

# A Novel Key Exchange Algorithm

TajElsir Hassan Suliman Hamad

Program of Banking Information Systems,  
Sudan Academy for Banking and Financial Sciences, Sudan  
tagsir@yahoo.com

**Abstract:** *One of the most important tasks of the information security issues is how to exchange the secret key between the communication parties securely, it is possible to create a secure cipher and send it to the recipient without one to one meeting. In this paper we have proposed a simple and novel key exchange algorithm that can easily generate the shared-key between the communication parties, hereafter is called (Tajo), however, the generated key is used for distributing the secret key and is not used for encryption. Analytically, we have proved that Tajo outperformed the relevant algorithms in terms of simplicity of generating the shared key and defeating the brute force attack..*

**Keywords:** *Tajo, shared key, Vulnerabilities, Algorithm.*

Received: June 9, 2016 | Revised: June 10, 2016 | Accepted: January 21, 2017

## 1. Introduction

The development of digital computers made possible much more secure ciphers. Furthermore, computers were able to encrypt any type of data represented in any binary format, unlike classical ciphers which only encrypted written language texts. Computer use has thus supplanted linguistic cryptography, both for cipher design and cryptanalysis.

But usage of computers brought out new difficulties. The important part of encrypting and decrypting a cipher is knowing a key – a parameter that determines the functional output of a cipher. Without a key, the algorithm would have no result. A key specifies the transformation of plaintext into cipher text, or vice versa during decryption. Now it was possible to create a secure cipher and send it to the recipient without one to one meeting, without using an extra-safe channel. However, there still was no way to safely send the key – if it got to the eavesdroppers hands, data was easily decrypted.

Diffie and Hellman published the first public-key algorithm known as a “Diffie-Hellman key exchange” in 1976, making exchange of the keys real and secure.[6], thereafter, new key exchange algorithms has been published such as Elliptic curve Diffie-Hellman(ECDH) which is a relatively new key agreement algorithm based on Diffie-Hellman but using the elliptic-curve Cryptography[11].

Most of the key exchange algorithms used very hard computational operations to generate the shared key,

which is used for distributing the secret key in symmetric encryption algorithms, however, they are prone to many types of attacks such as Man-In-Middle(MIM) attack, and must be enhanced to defeat it.

This paper proposed a simple and novel key exchange algorithm, hereafter is called (Tajo) and it depends on two prime numbers and one operator that formulate a mathematical function to generate the shared key.

The rest of the paper is organized as follows: In Sections II algorithm overview is presented, in section III changing the shared key is presented. Section IV is devoted to the proposed algorithm versus related algorithms, the proposed algorithm vulnerabilities are demonstrated in IV, the paper is concluded in V.

## 2. Algorithm Overview

Our proposed algorithm, “Tajo” is not an encryption mechanism that we do not typically use it to encrypt data. Instead, it is algorithm to securely exchange the keys that encrypt data. Tajo accomplishes this secure exchange by creating a “shared secret” between two devices. The shared secret then encrypts the symmetric key for secure transmittal. The symmetric key provides for secure delivery of the data itself.

The algorithm has two parameters  $p$  and  $q$  and one operator ( $op$ ). Both  $p$  and  $q$  are prime numbers,  $p$  is a private and constant while  $q$  is public and

changeable,  $op$  is modulus operator ( $\%$ ), and the two parties must agree on  $p$  and  $op$ , and must be kept away. To make a more simple description we shall imagine two people – Alice and Bob who want to securely exchange data.

Suppose Alice and Bob want to agree on a shared secret key using Tajo key agreement algorithm. They proceed as follows: Alice and Bob agree on  $p$  and operator ( $op$ ). (This is usually done long before the rest of the algorithm) First, Alice generates  $k = p \text{ op } q$ ,  $op$  is  $\text{mod}$ , Alice sends  $q$  which has been used in generating  $k$  to Bob, and Bob performs the same operation by using the received number  $q$  to generate  $k$  ( $k = p \text{ op } q$ ). Finally, Alice computes  $k = p \text{ op } q$  and Bob computes  $k = p \text{ op } q$ . Since  $k = k$ , Alice and Bob now have a shared secret key  $k$ . The important point is that the two values generated are identical. They are the “Shared Secret” that can encrypt information between systems [see illustration 1].

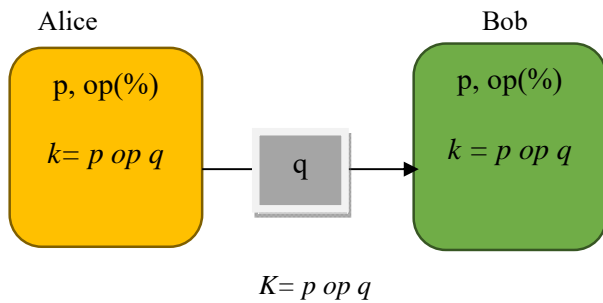


Illustration 1: Tajo key exchange

Here is an example of the algorithm, with non-secret parameter in **green**, and secret values in **Boldface red**:

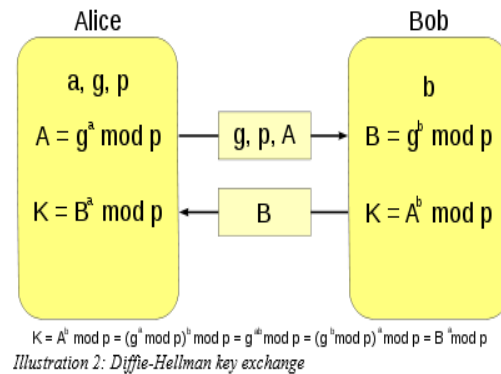
1. Alice and Bob agree to use prime number  $p = \mathbf{23}$  and operator ( $op$ ) is ( $\%$ ).
2. Alice chooses  $q = 19$  and computes  $k = p \text{ op } q$   
 $\mathbf{23 \% 19 = 4}$ .
3. Alice sent  $q$  to Bob.
4. Bob receives  $q$  and computes  $k = p \text{ op } q$ 
  - $\mathbf{23 \% 19 = 4}$  ( $k$  is shared key)

At this point, Tajo operation could be considered complete. This shared key ( $k$ ) can be used to transmit the symmetric key in any symmetric algorithms such as Triple DES, AES and IPsec. Which side of the communication actually generates and transmits the symmetric key varies. However, it is most common for the initiator of the communication to be the one that transmits the key. Once secure exchange of the symmetric key is complete, data encryption and secure communication can occur. Changing the symmetric key for increased security is simple at this point. The longer a symmetric key is in use, the easier it is to perform a successful cryptanalytic attack against it. Therefore, changing keys frequently is important. Both sides of the communication still have the shared secret and it can be used to encrypt future keys at any time and any frequency desired. In some IPsec

implementations for example, it is not uncommon for a new symmetric Data Encryption Key to be generated and shared every 60 seconds. [9]

## 2.1 Changing Shared Key

The shared key ( $k$ ) can be changed frequently (after  $t$  time) by the initiator or receiver, simply by changing the value of  $q$  and re-calculating  $k$  and sends the value of chosen  $q$  that used in the generation process to the corresponding party to perform the same operation ( $k = p \text{ op } q$ ). If this process is performed with each new shared key generation, then (Tajo) can be classified as One Time Pad algorithm, and can resist against server attacks (i.e. man in the middle and brute force).



## 3.2 Proposed Pre-Processing

In the pre-processing phase, we examined a wide variety of Arabic linguistic techniques, which we implemented using Python programming language. Before working on the tweet, we first tokenized it into lists of words, numbers, and symbols which are called tokens by using NLTK tokenization. Then we removed stop words such as prepositions, articles, and pro-nouns, e.g. (أنا, من, هذا, في, etc.). After that, we removed punctuation and blank spaces. In addition, we removed the diacritics from both the tweets dataset and the Arabic sentiment lexicon to reduce confusion when searching for words in the lexicon. Among the social media platforms, Twitter users' posts tend to be the worst in terms of spelling and grammar. In fact, one in 150 English words posted on Twitter is misspelled [2]. To overcome this problem, we used an Arabic tool called Ghalatawi: Arabic Auto-Correct, which is supported in Python. To reduce the neutral results, we applied the proper noun removal feature, because most proper nouns have a neutral score in Arabic SentiWordNet (ArSenL). However, we used a list of 14,000 Arabic proper nouns including most Arabic family names, cities, etc. Also, the pre-processing phase contains the following tasks:

(1) **Arabic Stemmer:** As we know, the words in the Arabic language are derived from sets of roots that

describe a basic idea with added affixes, which change the word's pronunciation [1]. So, in our system we used two main types of stemming processes that apply to the Arabic language: a Root extraction stemmer, such as The Information Science Research Institute (ISRI) Stemmer; the second type is Light stemmer.

|                |   |
|----------------|---|
| Row Tweet:     | لون الأرض في احضان بحارها التي ترفع كفيها نحو السماء تسأل الله النصر والسلام و النصر يحضنها و سما ع صدره العن لأنها الوطن<br>The color of the ground in the embrace of seas that raise her hands toward the sky ask God's for victory and peace, victory hug like insignia on his chest because she is the homeland |
| ISRI Stemmer:  | لون ارض احض بحر رقع كفي سمء سأل ال نصر سلم نصر حضن سما ع صدر ممن لأن وطن  |
| Light Stemmer: | رض حض حار رقع ف سماء سأل ل نصر سلام نصر حفض ام ع صدر وطن  |

Figure 2. Arabic stemmer example.

Using Arabic stemmers can be quite problematic, since there are no accurate Arabic stemmer tools we can rely on to use in our experiment and support python program language. Therefore, we carried out some additional steps to increase the level of accuracy. First the system will search for a word without using the stem in ArSenL; if it cannot find the sentiment of the word, then it will stem the word by ISRI stemmer; if a word cannot be stemmed, then it will go to the second approach, the Light stemmer, instead of leaving the word unchanged.

(2) **Part of Speech (POS):** Arabic has a scientific grammatical system, where an Arabic word can be classified into three major parts of speech: Nominal (Ism - اسم), Verb (Fia'al- فعل) or Particle (Harf - حرف) [16]. We used POS tagging to choose accurate synsets, because the SentiWordNet lexicon relied on POS for classification. There is only one Arabic POS tagging tool supported by Python; it is NLP Sanford POS tagging and it is less accurate than MADAMIRA<sup>1</sup> which, unfortunately, is not supported by Python. The NLP Sanford POS tool reads Arabic text and assigns parts of speech to each word using Penn Treebank Tags. However, before operating this tool, we had to download it on a machine that was equipped with Java.

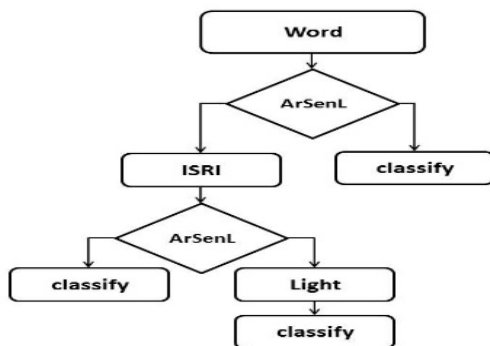


Figure 3. Flow chart of proposed Arabic stemmer technique.

(3) **Handling Negation:** Negation plays a fundamental role in sentiment analysis by affecting the text polarity. Furthermore, negation handling will be more difficult in the implicit form, where the sentence carries a negative sentiment without the use of negative words, such as sarcastic sentences, e.g. ("This is the first and last time ", "آخر مرة", "هذي أول و آخر مرة"). For the purpose of the study, we collected a list of the most frequently used negation words in the Arabic language; when the classifier found such negation words, it would give the word in the list a score of (negative = 1). The goal of this method of handling negations is to increase the total of the negative score in a sentence.

### 3.3 Arabic Sentiment Lexicon

For Arabic Sentiment analysis we used the Arabic SentiWordNet lexicon (ArSenL), created by Badaro et al. [8], and which is similar to the English SentiWordNet. In addition, the ArSenL lexicon contains 28,780 lemmas and 157,969 synsets [14]; each lemma in the lexicon has three scores associated with three sentiment labels (positive, negative, and objective).

## 3. Sentiment Classification

There are different ways to calculate the sentiment of a text when using SentiWordNet as a sentiment lexicon. SentiWordNet lexicon assigns different sentiment weights to different words. However, the classification begins with a search for words in SWN with their corresponding synsets. Each word in SWN contains numbers of synset, and each synset contains three types of scores: positive, negative, and neutral. For classification, first calculate the arithmetic average of the positive, negative and neutral scores of each word synsets of the tweet by summing up the different scores of each synset class and dividing by the total number of synsets, see equation (1):

$$\begin{aligned}
 \text{Score pos (W)} &= \frac{\sum_{i=1}^n \text{score pos (Si)}}{n} \\
 \text{Score neg (W)} &= \frac{\sum_{i=1}^n \text{score neg (Si)}}{n} \\
 \text{Score neu (W)} &= \frac{\sum_{i=1}^n \text{score neu (Si)}}{n}
 \end{aligned} \quad (1)$$

Where S is the score of synset i, for word, and n is the number of synsets of word, where W is a word that belongs to a set of words.

The second step in classification is determined by the sentiment scores of the tweet by averaging each word score in T, where it represents a single tweet from the set of tweets; see equation (2):

$$\begin{aligned}
 \text{Total Score pos (T)} &= \frac{\sum_{i=1}^n \text{score pos (Wi)}}{n} \\
 \text{Total Score neg (T)} &= \frac{\sum_{i=1}^n \text{score neg (Wi)}}{n} \\
 \text{Total Score neu (T)} &= \frac{\sum_{i=1}^n \text{score neu (Wi)}}{n}
 \end{aligned} \quad (2)$$

### 4.1 In Formula (1) [13],[23]

<sup>1</sup> <http://nlp.ldeo.columbia.edu/madamira/>

The class of the tweet corresponds to the greatest positive, negative, and neutral score:

$$\text{Sentiment} = \begin{cases} \text{Positive tweet if } \text{TotalScorepos}(T) > \text{TotalScoreneg}(T) \\ \text{Negative tweet if } \text{TotalScorepos}(T) < \text{TotalScoreneg}(T) \\ \text{Neutral tweet if } \text{TotalScorepos}(T) = \text{TotalScoreneg}(T) \end{cases}$$

#### 4.2 In Formula (2) [17],[19],[26],[9],[22],[11]

The difference of the total positive and negative scores of the tweet represents its overall sentiment score. In addition, the overall score will be in the range (-1, 1) where a value greater than zero denotes a positive sentiment, and a value less than zero denotes a negative sentiment. Otherwise, the tweet is neutral.

$$\text{Total Score} = \text{Total Score pos}(T) - \text{Total Score neg}(T)$$

$$\text{Sentiment} = \begin{cases} \text{Positive if Total Score} > 0 \\ \text{Negative if Total Score} < 0 \\ \text{Neutral if Total Score} = 0 \end{cases}$$

#### 4.3 Proposed Technique to use Arabic SWN Lexicon in Arabic tweet

To improve the efficiency of using SentiWordNet in sentiment classification for Arabic tweets, we propose a technique that consists of two phases:

##### (1) SentiWordNet Interpretation Phase

As previously mentioned, each word in SentiWordNet lexicon can have multiple synsets. Therefore, to assign the positive, negative and neutral scores to each word in the text, we need to be performing word synset disambiguation. Initially we removed the unnecessary words from each tweet. Then, we had to import the Arabic Sentiment lexicon (ArSenL) as an MS excel sheet; after that, we looked up each word to check if it belonged to the list of Arabic negation words; if true, the sentiment score was assigned as negative (Scoreneg= 1; Scorepos= 0; Scoreneu = 0). If not, we searched it in ArSenL; if we found it there, we would take all possible synsets scores that are relevant with the POS of the word. If we did not find it, the word would be stemmed by ISRI and then we searched for it again in the ArSenL; also if we did not find it in the lexicon, we would stem the word by Light Stemmer as a second option to find the word in the lexicon. Finally, if the word was not found there either, we would assign a zero value to all scores.

---

#### Algorithm 1 : Word Score Summation

---

```

if WordAveragPositive >= 0 then
  TotalWord= WordAveragPositive - WordAveragNegative
  if TotalWord > 0 then
    | TotalPositiveScor+= WordAveragPositive
  end
  else if TotalWord < 0 then
    | TotalNegativeScor+= WordAveragNegative
  end
  else
    | TotalNeutralScor+= WordAveragNeutral
  end
end
if WordAveragPositive == 0 then
  TotalWord= WordAveragNeutral - WordAveragNegative
  if TotalWord >= 0 then
    | TotalNeutralScor+= WordAveragNeutral
  end
  else if TotalWord < 0 then
    | TotalNegativeScor+= WordAveragNegative
  end
end
end

```

Figure 4. Algorithm1.

##### (2) Sentiment Calculations Phase

In this phase, we apply two calculation methods on tweets to classify them as positive, negative or neutral using word scores obtained from the first phase.

**Word score summation method:** In this method, we applied Algorithm (1) where summation is the positive, negative and neutral scores for each word in a tweet then calculated the average of each category's scores in each word using Equations (2). Then, to determine the sentiment of the word we followed Formula 2, where we took only the result score in the second step and ignored the other score. To classify each word, we had to calculate the total result of each class (positive, negative and neutral) score. Before applying Algorithm (2) we only considered the result of each word to calculate the total result of the tweet and ignored the other scores.

**Tweet score summation method:** In this method, we summarize all the positive, negative and neutral scores obtained from the first step. We applied Algorithm (2) to solve the neutral classification issue; we implemented a condition that if the total positive score equals or is more than 0.25, then we performed Formula 2; if not, we implemented another condition, which is: if the difference of the total negative score and the total neutral score is less than zero if the total negative score is more 0.25, then the sentiment of the tweet is negative. Since all words in the SentiWordNet lexicon have three values (positive, negative and neutral), most of the words have low positive and negative scores and high neutral scores. So, we applied these thresholds to reduce the neutral results.

#### 4.4 Evaluation Measures

In order to decide whether the classifiers were accurately capturing a pattern, we evaluated the model. Furthermore, confusion matrices, precision,

recall, F-measure and accuracy methods were used for evaluating the proposed classifiers performance and comparing them with each other.

---

**Algorithm 2** : Tweet Score Summation

---

```

if TotalPositiveScore >= 0.25 then
  TotalTweet=TotalPositiveScore-TotalNegativeScor
  if TotalTweet > 0 then
    | Print Tweet is Positive
  end
  else if TotalTweet < 0 then
    | Print Tweet is Negative
  end
  else
    | Print Tweet is Neutral
  end
end
else
  TotalTweet=TotalNeutralScor-TotalNegativeScor
  if TotalTweet < 0 Or TotalNegativeScor >= 0.25 then
    | Print Tweet is Negative
  end
  else if TotalTweet < 0 then
    | Print Tweet is Neutral
  end
end
end

```

**Figure 5.** Algorithm 2.

## 4. Results And Discussion

In this section, we evaluate our semantic approach in a Twitter Arabic sentiment analysis by two experiments performed on Arabic tweets: In the first experiment, we used the Arabic Sentiment Lexicon (ArSneL) and in the second, we used the English SentiWordNet. In addition, we presented the results of different experiments by observing the performance of each classifier to obtain the best one for Arabic sentiment analysis. Also, we measured the effectiveness of both sentiment lexicons: Arabic and English.

### 5.1 Experiment 1

In this experiment, we studied the performance of using the Arabic Sentiment Lexicon (ArSneL) classifier in an Arabic sentiment analysis of Twitter. Also, we tested the effect of preprocessing on the ArSneL classifiers performance, as well as the effect of using Arabic stemming tools, the Arabic NLP Stanford POS tagging tool and negation on the performance. We divided this experiment into four classifiers.

In classifier (1), we used an unlabeled Arabic tweet dataset that consists of 2,000 tweets, which we collected before the first experiment and which were cleaned of noisy Twitter symbols and duplicated tweets. We applied both formulas (1 and 2) and used ArSenL on the unlabeled Arabic tweets dataset that consists of 2,000 tweets, and which was cleaned of noisy Twitter symbols and duplicated tweets. In

addition, classifier (2) was applied to the same dataset in classifier (1) but using the proposed technique. In classifier (3), we applied different preprocessing steps to this dataset, such as tokenization, spelling correction and the removal of all punctuation, Arabic stop words, tashkeel, and proper nouns, but there was little impact on the results of the classification. According to the structure of SentiWordNet, part of speech (POS) is considered an important attribute. Therefore, we used POS tagging as a feature in the classifier (4) by using the Arabic NLP Stanford POS Tagging tool. We also used ISRI and Light stemming, as well as handling negation. We applied these features because they may support and accelerate the process of extracting words from the ArSneL lexicon. After extracting scores for each word, we applied our proposed technique for the sentiment classification of Arabic tweets into one of three classes: positive, negative or neutral. All these steps were implemented using Python.

For the evaluation of this experiment, we compared the results of classifying the labeled dataset that we used in the previous experiments by four performance measures: confusion matrices, precision, recall, F-measure and accuracy.

In the first experiment, as shown in Table 2, we applied Formula (1) and (2) and used ArSenL to achieve an accuracy of 41 %, with high precision in the positive class of 72.91 %, but less than reasonable in the neutral class at 2.18 %. Also, in the second classifier we classified the same dataset but used the proposed technique to reduce the neutral result, where the performance accuracy improved, at 42.75 % with the logical distribution of the precision, recall and F-measure percentages.

With regards to improving the performance of the second classifier, we applied different preprocessing steps to the third classifier. However, there was very little improvement in the result of 0.40 %, but, on the other hand, the process was accelerated. Some issues led to this result, which we will explain in detail: We aimed to remove the stop words from the list because some words may cause confusion in classification. For example, ("الذي" - Who) and ("انا" - I) are stop words that have neutral weight equaling 1 in Arabic and do not add any sentiment to the tweet, so we kept them on the stop words list, but all negation words were removed from the list such as ("لا" and "لم" - No).

**Table 1.** The synsets of word "من" in ArSneL lexicon.

| POS | Positive | Negative | Objective | English Equivalent  |
|-----|----------|----------|-----------|---|
| V   | 0.125    | 0        | 0.875     | yield, grant, concede, allot, accord, award   |
| V   | 0        | 0        | 1         | grant, deed_over, confer, bestow, yield, concede, cede, give  |
| V   | 0.50     | 0.25     | 0.25      | lend, impart, contribute, bring, bestow, add  |
| N   | 0.125    | 0        | 0.875     | favour, favor   |
| N   | 0        | 0        | 1         | party_favour, party_favor, favour, favor, thanksgiving, grace, blessing, state_of_grace, saving_grace |
| N   | 0.778    | 0.222    | 0         | grace, goodwill, good_will  |
| N   | 0.50     | 0        | 0.50      | grace_of_god, grace, free_grace, seemliness, favour, favor  |

**Table 2.** The results of Arabic tweets sentiment analysis classifier.

| Arabic Sentiment Lexicon  |          | Confusion matrices |          |         | Results   |         |           | Accuracy |
|---------------------------|----------|--------------------|----------|---------|-----------|---------|-----------|----------|
|                           |          | Positive           | Negative | Neutral | Precision | Recall  | F-measure |          |
| <b>Formula (1-2)</b>      | Positive | 463                | 323      | 426     | 72.91 %   | 31.48 % | 43.97 %   | 41.00 %  |
|                           | Negative | 136                | 342      | 248     | 50.59 %   | 28.79 % | 36.70 %   |          |
|                           | Neutral  | 36                 | 11       | 15      | 2.18 %    | 2.88 %  | 2.48 %    |          |
| <b>Proposed Technique</b> | Positive | 227                | 111      | 187     | 35.75 %   | 23.55 % | 28.39 %   | 42.75 %  |
|                           | Negative | 108                | 247      | 121     | 36.54 %   | 25.65 % | 30.14 %   |          |
|                           | Neutral  | 300                | 318      | 381     | 55.30 %   | 31.28 % | 39.96 %   |          |
| <b>Preprocessing</b>      | Positive | 381                | 269      | 302     | 60.00 %   | 30.14 % | 40.13 %   | 43.15 %  |
|                           | Negative | 74                 | 228      | 133     | 33.73 %   | 24.86 % | 28.63 %   |          |
|                           | Neutral  | 180                | 179      | 254     | 36.87 %   | 26.57 % | 30.88 %   |          |
| <b>POS</b>                | Positive | 376                | 233      | 310     | 59.21 %   | 29.79 % | 39.64 %   | 42.75 %  |
|                           | Negative | 64                 | 196      | 96      | 28.99 %   | 22.76 % | 25.50 %   |          |
|                           | Neutral  | 195                | 247      | 283     | 41.07 %   | 27.69 % | 33.08 %   |          |

**Table 3.** The results of translated Arabic tweets sentiment analysis classifier.

| English SentiWordNet         |          | Confusion matrices |          |         | Results   |         |           | Accuracy |
|------------------------------|----------|--------------------|----------|---------|-----------|---------|-----------|----------|
|                              |          | Positive           | Negative | Neutral | Precision | Recall  | F-measure |          |
| <b>Without Preprocessing</b> | Positive | 445                | 254      | 371     | 70.08 %   | 23.07 % | 34.71 %   | 45.60 %  |
|                              | Negative | 65                 | 257      | 108     | 38.02 %   | 27.75 % | 32.08 %   |          |
|                              | Neutral  | 125                | 165      | 210     | 30.48 %   | 25.64 % | 27.85 %   |          |
| <b>Preprocessing</b>         | Positive | 312                | 204      | 217     | 49.13 %   | 27.88 % | 35.58 %   | 43.50 %  |
|                              | Negative | 51                 | 160      | 74      | 23.67 %   | 20.67 % | 22.07 %   |          |
|                              | Neutral  | 272                | 312      | 398     | 57.76 %   | 32.17 % | 41.33 %   |          |
| <b>POS</b>                   | Positive | 448                | 275      | 363     | 70.55 %   | 32.18 % | 44.20 %   | 43.45 %  |
|                              | Negative | 48                 | 178      | 83      | 26.33 %   | 21.95 % | 23.94 %   |          |
|                              | Neutral  | 139                | 223      | 243     | 35.27 %   | 26.19 % | 30.06 %   |          |

There are other words we could not remove such as ("من"- Mn), which in most Arabic tweets means from or who. On the other hand, the word ("من"- Mn) could mean grant or favor or dead\_over, but when it is written with tashkeel "مَنْ", it is different from "مَنُ", where the former means grant and the latter means who. In Table 1, the ArSenL lexicon has different synsets for the word "من" with the total scores being positive = 3.528, negative = 0.597, neutral = 5.875. However, before removing any stop words, we must remove the tashkeel from the words to avoid these problems. Another issue is misspellings. While a small

percentage of tweets are correct and do not include any misspelled words, most mistakes are made in vowel letters; for example "نتهمه" should be "نتهمه" with (ha' alddamir - هاء الضمير) not (altta' almarbuta- التاء المربوطة), the other word "إستعماري", should not include hamzah "استعماري", in the word "اوربا" the "و" should be included to be "اوروبا", and the word "اليونانيين" should be written as plural (Greek), but with the "ي" letter included at the end to be "اليونانيين". However, all these words and those like them must be preprocessed regarding spelling correction and stemming before classification.

| Means | Tweet   |
|-------|---|
| From  | يارب توفيق الحب و يرجع من طهران بثلاث نقاط و صداره  |
|       | Oh God, help my lover (Alhilar FC) and back from Tehran with three points and be the best |
| Who   | من لم يتقن فن الصمت لن يتقن فن الكلام   |
|       | Those who have not mastered in silent skill will not mastering in speech skill            |

Figure 6. Word "من" examples.

In the POS classifier, we added the POS tagging feature to our Arabic sentiment classification. We expected POS would play a major role in sentiment analysis because it is an important attribute in sentiment lexicons. However when using POS tagging, there have not been any noticeable improvements in the performance results of the Arabic SWN classifier at almost 42.75%, aside from accelerating the search process.

On the other hand, we also applied the stemming feature in the POS classifier, including both the Arabic stemmer methods ISRI and Light sequentially; as we explained, the previous ISRI method relies on extracting the root of the word by looking it up in a root dictionary. Therefore, we applied the first step, and if the root of the word was not found, the classifier would go to the next step, Light stemming. This technique allowed us to take advantage of both methods, which may provide a greater possibility of finding a word in the ArSenL lexicon. ISRI Arabic stemming is generally a good method, but it requires updating and a better review of roots because there are many mistakes; for example, if we stemmed the word "بارضنا" meaning our land, it becomes "برض" and this is incorrect. However, if we apply our technique it will not solve the problem because the word "برض" has a score in the ArSenL lexicon.

## 5.2 Experiment 2

In this experiment, we followed the same steps as in the first experiment, but we used English tweets and English SentiWordNet. Initially, we translated all the Arabic tweets dataset to English using Google Translator API. The total number of tweets we translated was 2,000 Arabic tweets; however we translated twice: the first time without preprocessing and the second after preprocessing. In Table 3, it is clear that when we classified the translated tweets without any preprocessing, the classifier works well, producing fewer classification errors with an accuracy of 45.60%, but in the second classifier after preprocessing the accuracy was 43.50%. In the third classifier, we applied the English POS tagging feature on the translated tweets; after preprocessing we noticed no improvement in accuracy as a whole, but the

Positive precision clearly improved with 21.42% from the first classifier.

Furthermore, this experiment revealed a number of issues that led to an incorrect translation of the Arabic tweets, and in addition, the classifier was not able to manage this:

One of these issues was that some words, such as "كوباني" were translated to be "Kobânî", contained non-English letters such as (â, è, î, ü), so English SWN does not contain words with these letters.

Also there were many misspelled words in the Arabic tweets, as previously mentioned, which is a problem that continued to be one reason for incorrect translations, for example in this tweet "اقسم بالله ان اهل اليمن كلهم شرفاء", the word "كلهم", which mean "All" is incorrect, so when we translate, it becomes "swear god people kllahm honest". Another issue was that some derivative words could not be translated by Google (see Figure 7).

|    |  |
|----|--|
| Ar | أعشق قراءة تغريداتها كل يوم حتى و أن لم أتابعها فكل تغريدة تكتبها لها ذكرى بداخل |
| En | Adore reading <u>n</u> gredatha otabaha every tweet <u>tk</u> otha memory inside |

Figure 7. Google translate examples.

With regard to this issue, the tweet in Figure 7 contains the word "تغريداتها"-"her tweets", derived from the word "تغريدة" - "tweets", which Google failed to translate, and also there is the misspelled word, "تكتبها"; the correct word should be "تكتبها" - "Writes it". All these words were neglected by the classifier, which contributed to the low result. Thus, the correct translation for this tweet if there were no misspellings is "I love to read her tweets every day, even if I do not follow her. Any tweet she writes has a memory inside me".

Another issue that appeared in this experiment was with homonyms, which are a group of words that share the same spelling but that have different meanings such as "like" (similar or comparable) and "like" (agreeable or enjoyable). Furthermore, the classifier could not distinguish between the meanings of words, so we had to overcome this issue by translating each Arabic word individually. For example, the translation of the sentence "أنا أحب التفاح" becomes "I like apple"; however, the same sentence translated manually is, "I love apple".

|                            |                |      |     |
|----------------------------|----------------|------|-----|
| Arabic sentence            | أنا أحب التفاح |      |     |
| Translate sentence         | I like apple   |      |     |
| Separate words of sentence | التفاح         | أحب  | أنا |
| Translate each word        | Apple          | Love | I   |

Figure 8. Homonym example.

By using the first method, the sentiment classifier classified words "like" to neutral with total scores of scorepos=2, scoreneg=0 and scoreneu=3; however, when classifying the word "love", it was positive with total scores of scorepos=2.5, scoreneg=0.125 and scoreneu=1.375.

### 5.3 Comparison

**Table 4.** Accuracy of sentiment analysis systems on BBN datasets.

| System         | Lexicon                              | Accuracy |
|----------------|--------------------------------------|----------|
| Our system     | Arabic SentiWordNet                  | 60,25%   |
| SMK system [6] | Arabic Hashtag<br>Lexicon(dialectal) | 63.89%   |

We tested our system on the existing BBN Blog Posts Sentiment dataset<sup>2</sup>. Table 4 shows the results of ten-fold cross-validation experiments on the dataset, where the results shown for systems have to identify one of three classes: positive, negative, or neutral. The accuracy of our system is convergent to previously published results by Salameh et al. system [6], where they obtained an accuracy of 63.89%.

## 5. Conclusion And Future Work

In this paper, we explored a new model of Arabic tweets sentiment analysis using a semantic approach, and the effect of different techniques of text preprocessing on the classifications accuracy. The main contribution of this paper is the implementation of a novel Arabic tweets sentiment analysis system by using both Arabic and English sentiment lexicons. With regard to the results, it is clear there are no noticeable effects on performance when applying preprocessing and the POS tagging feature. This is because there are a lot of misspellings and dialect words in Arabic tweets. Moreover, most of the good text preprocessing tools do not support the Python programming language. Furthermore, the stemmer tools produced an incorrect root or a root without meaning. Hence, we conclude that there are two tools that must be improved to support and improve any Arabic natural learning process system. The first is the Arabic spelling correction tools, which need to be developed and more words added. The second is the Arabic stemmer tools that need to include more roots in order to support the searching process.

## References

[1] S. M. Oraby, Y. El-Sonbaty, and M. A. El-Nasr, Exploring the Effects of Word Roots for Arabic Sentiment Analysis, in *Proceedings of the International Joint Conference on Natural Language Processing*. Nagoya, Japan, pp.471-479, 2013.

[2] J. O'Mahony, Twitter users 'can't spell' Telegraph, <http://www.telegraph.co.uk/technology/twitter/10086819/Twitter-users-cant-spell.html>, December 2015.

[3] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, Sentiment classification: The contribution of ensemble learning, in *Decision support systems*, pp.77-93, 2014.

[4] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, Sentiment analysis of twitter data, in *Proceedings of the Workshop on Languages in Social Media*, pp. 30-38, 2011.

[5] S. Asur, and B. Huberman, Predicting the future with social media, in *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pp. 492-499, 2010.

[6] M. Salameh, S. M. Mohammad, and S. Kiritchenko, Sentiment After Translation: A Case-Study on Arabic Social Media Posts, in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL-2015)*, 2015.

[7] Y. Yu, and X. Wang, World Cup 2014 in the Twitter World: A big data analysis of sentiments in US sports fans' tweets, *Computers in Human Behavior*, pp. 392-400, 2015.

[8] A. R. Hedar, and M. M. Doss, Mining Social Networks' Arabic Slang Comments, in *Proceedings of IADIS European Conference on Data Mining (ECDM'13)*. Prague, Czech Republic, 2013.

[9] K. Denecke, Using sentiwordnet for multilingual sentiment analysis, in *Data Engineering Workshop, ICDEW 2008, IEEE 24th International Conference*, pp. 507-512, 2008.

[10] M. Anjaria, and R. M. R. Guddeti, A novel sentiment analysis of social networks using supervised learning, *Social Network Analysis and Mining*, vol.4, no.1, pp. 1-15, 2014.

[11] A. Montejo-Ráez, E. Martínez-Cámara, M. T. Martín-Valdivia, and L. A. Ureña-López, Ranked wordnet graph for sentiment polarity classification in twitter, *Computer Speech & Language*, vol.28, no.1, pp. 93-107, 2014.

[12] K. Denecke, Are SentiWordNet scores suited for multi-domain sentiment classification?, in *Digital Information Management, ICDIM 2009, Fourth International Conference*, pp. 1-6, 2009.

[13] M. Guerini, L. Gatti, and M. Turchi, Sentiment analysis: How to derive prior polarities from SentiWordNet, The arXiv, pp. 1309-5843, 2013.

<sup>2</sup> <http://saifmohammad.com/WebPages/ArabicSA.html>



- [14] G. Badaro, R. Baly, H. Hajj, N. Habash, and W. El-Hajj, A large scale Arabic sentiment lexicon for Arabic opinion mining, *ANLP 2014*, 2014.
- [15] A. Shoukry, and A. Rafea, Sentence-level Arabic sentiment analysis, in *Proceedings in International Conference of the Collaboration Technologies and Systems (CTS)*, pp. 546-550, 2012.
- [16] S. Khoja, APT: Arabic part-of-speech tagger, in *Proceedings of the Student Workshop at NAACL*, pp. 20-25, 2001.
- [17] E. Martínez-Cámara, S. M. Jiménez-Zafra, M. T. Martín-Valdivia, and L. A. Urena-López, SINAI: Voting System for Twitter Sentiment Analysis, *SemEval 2014*, 572, 2014.
- [18] A. Hamouda, and M. Rohaim, Reviews classification using sentiwordnet lexicon, In *World Congress on Computer Science and Information Technology*, 2011.
- [19] F. H. Khan, S. Bashir, and U. Qamar, TOM: Twitter opinion mining framework using hybrid classification scheme, *Decision Support Systems*, pp. 245-257, 2014.
- [20] E. Refaee, and V. Rieser, Evaluating Distant Supervision for Subjectivity and Sentiment Analysis on Arabic Twitter Feeds, *ANLP 2014*, 2014.
- [21] N. El-Makky, K. Nagi, A. El-Ebshihy, E. Apady, O. Hafez, S. Mostafa, and S. Ibrahim, Sentiment Analysis of Colloquial Arabic Tweets, 2015.
- [22] M. Abdalkader, Sentiment Analysis of Egyptian Arabic in Social Media, 2014.
- [23] R. Ortega, A. Fonseca, M. Mendoza, and Y. Gutierrez, SSA-UO: unsupervised Twitter sentiment analysis, in *Second Joint Conference on Lexicon and Computational Semantics*, vol.2, pp. 501-507, 2013.
- [24] A. Mourad, and K. Darwish, Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs, in *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 55-64, June 2013.
- [25] A. S. Aruna, and P. K. Wilson, Sentiment Analysis using Linguistic Structures-(Adv-Adj-Noun), in *International Journal of Advanced Research in Computer Science*, vol.5, no.1, 2014.
- [26] R. Pandarachalil, S. Sendhilkumar, and G. S. Mahalakshmi, Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach, *Cognitive Computation*, pp. 1-9, 2015.
- [27] A. Pak, and P. Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining, in *LREC*, vol.10, pp. 1320-1326, 2010.
- [28] X. Zhu, S. Kiritchenko, and S. M. Mohammad, Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets, *SemEval 2014*, 2014.

**Lamia Ali Al-Horaibi** received her M.S. degree in Information Systems from Al-Imam Muhammad Ibn Saud Islamic University (ImamU), Saudi Arabia, in 2016. She is working now as system analyst and developer in Princess Nourah bint Abdulrahman University, Saudi Arabia.

**Muhammad Badruddin Khan** He received his PhD degree from Tokyo Institute of Technology, Japan. He is currently an assistant professor in the Department of Information Systems, College of Computer and Information Sciences, Al-Imam Muhammad ibn Saud Islamic University.