritic mark only for each word to train our models. Part of speech (POS), prefix, and suffix annotations for each word in the extract sparse features [6].

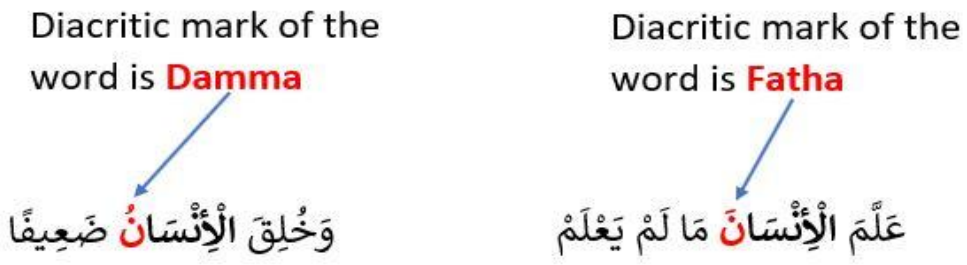

Figure 3 The word الانسان in the two sentences in the exact location (i.e., the second word in the sentence), but they have different syntactic diacritics marks (Fatha and Damma) based on the Arabic grammar [6]

For morphological diacritization, the automatic diacritization problem is an essential topic due to the high ambiguity of the undiacritized text and the free word order nature of the grammar. Table 3 illustrates the differences made by the possible diacritizations of the word علم. As one might see, the diacritization defines many linguistic features, such as the part-of-speech (POS), the active/passive voice, and the grammatical case [5].

Table 3 The word علم with their diacritizations can produce several meanings [5]

Word with diacritics	Meaning the Word
عَلِمَ	He knew
عَلَّمَ	He taught
عُلِمَ	It was Known
عُلِّمَ	It was taught
عَلَمٌ	A flag (nominative)
عَلَمٍ	A flag (genitive)
عِلْمٌ	A science (nominative)
عِلْمٍ	A science (genitive)

To create a better understanding of semantic changes and the position of the word in the sentence, you can also refer to the Table 4 below.

Table4 the position of the word علم in the sentence [5]

English meaning	Example
knowledge	و لقد كان لى مخالطتكم علم
knowledge	ما لدى الانسان من علم
know	فمن علم فيه صبرا
knowledge	يحيط به علم انسان
mountain	كانه علم فى راسه نار
Have been taught	من علم علما
Flag	علم الكويت رفع فى 24 نوفمبر 1961
teach	الذى علم بالقلم

In the past, traditional methods were used, such as the manual method is required human knowledge of Arabic grammar, to restoration diacritic marks in the Arabic language in order to distinguish between two words containing the same letters in number and writing. However, with different meanings, that method is straightforward and take time-consuming.

Due to the development of science and technology and the wide advancement of tasks related to the Arabic language, such as machine translation and several other tasks, many works compete to show new and unconventional methods for the purpose of restoring the diacritics of the Arabic language without human intervention, one of these methods used rules-based approach such as (Weighted Finite-State Transducers, etc.) [7] this method are flourishing as the error rates

were very few, However, this method requires a vast set of rules to contain the problem of restore diacritic marks, and this task is almost impossible because the Arabic language is from complex languages and one of the other methods statistical-based approach such as (Hidden Markov Models, N-grams models and, etc.) was used on word level and character level [8-10].

In last decease, deep learning techniques have been used widely, especially recurrent neural networks (RNN), Long Short-Term Memory (LSTM) were used techniques to restorative diacritics in the Arabic language that proven to be a high-performance technique and achieve high results compared to techniques other than deep learning.

Although Deep Learning achieves high results, that results from it needs some corrections at the level of characters and words output to achieve good outperformance, and we, believe that it is possible to obtain good results when combining Rule-based and Statistical-based techniques and using deep learning techniques to produce a hybrid approach with good results for restore diacritic marks in the Arabic language, and it can solve most of the problems of restoration diacritic marks in the Arabic language to get to the intended meaning of a particular word.

In this research, a network based on Bi-LSTM deep learning, and CRF with two-level inputs is presented. The detection results of this algorithm are then improved by making some post-processing and corrections.

1-4 Natural Language Processing

Natural language processing is one of the most important subfields of artificial intelligence technology, which refers to the interaction between computers and human languages. In other words, natural language processing focuses on the human-computer relationship. The main challenge in this field is to understand natural language and to mechanize the process of understanding the concepts expressed in a natural human language.

The primary purpose of NLP is to create computational theories of the language, using algorithms and data structures in computer science. so, in order to reach this goal, there is a require for extensive knowledge of the language, and in addition to computer science researchers, there is a need for the knowledge of linguists in this field. By processing linguistic information, the statistics needed to work with natural language can be extracted. Natural language processing applications can be divided into two general categories: written applications and spoken applications. Each of these categories contains different topics. Which sometimes form a framework together. In recent years, this field of research has attracted the attention of scientists, and considerable research has been done in this field [13].

Nevertheless, despite much research, natural language processing has not yet received adequate attention for the Arabic language [13]. The typical phases of ANLP (ARABIC natural language processing) applications-based machine learning are shown in the below Figure 4 [14]. Essential parts are shown in this Figure.

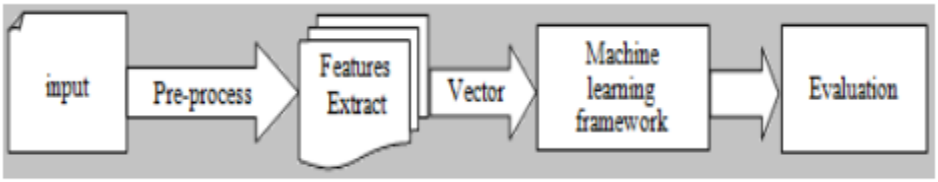

Figure 4 The typical phases of ANLP (ARABIC natural language processing) applications-based machine learning [14]

Also, As shown in Figure 5, to process natural Arabic language, one must have a good knowledge of the concepts of artificial intelligence, linguistic concepts, especially in Arabic, and computer science concepts. These disciplines will lead to a proper understanding and good application in everyday applications [13, 14].

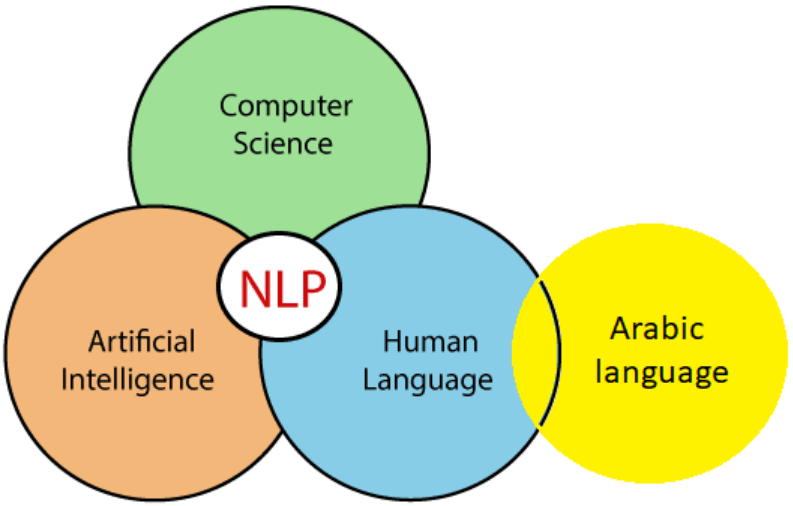

Figure 5 Natural language processing in Arabic

1-5 Deep learning

Deep learning is a machine learning method that teaches computers to do what humans usually do: Learn by example. Deep learning is the primary technology of

making cars; Deep learning gives cars the ability to recognize signs or distinguish electric light poles from humans. In-depth learning is the key technology used in the audio interface of mobile phones, tablets, TVs, and handsfree [11]. Deep learning can also be used in natural language processing. What we are looking for in this research. In short, deep learning has been gaining traction for some time; Because it has brought with it results that have never been possible before.

In-depth learning, a computer model executes classification commands directly from an image, text, or sound. Deep learning models can achieve the highest level of accuracy; So that sometimes they do better than humans. Deep learning models are trained using large sets of data and multi-layered neural networks.

How does deep learning achieve these fantastic results? In a word, accuracy. Today, deep learning has achieved a very high level of accuracy in diagnosis [11]. This precision enables electronic devices to meet the expectations of users. Recent advances in deep learning have reached the point wherein some tasks, such as categorizing images or reviewing texts and processing natural language, deep learning works better than humans. This tool will also play an essential role in our research [11, 12]. The second chapter will provide more details about it.

1-6 Research structure

This research is presented in five chapters. The first chapter introduces the subject. In the second chapter, related works were presented. The most crucial goal of this chapter will be to adequately express the existing challenges and present a new method that has appropriate innovation. In the third chapter, the method proposed in this research is presented. This method based on four layers, and we performed

the proposed method in the chapter four. The Implementation results must be able to detect the symbols accurately. The fifth chapter is a summary and conclusion. Hence, for this research, there are the following assumptions.

- Since there are some non-free data sets, so we will try to use free data sets after some preprocessing for it, such as remove URLs, delete lines that have no useful information, such as blank lines.
- Since we have 8 Diacritics marks, there will be 15 class Diacritics marks, i.e., with two Diacritics marks for one letter (i.e., Shadda + another diacritic) in addition to no diacritics mark.
- This work is not for solving class overlapping if it existed in the selected dataset.
- Data cleaning and normalization are out of our scope.
- Each letter or number within a word may be a potential host of a set of diacritics.
- The project design may consist of two or more levels or stages that can cooperate to produce the required output.
- All diacritics are counted on one letter as a single binary option.
- We assume that the selected dataset is Tashkeela, however, if we couldn’t manage the dataset in our work for one reason or another, we removed the dataset and replaced it with another.
- Feature selection is out of our scope, even, if we mentioned, we might use the features mentioned in the literature, or we use another manner for feature selection like existed tools.

In the following chapters, using the concepts and hypotheses presented, the proposed research method will be introduced and Implemented.

Chapter II

Background and Preliminaries

2- Background and preliminaries

2-1 Introduction

In this chapter, we will review background information and related work. Background information is, in fact, a review of the concept of recursive neural networks as the structure and learning tool of the proposed method. Below will be reviewed the related work. This chapter makes the position of the proposed method clear.

2-2 Background

In this section, we will review recursive neural networks.

2-2.1 Feed Forward Neural Network

Another old type of neural network is the feed forward neural network, also called FF for short; The approach to this type of neural network dates back to the 1950s. The feed neural network (FFN) is a type of neural network also known as the Multi-Layer Perceptron (MLP); In fact, several layers of perceptron that are connected to each other form a feed neural network. A feed neural network has an input layer, intermediate or latent layers, and an output layer that are interconnected. These layers are made up of several perceptron's or neurons, so the basic rules of this type of neural network include:

- Connect all nodes together.
- Activation from input layers to output layers without any reversing or reversing loops.
- Existence of a layer between input and output as a hidden layer.

The information travels along this network from the input to the output and passes through different layers to finally provide the desired output. There is no loop in this network that returns the network output as input to the network itself; For this

reason, this network is called the feed network. Therefore, this structure is not suitable for applications such as natural language processing, because in such applications there is a series of time and there must be backwardness and feedback. There is another type of neural network that has a loop and the output re-enters the network itself, called the Recurrent Neural Network (RNN). We will explain it below.

2-2.1 Recurrent Neural Network

Artificial neural networks are a functional unit for learning. However, a large part of the data, such as speech, time-based data, or so-called time series, data received from sensors, videos, text, and … Are inherently series (sequential). The main problem of Traditional Neural Networks is that they have no information, meaning that cannot predict subsequent events because they do not have information about past events, and here came the purpose of the idea of Recurring Neural Networks (RNN), where they take information from previous cells as input to current cells, so it predicts future events. Recurrent Neural Networks address this issue [15].

Recursive neural networks were created in 1980, but it has only been in recent years that such networks have been widely is used. One of the main reasons for such an event is the progress made in the design of neural networks in general and the significant improvement of computing power, and particular, the efficiency of the power of parallel processing units of graphics cards. These types of neural networks are beneficial for processing confidential or sequential data, in which each neuron or processing unit can maintain an internal state or memory in order to store information related to previous input. This feature is significant in various applications related to confidential data. For example, in natural language processing, a sentence like “I had washed my car” has a different meaning, such as

“I had my car washed”. In these two sentences, all the words are the same, but the meaning of each is entirely different. In the first sentence, we say “I washed my car” while in the second sentence, we say, “I ordered my car to be washed” (another person or group washed the car on my order) (they were hired or I “I ordered that 'others' wash my car in any case”. [15, 16]

This feature helps the network to maintain the internal state or memory capability to be able to understand and discover the connection between different words in longer sequences. It should be noted that we humans when we read a sentence, infer its meaning according to the context in which each word is placed. In other words, we deduce the content context according to the previous and (in some cases even the next) words and understand the meaning of a word [16].

The main idea behind this type of architecture is to exploit this data series structure. The name of this neural network derived from the fact that these types of networks operate in reverse. That is, the perform an operation for each element of a sequence (word, sentence,…), and its output depends on the current input and previous operations. This is done by repeating one output of the network at a time with the input of the network at a time. (That is, the output from the previous step is combined with the new input in the new step.) Cause. In other words, these networks have a loop inside them through which they can pass information to the neurons while reading the input.

Generally, in connection with this type of network, will encounter symbols and shapes such as the following:

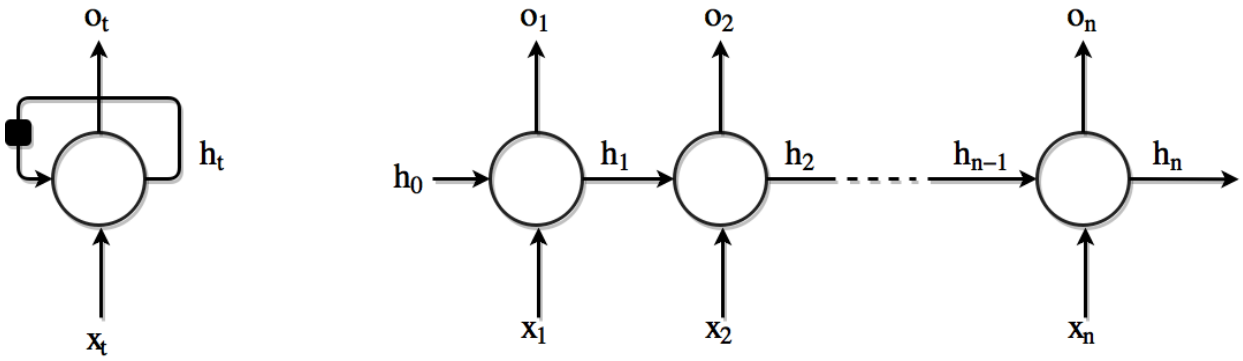

Figure 6 An Unrolled RNN structure

(Figure 1. Left side. Circuit diagram). The black square represents the time delay of one-time step. Right: The same grid is displayed as a computed graph. Each node is associated with a specific time. (Refers to a time step)

This structure with cycles can be a little confusing. Nevertheless, when we look at the chain formed after opening this computational graph, this becomes quite understandable. We now have an architecture that can receive multiple inputs at each x_t time step and generate O_t outputs at each time step, as well as an h_t memory state that contains information about what is on the network up to time t Keep data in itself [16].

For each timestep t_t, the activation $a^{<t>}$ is expressed as follow

$$a^{<t>} = g\,(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

the output $y^{<t>}$ is expressed follow:

$$y^{<t>} = g\,(W_{ya}a^{<t>} + b_y)$$

Where W_{ax}, W_{aa}, W_{ya}, b_a, b_y are coefficients that are shared temporally and g activation functions.

2.2.2 LSTM

The problem with the RNN it cannot process very long sequences because it has not the memory, which makes it difficult to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process, and predict time series given time lags of unknown duration. LSTM, which stands for Long Short-Term Memory and it is used for the purpose to learn on long dependencies, and is designed to avoid fading data from the long dependencies. Long Short-Term Memory came to solve the problem of the chain of transmission of data between cells of the Recurrent Neural Network RNN, where it was found that the transmission of data between the long chain of Recurring Neural Network RNN cells leads to the loss of data. In Long Short-Term Memory is a time series of data that passes through several gates, usually three gates, as we see in Figure 7.

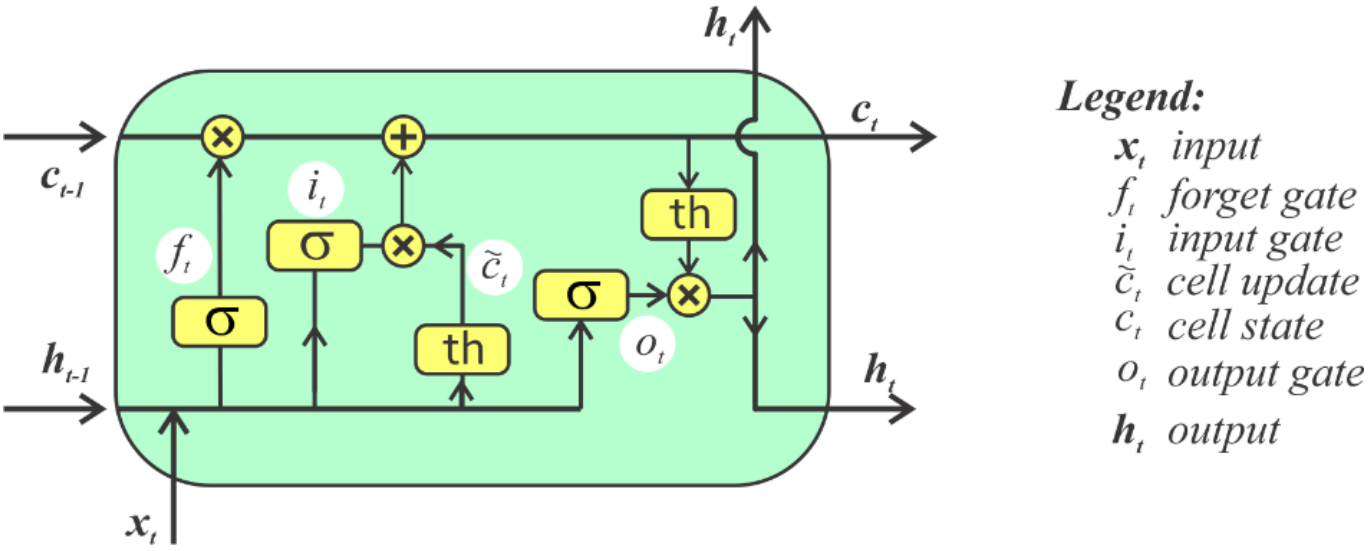

Figure 7 A cell of LSTM Architecture[2]

[2] https://www.rs-online.com/designspark/predicting-weather-using-lstm

Which are the forget gate, the input gate, and the output gate, gates in LSTM are the sigmoid activation functions, i.e., they output a value between 0 or 1, the "0" means the gates are blocking everything, and the "1" means gates are allowing everything to pass through it, and the function of each gate is to calculate certain pre-set cases.

The equations for the gates in LSTM are:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

[3]

Figure 8 The equations for the gates in LSTM[4]

The first equation is for Input Gate, which tells us that what new information we're going to store in the cell state, the second is for the Forget Gate, which tells the information to throw away from the cell state, and the third one is for the output gate which is used to provide the activation to the final output of the LSTM block at timestamp 't'.

[3] {i_t represents input gate, f_t represents forget gate, o_t represents output gate, σ represents sigmoid function, w_x weight for the respective gate (x) neurons, h_{t-1} output of the previous LSTM block, x_t input at current timestamp, b_x biases for the respective gates (x)}

[4] https://medium.com/@divyanshu132/lstm

The equations for the cell state, candidate cell state, and the final output:

$$\tilde{c_t} = tanh(w_c[h_{t-1}, x_t] + b_c)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c_t}$$
$$h_t = o_t * tanh(c^t)$$

[5]

Figure 9 The equations for the cell state[6]

We can see that at any timestamp, our cell state knows that what it needs to forget from the previous state (i.e., f_{t} * c_{t-1}) and what it needs to consider from the current timestamp (i.e., i_{t} * c`_{t}).

2.2.3 Bi-LSTM

A Bidirectional LSTM [17], is a sequence processing model that is consisting of two layers of LSTM, as we see in Figure 10, forward and backward, one taking the input in a forward direction and the second in a backward direction. Bi-LSTM [17] is widely used in natural language processing, in particular improving the context available to the algorithm (e.g., knowing what words immediately follow and precede a word in a sentence) and effectively increase the amount of information available to the network.

[5] {C_t represents cell state (memory) at timestamp(t), $\check{C}_t$ represents candidate for cell state at timestamp(t)}

[6] https://medium.com/@divyanshu132/lstm

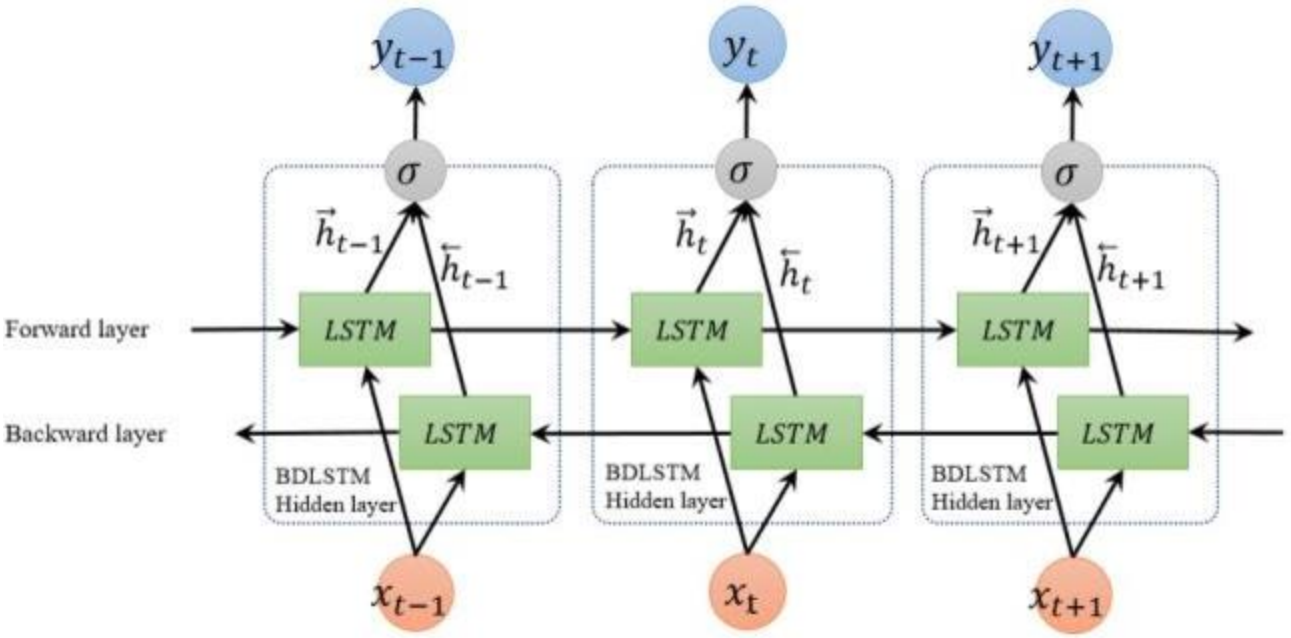

Figure 10 Bi-LSTM Architecture[7]

2.2.4 CRF

CRF is a stand for Conditional Random Field. The CRF that good content performance in sequence labeling often uses the CRF layer instead of the Softmax[8] layer to predict the network output CRF is more powerful than Softmax in particular of sequence dependencies in the output layer which exist in the diacritization problem. Furthermore, as we said previously considered to be a best practice in sequence labeling, but it increases the time complexity of any model, which is primarily affected by the input sequence length and output sequence length. In actuality, considered used the CRF layer instead of Softmax to complement the

[7] http://www.gabormelli.com/RKB/Bidirectional_LSTM_(biLSTM)_Model

[8] Softmax layer is used to predict the network output i.e., the final output layer in a neural network that performs multi-class classification.

BiLSTM for the classification layer. The standard maximum smooth function, sometimes called the Unit Softmax Function, is defined as follows.

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \text{ for } i = 1, \ldots, k \text{ and } z = (z_1, \ldots, z_k) \in R^k$$

Note that for each element z of the input vector z, a standard exponential function is applied, and dividing each value by the sum of all of them normalizes and ensures that the sum of the components of the output vector is one.

$$\sigma(z) = 1$$

Linear-chain CRF (L-CRF) is One of the most model used in natural processing language; as we show in Figure 11, the conditional density of a set of class labels Y has given a set of observations X. The density of an L-CRF is defined over a sequence of class labels Y (empty ovals) given a sequence of observations X (filled ovals) where there is a normalizing factor. The density models the dependency of the class labels not only on the observations but also on the other class labels. This property enables relational data to be processed.

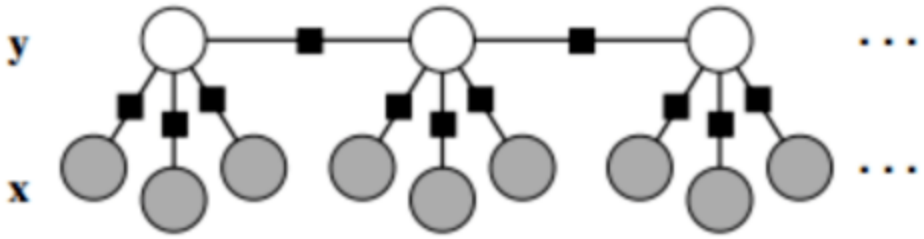

Figure 11 Linear Chain CRF Architecture[9]

2-3 An overview of related works

Reviewing related work will provide a deeper understanding of the research. As we mentioned earlier, diacritics in the Arabic language and its effect on the meaning of the word, according to the research, most of the proposed systems depend mainly on grammar, dictionaries, and linguistic resources that use linguistic information. On the other hand, little effort was made based on of data alone, so that we will review several previous works in the field of Arabic diacritization. So, in this section, the methods used to restore diacritization in Arabic will be reviewed for each article. The techniques used in diacritization are often categorized into three main categories:

- rule-based approach
- statistical approach
- hybrid approach

[9] https://www.commonlounge.com/discussion.

2.3.1 Rule-based approaches:

The authors in this article [18] borrowed the diacritic text to diacritize the highly cited text and also evaluated a strategy for shaping the heavily cited texts by borrowing the diacritics from their citations. This method exploits the manual diacritize process for other works and reuses it. To exploit fully diacritic text from various origins. This method is partially reworked text as a rendering source, by matching the non-written version of one word in the target text to its equivalent standard version in other books using n-gram word compatibility.

2.3.2 Statistical approaches:

Recurrent Neural Networks [19] by long short-term memory layers they develop a recurrent neural network to predict diacritics in non-diacritized texts in the Arabic script. Their language-independent approach is only train from formed text without relying on external tools using LSTMs. They experimentally demonstrated that their model could compete with the latest approaches with access to additional resources, the network structure in [19] is shown in the Figure 12 below.

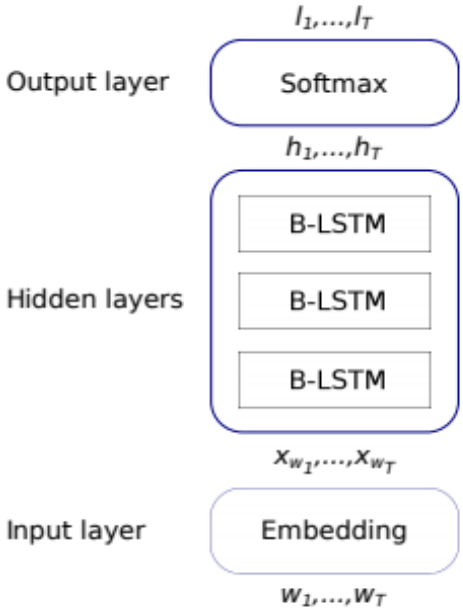

Figure 12 Network structure in [19]

2.3.3 Hybrid approaches

Typically, a mixture of rule-based methods and statistical methods in the one system. They include hybridization of rules and dictionary a replacement state-of-the-art publicly available Arabic diacritizer. Viterbi decoder [20] is use for word-level diacritization with retraction to morphological patterns and stems. Letter mining is also use in conjunction with serial labeling to diacritize named entities that need English translation. For ending-case, it uses SVMRank, including filter inference. For endings-case, they used the (SVM) Support Vector Machine based classification including grammar and morphological patterns to guess endings-case correctly.

In the paper [6] is used a hybrid LSTM/Max-Ent (Maximum Entropy) approach used to detect Syntactic diacritic mark is define as the last diacritic mark of the stem of a word. Based on the parsing tree of a given sentence, the hybrid tagger is based

on the Bi-LSTM architecture. The tagger consists of the word embedding layer is connected to the Bi-LSTM layer (bidirectional Bi-LSTM allows the integration of both past and future information. It is the forward direction and the backward direction. For improving the tagging accuracy and to help in the prediction of the diacritics marks.). The merged vectors of the forward LSTM and the backward LSTM are connected to a fully connected dense layer. Then this fully connected layer is connected to the output (SoftMax) layer, where the outputs are the diacritics marks for each word in the sequence. Bi-LSTM networks with maximum direct entropy increased with scattered direct connections. This is an increase between the input and output layers. The cross-target entropy function teaches the label to classify the diacritical marks for each word in the input sequence.

The paper [21]. Propose three deep learning models to recover Arabic text diacritics. The first model is a baseline model used to test how a simple deep learning model performs on the corpora. The second model is based on an encoder-decoder architecture, and the last model is based on the encoder part of the text-to-speech model CBHG. The baseline model consists of an embedding layer, followed by three bidirectional LSTM layers, then a fully connected layer used as a projection layer to project down the output of the last LSTM layer to the size of the diacritic's vocabulary. Lastly, a SoftMax layer is used to output a probability distribution over the output diacritics. The CBHG model composed the embedding layer first processes the input sequence. The embedding output is passed to two layers of non-linear transformation called pre-net. The pre-net output is then fed to the CBHG module, which outputs the input sequence's final representation. They added a fully connected layer to project down the CBHG module's output to the number of possible diacritics. Lastly, they used a SoftMax layer to output the probability

distribution for each diacritic. The motivate of behind the work, while building a text-to-speech system for the Arabic language, they found that the system synthesized speeches with many pronunciation errors. The primary source of these errors is the lack of diacritics in modern standard Arabic writing.

Authors in paper [5] used a pipeline of different components, a hybrid approach combining a deep learning model, rule-based corrections, and statistical corrections. In the preprocessing level, the input array and the two output arrays after the preprocessing, the input is mapped to a set of 38 numeric labels representing all the Arabic characters, in addition to 0 and the white space. The input is the characters of the text. the output is the corresponding diacritics for every character. Arabic letters can have up to 2 diacritics. one of them is Shadda; the output is represented by two vectors; one indicates the primary diacritic, and the other indicates Shadda and the Shadda as a binary vector. In the deep learning model level, the is used RNN model, composed of 2 biLSTM layers of 64 cells in each direction and parallel dense layers of sizes 8 and 64; the previous layers use hyperbolic tangent as an activation function. The first parallel layer is connected to a single perceptron that have the sigmoid activation function. the second is connect to seven perceptrons. this have SoftMax. i.e SoftMax is activation function in this state.

Bidirectional long short-term memory neural networks with conditional random fields (CRF) [4] which used for the restoration of Arabic diacritization by uses a sequence-to-sequence schema. The input is a sequence of characters that constitute the sentence, and the output consists of the corresponding diacritics for each character in that sentence. The problem is that most proposed systems are based on dictionaries and rules, language resources, or feature engineering approaches that

employ linguistic information. In contrast, few efforts have been made based on data alone. The Bi-LSTM and CRF approach be suitable for many natural language processing and text mining tasks that depend on sequence tagging, such as named entity recognition, POS tagging, and sentiment detection.

The Recurrent Neural Network (RNN) Approach needs massive data to train on and learn high-level linguistic abstractions, the authors in paper [22] prepare an external training dataset, they present three models based on it, the first is Basic Model experiment with the number of Bidirectional CuDNN Long Short-Term Memory (BiCuDNNLSTM), use two BiCuDNNLSTMs in further experiments as well as 256 hidden units per layer, using Adam optimization algorithm, two Dropout layers for each one 50% Dropout Rate, embedding layer that gives the best results is 25 D randomly, training the models for more than 50 epochs, the second is Conditional Random Field (CRF) Model or CRF classifier is used in this model instead of the Softmax layer to predict the network output, the last is Block-Normalized Gradient (BNG) Model or BNG method is applied to normalize gradients within each batch, 2 dimensions instead of 25 using t-SNE dimensionality reduction algorithm, using optimizers Adam, performing much better than other two model. There are other articles in this field that can be referred to for further reading [23-29].

2-4 Conclusion

This chapter reviews some related articles. The following challenges were raised when reviewing existing articles:

- Challenges of two Diacritics on one letter within a word.

  Most of the articles that we have seen indicate that there is a challenge, which is the difficulty in learning the models suggests that the Shadda class and the hybrid classes (i.e., Shadda + another diacritic) (i.e., Shadda with Damma, Fatha, and

Kasra) for the network compared to other classes, the reason is that there are two Diacritics on one letter within a word.

- Challenges related to the datasets.
  Unfortunately, when we read the articles, there were many problems related to the data set used for training, testing, and validation. The first problem is that most of the sources for data sets are not available (not free); most of them need to pay the money. The second problem is that there are data sets available, which are free but need much treatment or preprocessing before Use it for training, testing, and validation.

- Lack of available resources.
  According to many researchers, diacritiztion Arabic text is among the most challenging problems in Arabic NLP. In order to solve this problem, open-source resources are required.

  Although the problem's importance, it received limited attention. One of the reasons for this is the scarcity of freely available resources for this problem.

- The problem of the dependency of the diacritization of the current word on both the previous and following words.
- Calculate the Case-Ending or without Case-Ending.
  The calculation without Case-Ending excludes each word's last character from error calculation since they mainly depend on grammatical rules.

- Challenges added some suffixes to the end of the word.
  Usually, there is a diacritical mark placed at the end of a word on the last letter as known (end case). There is a problem when some suffix comes to this word,

noting that the suffixes are added or placed at the end of the word. Given these challenges, the proposed method will be present in the next chapter.

Chapter III

Proposed method

3- Proposed method

3-1 Introduction

The proposed method of this research is fully introduced in this chapter. This method is, in fact, a framework that seeks to put diacritics on a raw Arabic sentence by applying the concepts of deep learning and classification and, of course, the principles of natural language processing. A raw sentence means a sentence on which there are no diacritics. After diacritics, there are several steps of correction and post-processing to achieve a better result. All of these will be explained in detail.

3-2 Outline of the diacritics detection system

The outline of the diacritical detection system has four blocks. The first block is the system input. These inputs are raw Arabic sentences. The sentences must be specifying as an array of numbers for the system. In the second block, the diagnosis of diacritics is performed with the help of deep neural crests (bi-LSTM). This network consists of various layers. In the third block, several stages of post-processing or correction are performed. This step makes the output more accurate. Finally, the outputs, the sentences have diacritics as we shown in the Figure 13 below.

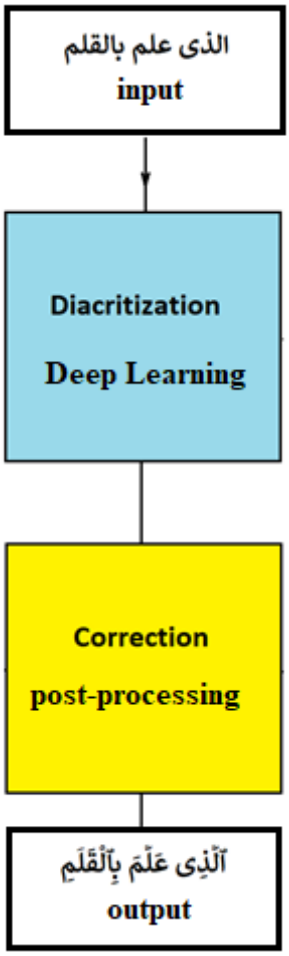

Figure 13 The outline of the diacritics detection system has four blocks.

3-3 Basic network

The outline of the diacritics detection system is shown in the Figure 14 below. According to this figure and comparison with the four blocks of the previous figure, the input, deep network, correction layer, post-processing, and output can be identified. The basics of the primary article are based on the [5].

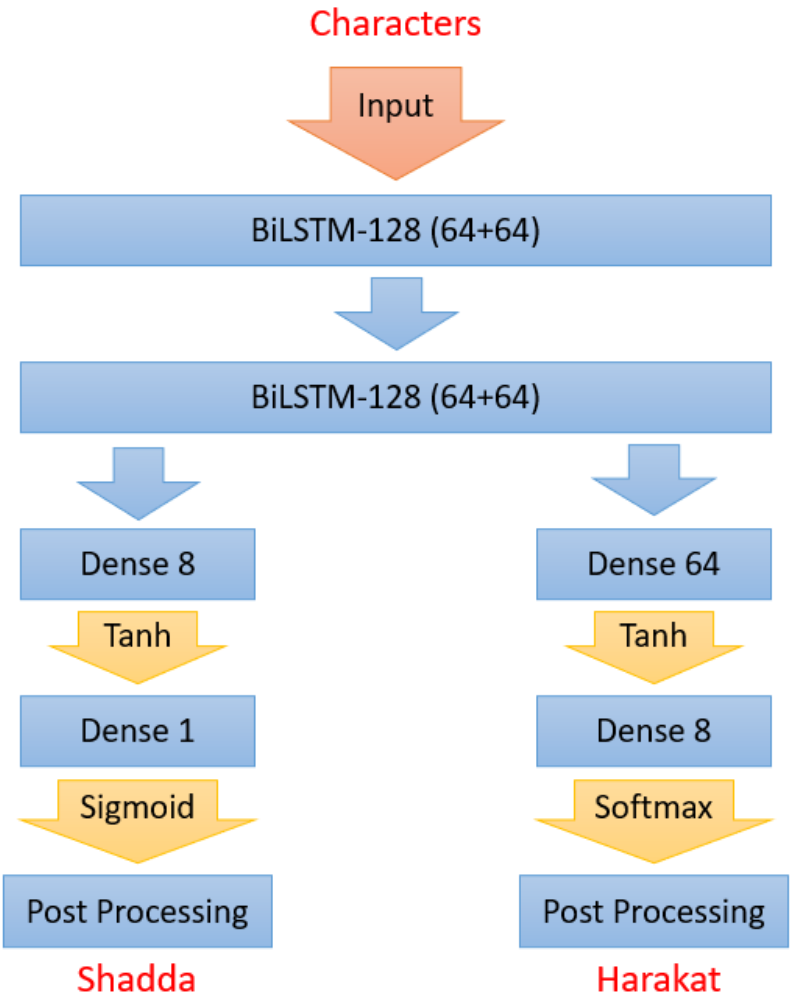

Figure 14 Basic network structure

The inputs in this system are three-dimensional. There are also two categories of outputs. The input and output structures are shown in the Figure 15 below.

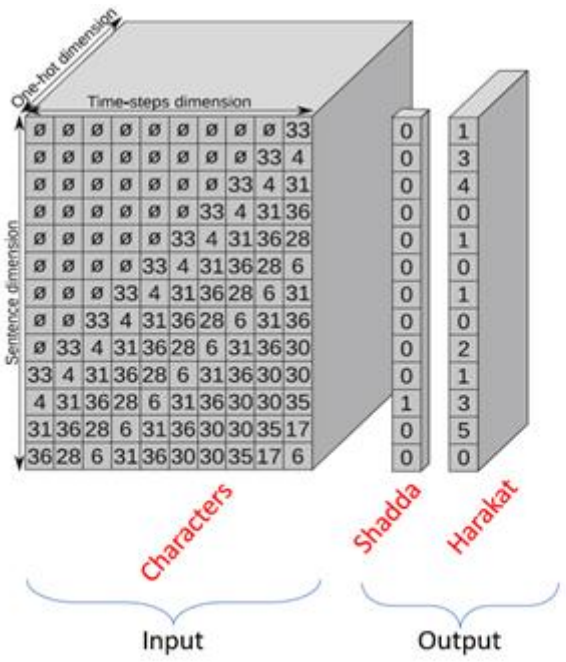

Figure 15 Structure of inputs and outputs in the base network

In inputs, the first dimension is the length of the sentence. Obviously, sentences with different lengths can be considered. The second dimension is called the time step dimension. We have considered this dimension 10. The value ten is chosen to cover the maximum lengths of a word in Arabic. By decreasing this value, the detection speed may increase, but the accuracy will decrease. The third dimension is also called the one-hot dimension. This dimension somehow represents the letter. In this dimension, 38 characters can be modeled. These characters, which include 36 letters of the Arabic language and two characters, zero and space, are named from zero to 37. In this display method, we consider a zero array to display the beginning of the sentence. In short, to access each character, it is necessary to place it in a vector. For example, the following Figure 16 shows how the two characters, 1 and 3, are

displayed. Obviously, places one and three; Equals one. Hence the desired character is selected.

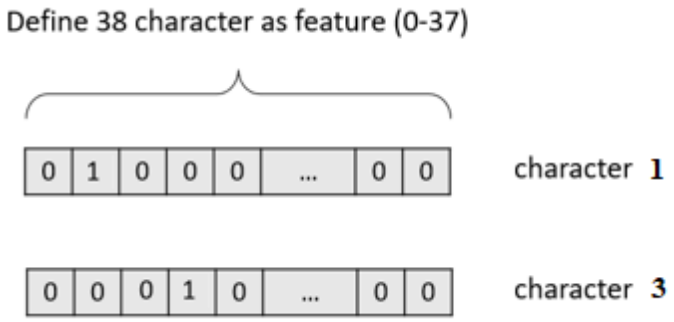

Figure 16 characters 1 and 3 are displayed.

There are also two levels in the output. In the first level, we have a vector called shadda; This vector indicates the presence or absence of shadda on a character. The second vector also represents the diacritics of the letters. The following is an example of converting a sentence to numbers (tagging), both in input and output.

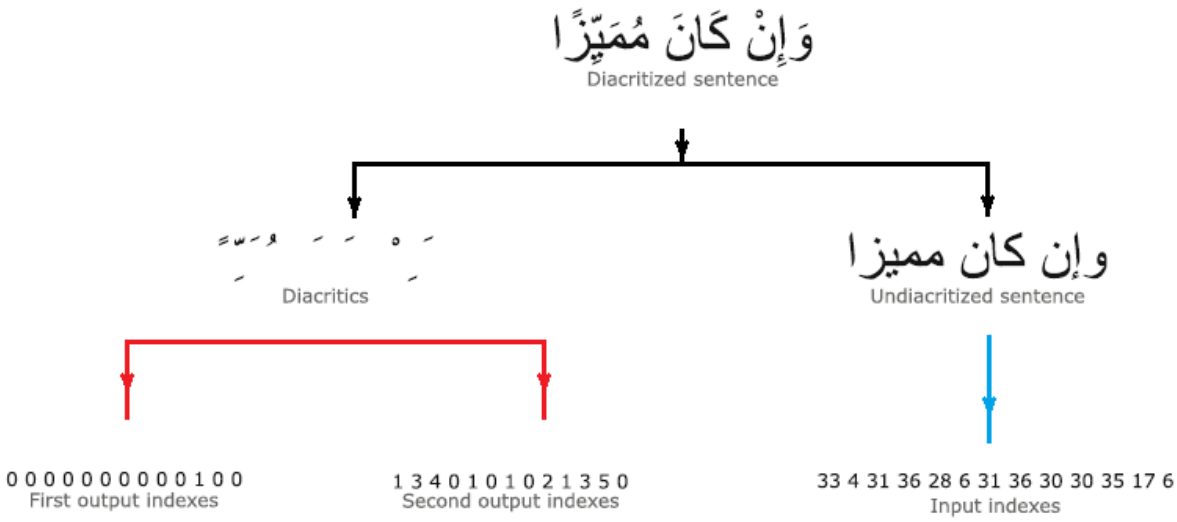

Figure 17 How to convert a sentence to numbers (tagging), both in input and output

After creating the inputs in the primary network structure, there are two Bi-LSTM layers with the dimensions shown. The detection network is then divided into two branches. In the left branch, first a dimensional layer (Dense 8), then a Tanh activator function, then another dimensional layer (Dense 1), then a Sigmoid activator function, and finally a back layer there is processing.

Similar to the left branch, in the right branch, first a dimensional layer (Dense 64), then a Tanh activator function, then another dimensional layer (Dense 8), then a Softmax activator function, and finally there is a post-processing layer. The reason for the difference in dimensions in the Dense layers as well as the activation functions in the two branches is due to the different dimensions of the outputs.

After diagnosis, there are three layers of correction. These layers are a kind of post-processing and increase the accuracy.

Level one, inputs of this level are the outputs of the previous level (activation function), and the process that occurs within this level is corrections at the letter level, for example, there are some letters do not contain diacritics, or some characters that cannot be placed a specific diacritical mark on them and after the corrections are passed Outputs to the next level.

At this level, several conditions must be considered for correction, and the results must be weighed against these conditions. For example, the first letter of a word is not accepted Shadda. Hence it must be removed. At this level, we have used grammar rules.

These editing rules can be summarized as follows:

- If the letter is 0, space, or letters (ء – أ-ى-أ-ئ-آ-ة), it does not have Shadda.
- The first letter of a word cannot have Shadda.

- It has Sukoon as a predicted primary diacritic.
- The first letter of a word cannot be Sukoon.
- The last letter of a word cannot have Fatha unless it is one of the two letters (ة-ء).
- Do not put a diacritic for the letter zero.
- Do not put a diacritic for the space character.
- Tanween should only be used for the last letter. Otherwise, delete that.
- If the current letter is (ة -ى), the previous diacritic must be Fatha.
- If the current letter is أ, the diacritic should be Kasra.
- The letter that does not move is called Sukoon, and its symbol is called Sukoon.
- Tanween is a symbol used at the end of some Arabic words.

For example, in the following sentence, the first letters have neither Sukoon nor Shadda.

إِنَّكَ لَمِنَ الْمُرْسَلِينَ عَلَى صِرَاطٍ مُسْتَقِيمٍ

Level two, inputs of this level are the outputs of the previous level, and the process that occurs within this level are corrections at the word level, for example, there are some words in the dictionary that have specific diacritics marks, so the word that content specific diacritics is close to the word will be suggested. If any of the proposed words do not match, this level will be neglected, and the previous level will be relied upon, and the outputs will be inputs for the next level.

In summary, Language modeling is a significant part of many of the tasks associated with natural language processing. In this study, we also used it in the second step of correction. Language modeling is simply the act of predicting the next word in a given sequence. Take the phrase “I'm writing a…” as an example. The next word that can follow is “letter”, “sentence” or “blog post” and …. In other words, for the given words x (1), x (2), and… x (t), the language models calculate the probabilistic distribution of the following word (t + 1) x.

The most basic language model is the n-gram model. N-gram is one of the most common methods of statistical language modeling. An n-gram is a set of n words in a row.

For example, for the phrase "I am writing a book" we have the following n-grams:

- unigram: "I", "am", "writing", "a", " book "
- bigram: "I am", "am writing", "writing a", "a book"
- trigram: "I am writing ", "am writing a", "writing a book"
- 4-gram: "I am writing a", "am writing a book"
- 5-gram: "I am writing a book"

The main idea behind n-gram language modeling is to collect statistical information about the frequency of different n-grams (how many times each n-grams is repeated) to use this information to predict the next word. However, n-gram language models face the problem of sparsity, in which we do not see enough data in a body of text (corpus) to be able to model the language correctly (accurately) (especially when n increases). Therefore, at this level, we must select the correction at the word level and based on the language model and dictionary. For example, using word correction

(trigram), we choose the word صِرَاطٍ as the correct word. So, the third sentence is correct.

إِنَّكَ لَمِنَ الْمُرْسَلِينَ عَلَى صِّرَاطٍ مُسْتَقِيمٍ

إِنَّكَ لَمِنَ الْمُرْسَلِينَ عَلَى صَرَاطٍ مُسْتَقِيمٍ

إِنَّكَ لَمِنَ الْمُرْسَلِينَ عَلَى صِرَاطٍ مُسْتَقِيمٍ

It should be noted that editing at this level is dictionary-based. Therefore, the better the dictionary content, the better the correction result. Also, if no match is made, the same results as the previous level will be accepted. After these two levels, a third semantic level can be added to the correction process. Obviously, its presence will lead to increased accuracy.

3-4 CRF based network

The basic design reviewed in the previous section can be improved by applying some ideas. One of these ideas is to add a CRF layer to the network. The network, in this case, will be as follows. In this network, the dimensions of the Dense layers have changed. The reason for this will be explained below. Another critical point is to remove the two layers of Bi-LSTM and turn them into two completely separate surfaces. With a simple comparison between the primary network structure and the CRF-based structure, these changes are simply noticeable.

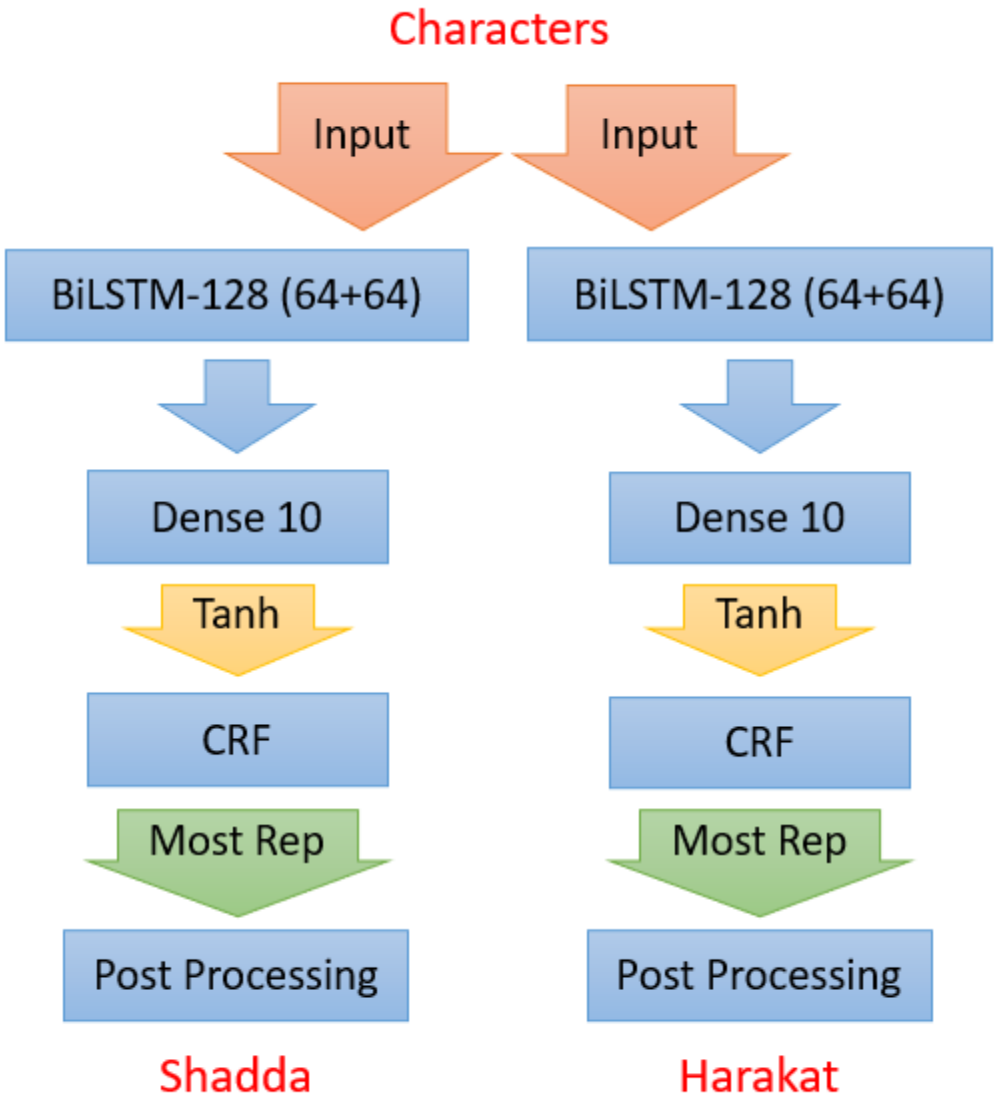

Figure 18 CRF-based network structure

CRF calculates the probability of an output occurring according to the system input and obtains the output as a probability. This design increases accuracy and reduces error. Obviously, due to the CRF structure, the corresponding input is added to the output dimensions. Network inputs and outputs are shown in the Figure 19 below. In the created network, we have to consider the maximum iteration to calculate the

output. For example, in the Figure 19 below, for character 33, 9 to one and one two are created, which ninety percent of the output is the number one, we choose one as the output. Therefore, since the CRF level output is ten, the proposed network dimensions such as Dense and the type of activation function have changed. Also, the separation of branches in a CRF-based network is that the CRF library cannot be Implementedin one place. Therefore, the most critcal points in the network and CRF-based algorithm are:

• CRF uses the probability of occurrence of each diacritic depends on the input character.

• deciding on ten outputs reduce the probability of error.

• In CRF network, for each letter, there are ten predictions.

• The number with the most repetition is chosen as output.

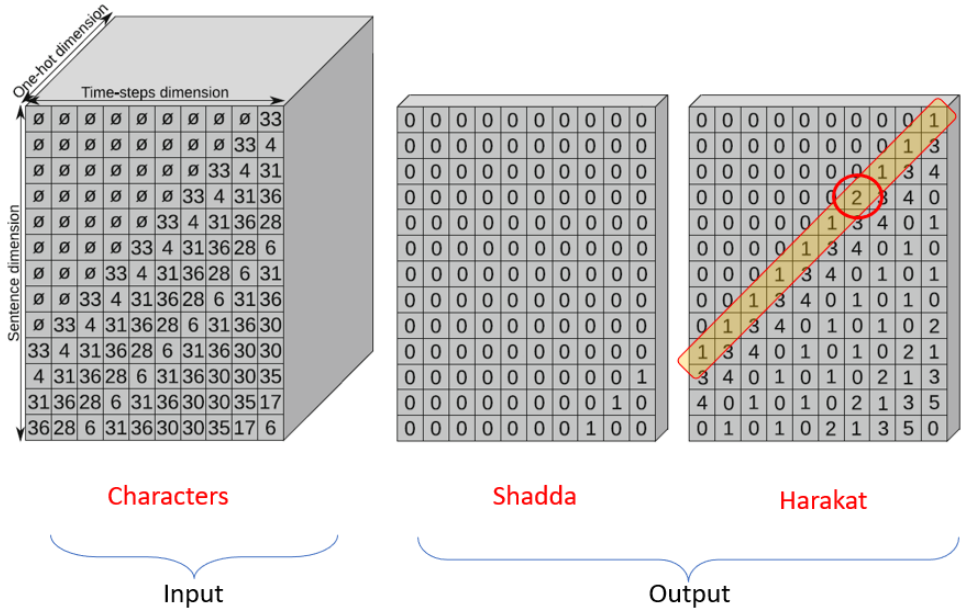

Figure 19 Inputs and outputs in CRF-based network

3-5 Network based on two-level inputs (character-word)

The following modification in the base algorithm is the correction of the structure of the inputs, in this case, instead of character inputs, we use words as inputs. The network structure in this case is as follows. The dimensions of the Dense layer and the type of activator function are apparent in both output branches.

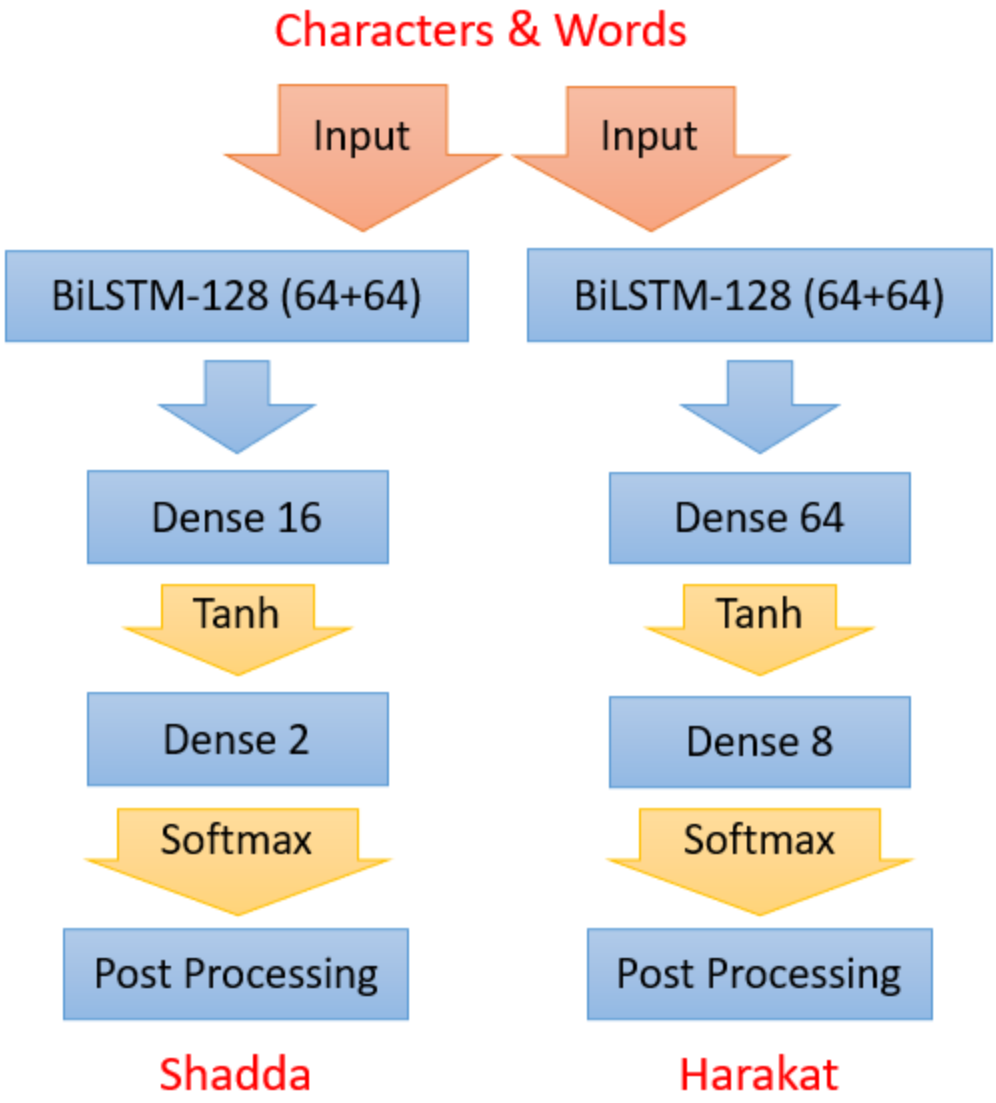

Figure 20 Network structure based on two-level inputs (character-word)

The inputs and outlets in this network are as follows. It should be noted that the presence of 38 zeros and ones at the network input does not help the algorithm much. The reason is apparent. This is because the characters alone do not have valuable information. Now in the proposed design, in addition to characters, we create word and dictionary entries. In this case, we will have two levels at the entrance, which are clearly shown in the Figure 21. To create a word level in the input, we must remove all extracted words and duplicate words. With this design, about 94,000 words have been created. Obviously, if the inputs are considered this way, we will have a 19-fold increase in speed. Of course, minor changes in the definition of inputs seem necessary. Here we use the number 38 to start the sentence and the number 39 to end so that all modes can be defined. Therefore, the most important points in the network and algorithm based on two-level inputs (character-word) are:

- Change input dimensions and data.
- D (38 * TimeStep * SentenceLength) => D (2 * TimeStep * SentenceLength).
- Add a word dictionary layer.
- Also, replace Ø by number 38 as “start character” and extend the sentence by number 39 as “end character”.
- Word dictionary consists of 94000 words without diacritics.
- This procedure reduces the input layer of the network and speed up the training process.
- Also, some diacritic depends on word position in the sentence.

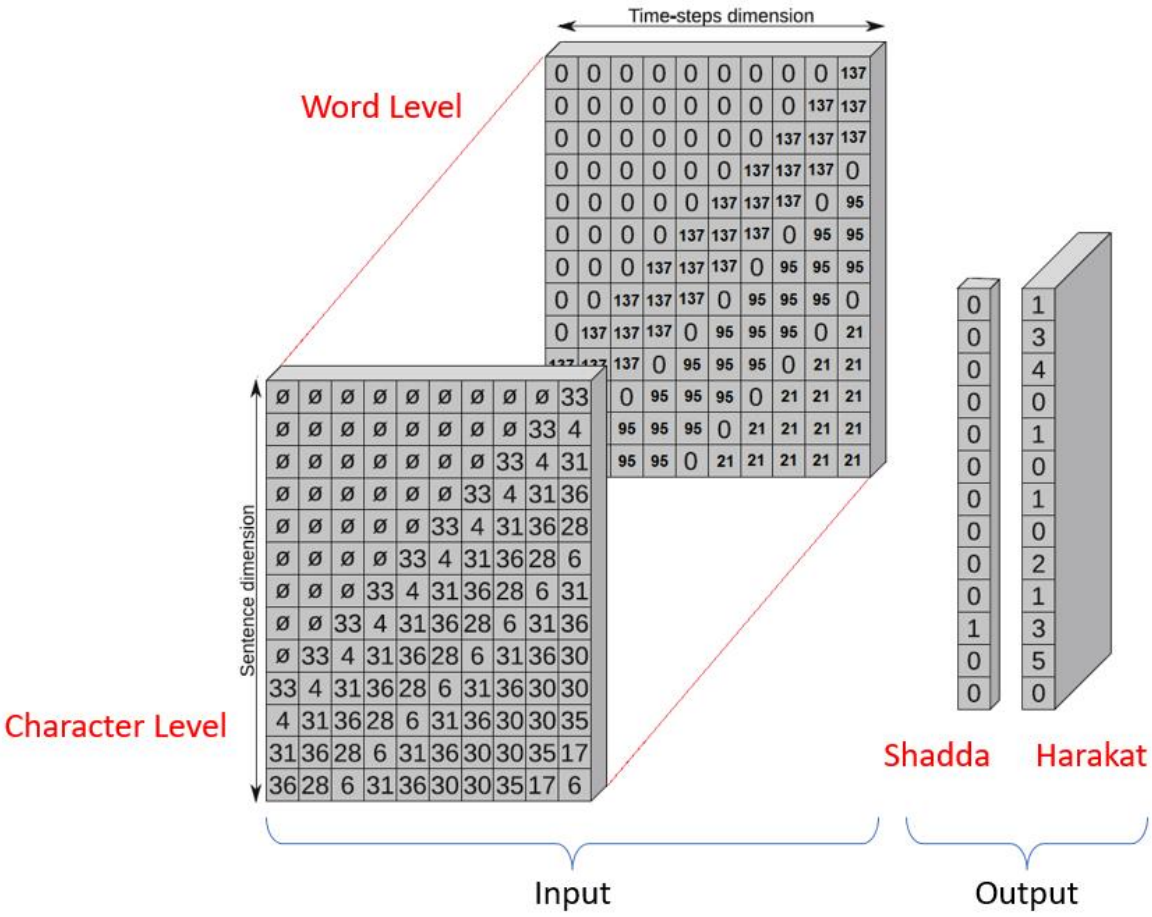

Figure 21 Structure of inputs and outputs in a network based on two-level inputs (character-word)

3-6 Proposed network (CRF-based network with two-level inputs)

So far, a primary network has been introduced. Then two improved networks (algorithms one and two) were presented. The first algorithm was a CRF-based network. The second algorithm was a network with two-level inputs. The combination of these two systems creates the proposed. Obviously, each of these algorithms has advantages that combine to create a powerful diacritics detection

system. Network shape and inputs and outputs in the proposed network (CRF-based network with two-level inputs) are shown in the Figures 22 and 23 below.

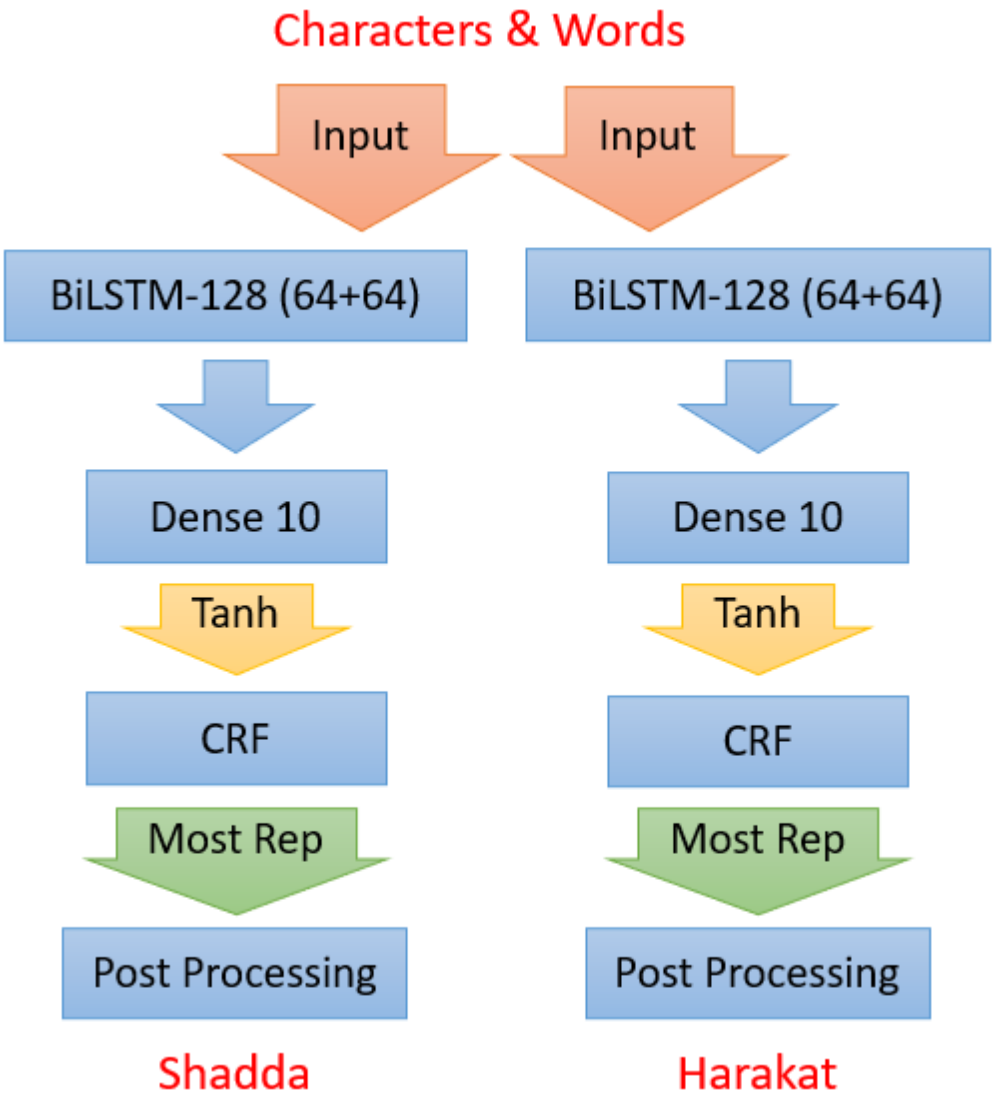

Figure 22 Proposed network structure (CRF-based network with two-level inputs)

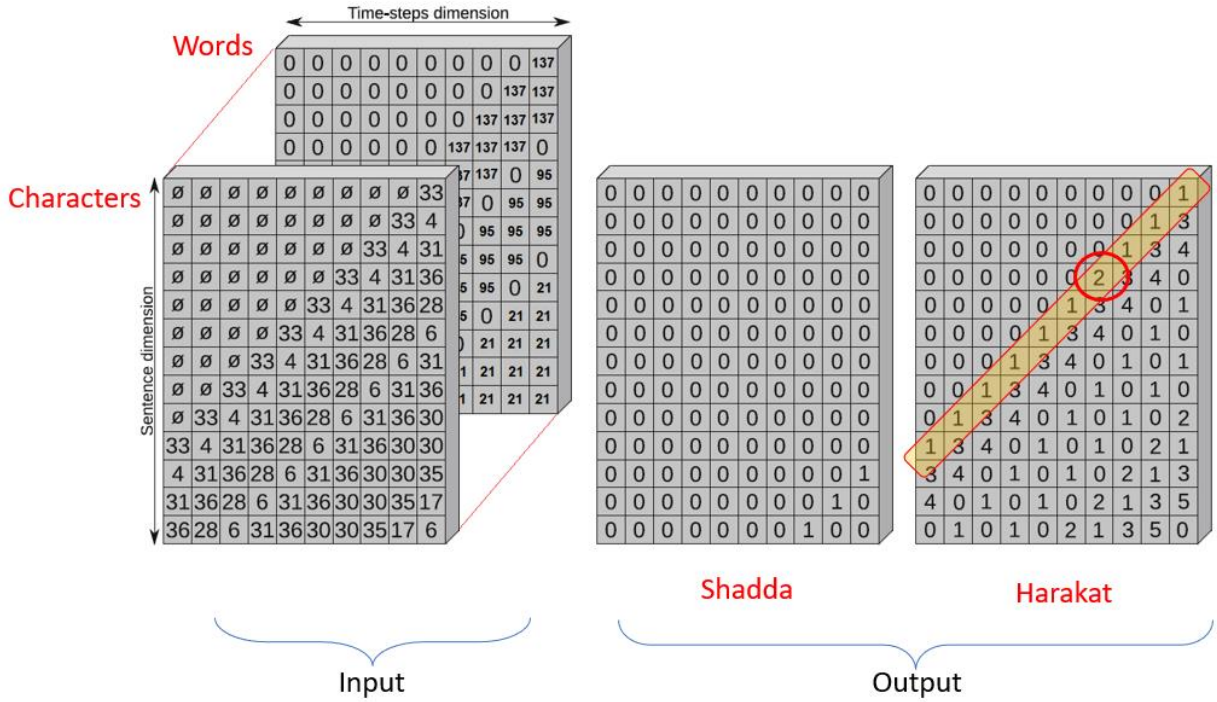

Figure 23 Inputs and outputs in the proposed network (CRF-based network with two-level inputs)

3-7 Conclusion

The method proposed in this research; Created with an improving design. In this way, a primary network was first fully described (Figure A). Then, using a CRF layer, a CRF-based network was created that had a higher accuracy than the base network (Figure B). In the third step, a network with two-level inputs was created (Figure C), and finally, from the combination of the last two networks, the proposed final network was introduced (Figure D). These networks will be Implementedin the next chapter.

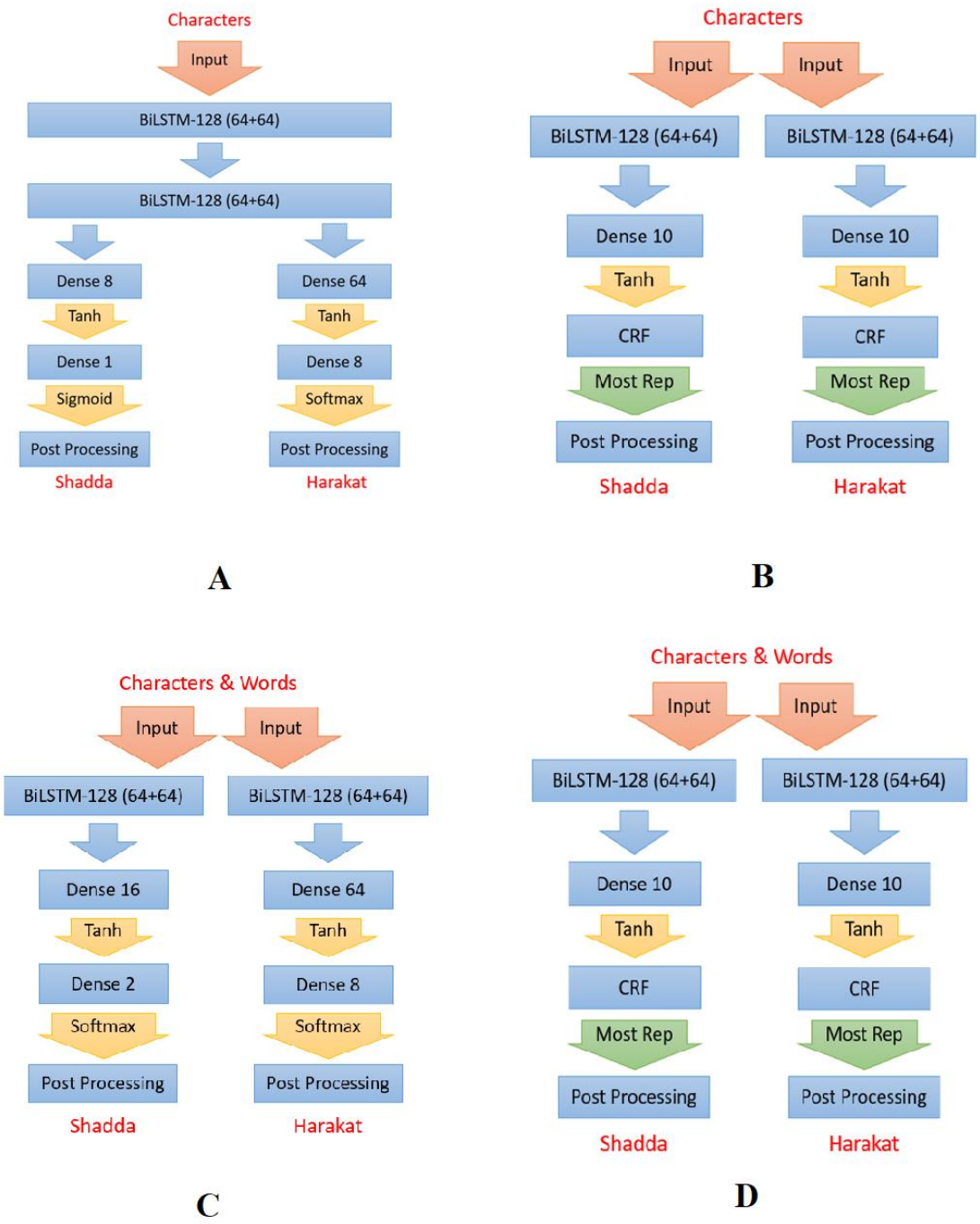

Figure 24 View of the four networks introduced

Chapter IV

Implementation of the proposed method

4- Implementation of the proposed method

4-1 Introduction

In this chapter, the proposed method is Implemented. The data set used introduced at the beginning of this chapter. Then the evaluation criteria of the method are introduced. The following sections provide results and comparisons for four networks (basic, CRF-based, dictionary-based, and hybrid).

4-2 Dataset

In the first chapter, it was stated that little research had been done on the processing of natural language in the Arabic field. Hence, it is clear that there is little data set. One of the future researches could be the production and labeling of a suitable data set for research. We are in the process of selecting a Tashkeela data set [30] to test our work; This dataset is extracted from religious books. The original dataset has over 75.6 million words, where over 67.2 million are diacritized Arabic words. These dimensions are suitable for research work. The table 5 below provides information from this dataset.

Table 5 Statistical information from the data set

	All tokens	Only numbers	Only Arabic words	Undiacritized words	diacritized words
Train phase	31774000	80415	27657285	351816	626696
Val phase	1760395	4460	1532628	100800	152751
Test phase	1766840	4395	1537875	101262	153315

In using the data set, we must pay attention to an important issue (preprocessing). In the process of Arabization, numbers and spaces must be eliminated. This is a kind of normalization. The purpose of this tool is to clean and sort the text and unify the characters by replacing the standard characters in the input text. In fact, pre-processing should be done before standardizing texts to standardize letters and spaces. In fact, at this stage, all the letters of the text should be standardized by replacing them with their standard equivalent. In the processing of Arabic calligraphy, there is always the problem of using equivalent Arabic characters in writing a number of professions; Among them, we can mention the letters "ک", "ی" and Hamzeh. In the first step, the problems related to these letters should be solved by unifying them.

In addition, the modification of the semicolon and distance characters in its various applications, as well as the elimination of the Arabic characters, intensification,

hyphenation, and "-" used to stretch the sticky characters, and similar cases for the unification of texts, are necessary measures before starting. Is word processing.

4-3 Explanations about implementation

We have implemented the proposed approach in the Google Collaboratory environment, as this environment is considered one of the best environments because the processing and run take place in the Google servers., This environment also contains most of the necessary libraries, including the TensorFlow library, as Google is the owner of these libraries we implemented with Python 3.7 using the Keras library with a TensorFlow version 2.7.1, and each processed sentence of text is considered a single batch of data when fed into the Deep learning approach. The training data was transformed into NumPy arrays for both input and output. We used the optimizer for adjusting the model weights by ADAM.

4-4 Evaluation criteria

There are several criteria for evaluating the proposed method, which we will introduce in the following. The first criterion is called Diacritization Error Rate (DER). As it turns out, this criterion gives the error rate for a letter. Equation one shows this.

$$DER = \frac{\text{number of incorrectly diacritized characters}}{\text{number of all characters}}$$

The second criterion is called Word Error Rate (WER). As it turns out, this criterion gives the error rate for a word. Equation one shows this.

$$WER = \frac{\text{number of incorrectly diacritized words}}{\text{number of all words}}$$

Of course, we turn these two criteria into two separate criteria. In DER1 and WER1 criteria, we consider all letters. However, in DER2 and WER2 criteria, we do not consider the last letter. According to the definitions, two general conclusions can be drawn.

The first result is that the DER error is always less than the WER error. This result is clear. If the diacritic of a letter of a word is wrong, the whole word is wrong. For example, if there is a sentence with three words and ten letters and the diacritic of one letter of this sentence is misdiagnosed, then we will have a ten percent error in DER. In this case, however, 33% error is generated for WER. That is, one of the words is also considered an error.

The second result is that the DER1-WER1 criterion is more honest than the DER2-WER2 criterion and therefore has more error. Because all letters must be weighed. But to compare with previous work, we have considered both criteria.

For evaluation, the result tables have been divided into two parts. In the case of Include no-diacritic letters, letters that do not have diacritic are also considered. These letters are numbers, distances, and so on. In the second case, “Exclude no-diacritic letters”; these letters are not considered. Obviously, in the case of “Include no-diacritic” letters the accuracy will be higher because these letters, were without diacritic from the beginning.

4-5 Base network results

The results of the primary system are presented in the Table 6 below. As we can see, the DER error is always less than the WER error. It is also clear that the DER1-

WER1 criterion is more honest than the DER2-WER2 criterion and therefore has more error. In the case of “Include no-diacritic” letters, letters that do not have diacritic are also considered. These letters are numbers, distances, and so on. In the second case, “Exclude no-diacritic” letters; these letters are not considered. Obviously, in the case of “Include no-diacritic” letters, the accuracy will be higher because these letters were without diacritic from the beginning. The basic system is without any improvements. We expect the results to continue to improve as this network improves.

Table 6 Base Network Results

Exclude no-diacritic letters				**Include no-diacritic letters**			
DER1	WER1	DER2	WER2	DER1	WER1	DER2	WER2
3.39%	9.94%	2.61%	5.83%	3.34%	7.98%	2.43%	3.98%

4-6 CRF based network results

The results of the basic design presented in the previous Table 6 can be improved by applying some ideas. One of these ideas is to add a CRF layer to the network. The CRF calculates the probability of an output occurring according to the system input and obtains the output as a probability. This design increases accuracy and reduces error. The results in the Table 7 below illustrate this well. For example, in this case, the DER1 value has improved by a factor of 1.05.

Table 7 CRF-based network results

	Exclude no-diacritic letters				Include no-diacritic letters			
	DER1	WER1	DER2	WER2	DER1	WER1	DER2	WER2
Base article	3.39%	9.94%	2.61%	5.83%	3.34%	7.98%	2.43%	3.98%
Algorithm 1	3.22%	8.36%	2.31%	5.13%	3.12%	7.14%	2.12%	3.73%

4-7 Network results based on two-tier inputs (character-word)

The following modification in the base algorithm is the correction of the structure of the inputs, in this case, instead of character inputs, we use words as inputs. The results in this case are presented in the Table 8 below. Here, the CRF layer has been removed to understand the results better. This design also increases accuracy and reduces errors. The results in the Table 8 below illustrate this well. In this case, for example, the DER1 value is improved by a factor of 1.33 compared to the base network.

Table 8 Network results based on two-level inputs (character-word)

	Exclude no-diacritic letters				Include no-diacritic letters			
	DER1	WER1	DER2	WER2	DER1	WER1	DER2	WER2
Base article	3.39%	9.94%	2.61%	5.83%	3.34%	7.98%	2.43%	3.98%
Algorithm 1	3.22%	8.36%	2.31%	5.13%	3.12%	7.14%	2.12%	3.73%
Algorithm 2	2.54%	7.85%	2.16%	4.59%	2.75%	6.78%	1.98%	3.52%

4-8 Proposed network results (CRF-based network with two-level inputs)

In the results sections, two improved networks (algorithms one and two) are presented and compared with the base network. The first algorithm was a CRF-based network. The second algorithm was a network with two-level inputs. The combination of these two systems creates the proposed. Obviously, each of these algorithms has advantages that combine to create a powerful diacritics detection system. The Table 9 below shows the results in this case. In this case, the DER1 rate is improved by a factor of 1.86 compared to the base network. This number indicates a significant improvement.

Table 9 Results of the proposed network (CRF-based network with two-level inputs)

	Exclude no-diacritic letters				Include no-diacritic letters			
	DER1	WER1	DER2	WER2	DER1	WER1	DER2	WER2
Base article	3.39%	9.94%	2.61%	5.83%	3.34%	7.98%	2.43%	3.98%
Algorithm 1	3.22%	8.36%	2.31%	5.13%	3.12%	7.14%	2.12%	3.73%
Algorithm 2	2.54%	7.85%	2.16%	4.59%	2.75%	6.78%	1.98%	3.52%
Proposed	1.82%	5.32%	1.48%	3.09%	2.05%	5.04%	1.72%	3.07%

To better understand the extent of improvement, the following diagrams are presented visually for the DER1-WER1 criteria in both Include no-diacritic letters

and Exclude no-diacritic letters. These diagrams show well how proposed the (combined) method has been.

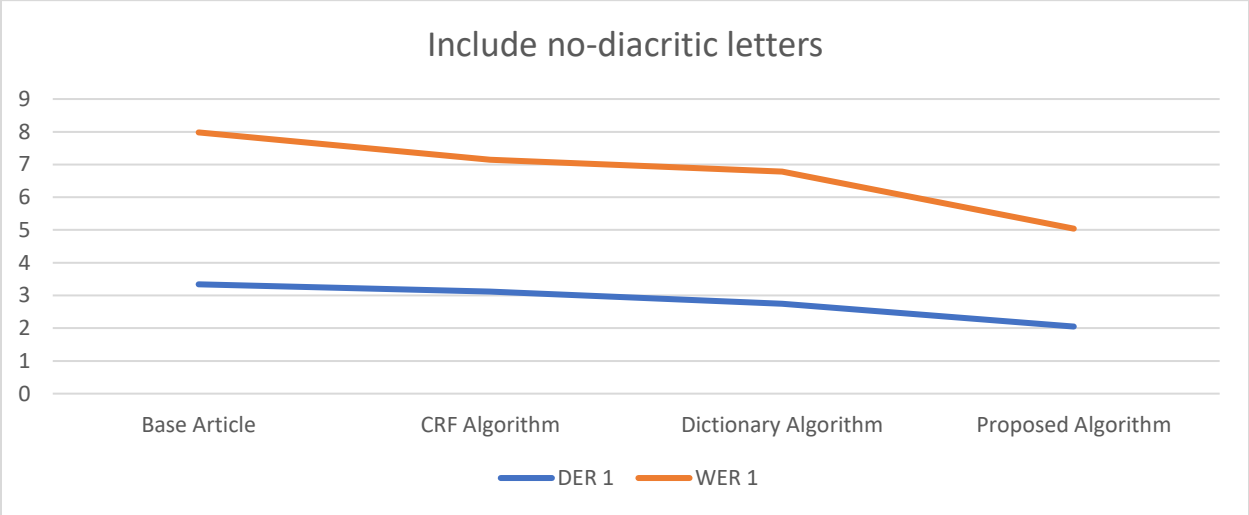

Figure 25 Results chart for the two criteria DER1-WER1 in Include no-diacritic letters

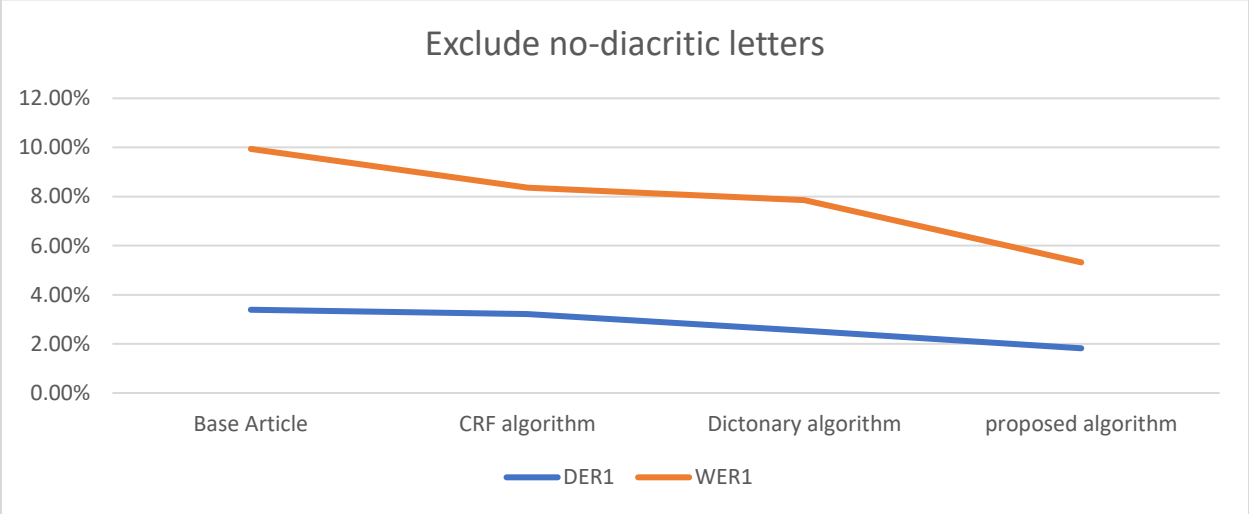

Figure 26 Graph of results for two criteria DER1-WER1 in Exclude no-diacritic letters mode

The results of our work can be the best results because the rest of the systems do not make any corrections to the output of the deep learning model, while the results of

our work include a series of corrections at multiple levels that correct many of the output errors. In addition, the semantic level has been added, which is the last level that gives the exact meaning of the word.

4-9 Conclusion

In this chapter, the proposed method is Implemented. At the beginning of this chapter, the data set used was introduced. This data set was suitable and useful for this research because it considered different modes. Then the method evaluation criteria were introduced. In the following sections, results and comparisons were presented for four networks (basic, CRF-based, dictionary-based, and hybrid). The results of tables and graphs showed that the proposed method has high capabilities and has very little error. Therefore, it can be used with high confidence to determine the diacritic letters.

Chapter V

Summary and Conclusion

5- Summary and Conclusion

Arabic is one of the most popular languages in the world. Although the choice of Arabic as the language of the Qur'an has made this language honorable, However, the perfection of the Arabic language is not due to the completeness of the Qur'an, but the unique features that exist only in this language have made the Arabic language the most complete or one It is one of the most complete and wealthiest languages and perhaps one of the wisdom that has caused (in addition to other wisdom) that God revealed His verses and enlightening words to humanity in the form of this language is the existence of these unique characteristics. Here are some of them:

1- It is one of the most exhaustive languages in words and terms.

2- It is diverse in interpretation and has different interpretations for meaning or meanings close to each other.

3- It has a stable language system.

4- It has the capability of Arabs. This feature, which is not present in other languages, is the change of the last state of the word, according to which the role of the word in the sentence is determined, and thus both the reader and the speaker are protected from speech errors.

Diacritic (French: Diacritique) means the use of glyphs or special symbols placed on letters to determine how a letter is pronounced, or in other words, how lips move.

5- It has the appropriate letters with meanings. It has been argued by some past and later lexicographers that there is a particular fit between a series of words and their meanings

6- Derivation: means the production of multiple words and the achievement of multiple meanings from the origin of a word (source).

7- Other characteristics such as accuracy and delicacy in meaning, the multiplicity of synonyms and many meanings with a single word, multiple words for a single meaning, etc.

These features, along with another essential feature and a large number of contacts in this language, have attracted a lot of attention. However, little research has been done on natural language processing in this area. In this research, a network based on Bi-LSTM deep learning, and CRF with two-level inputs is presented. The detection results of this algorithm are then improved by making some post-processing and corrections. The Implementation results show that the algorithm proposed in this research can perform up to twice as much as the best methods in this field and detect the diacritics in the Arabic phrase.

Future research in this area could be based on more corrections and correctional layers. The use of stronger deep learning structures or the combination of several network structures to achieve a stronger structure can also be explored. Deep networks with new structures can also be studied [31, 32]. In addition to all this; Creative and new ideas, can also lead to better results. By creative ideas we mean creating structures that are completely different from the proposed structure.