Journal of Theoretical and Applied Information Technology
30th June 2022. Vol.100. No 12
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

QURANIC COLLOCATIONS EXTRACTION USING STATISTICAL MEASURES

MAJED ABUSAFIYA[1]

[1]Al-Ahliyya Amman University, Department of Software Engineering, Jordan

E-mail: [1]majedabusafiya@gmail.com

ABSTRACT

Collocations are a common phenomenon in natural languages in general and in Arabic in specific. Quran is widely considered to be the most eloquent and authentic Arabic text. This fact should motivate researchers to explore and study the collocations of this miraculous book. One approach to find collocations for a given corpus is by identifying pairs of words that are more likely to exist adjacent than being separate using special statistical measures. *Chi-squared*, *t* and *mutual information* are three broadly accepted statistical measures that are used for this purpose. In this paper, these three statistical measures are used to extract the collocations of Quran. None of these measures gave perfect results. So, a human interaction is required to filter the best candidates from the found collocations. The *t* measure gave the best set of collocations. The other two showed bias towards pairs of adjacent words that are unique or exist with very low frequencies in Quran. To give a generic notion of Quranic collocations that are extracted using this approach, those that were found by the *t* measure are studied and categorized.

Keywords: *Natural Language Processing, Quran, Collocations, Statistical Approach, Algorithms*

1. INTRODUCTION

Natural language processing is becoming a wide research area in information technology field. One common concept of natural languages is what is called collocations. A *collocation* in its simplest form is a pair of words that frequently exist adjacent in the text of a natural language. Linguistics paid attention to this concept for many reasons especially in the context of translation. Collocations may reveal extra meaning that is beyond its word-by-word meaning. Moreover, a word-by-word translation of a collocation may not capture its intended meaning. For example, take the following collocation in Arabic language (ضرب مثلا) that means (*gave an example*). A word-by-word translation is (*hit an example*) which is far from its intended meaning.

Another issue - also related to translation - is that collocations translation requires careful selection of the most suitable words among its synonyms in the translated-to language. This selection of words is not disciplined by a generic rule but is more likely considered to be a convention in the translated-to language. For example, *a group of sheep* is usually called (قطيع أغنام) in Arabic and not (مجموعة أغنام). Moreover, one possible word-by-word translation for this collocation – (قطيع أغنام) - to English is (*a herd of sheep*). This translation reveals the intended meaning, yet the word (*herd*) is not the word that is usually used in English for a group of sheep. Conventionally, the word (*flock*) is used for a group of sheep and (*herd*) is used for a group of cows. This also seen when translating collocations from English to Arabic. Take for example (*flock of birds*) which is conventionally translated to Arabic as (سرب طيور) and not (قطيع طيور). So, the word (*flock*) is once translated to the word (سرب) and once to the word (قطيع).

Quran, the Holy book of Muslims, was and still the focus of many researchers in information technology and in many other fields. A lot of computational research is directed towards serving this miraculous book. In addition to its sacredness, it is widely believed to be the most eloquent and authentic Arabic text. It is also one of the most translated books in History. This motivated the author to computationally extract and study the collocations of this book using the selected statistical approach. Knowing Quranic collocations may be useful in many ways: it would prompt translators to be more careful when translating them. Secondly, it motivates pondering of these tightly associated words to better interpret and understand the verses containing them. Thirdly, it helps writers to select most suitable combinations of words when composing in or when translating to Arabic language.

One widely used approach to find collocations in a given corpus is using specific statistical measures [1]. The main idea of this approach is to find these pairs of adjacent words that are more probably to exist adjacent than existing separate in the corpus under study. In this approach, the statistical measures are calculated for every pair of adjacent words in the corpus. Pairs that score the highest values are assumed to form collocations. In this paper, this approach is used to extract collocations from Quranic text using there statistical measures: *Chi-squared*, *t-values* and *mutual Information*. Collocations that are found by these three measures will be studied and compared to evaluate their validity in finding Quranic collocations. To have a feeling of the collocations that may be extracted using this statistical approach, the collocations that are found by the *t-values* measure will be studied and categorized.

In Literature, many papers may be found in the context of Quranic collocations. Most of them are in the context of Quranic collocations translation [1,2,3]. The author of [4] worked on a series to gather Quranic collocations. The reference [5] was in the context of classifying Quranic collocations. In the context of discovering Quranic collocations computationally, much less number of papers are there. The closest work in this context can be found in [6,7]. A specific tool (GATE) was used where special rules are formulated to define patterns for the collocations to be found (for example *Noun-Noun*). These rules are then applied to Quran. Their approach differs than our approach in many ways. First, our approach is purely statistical. It is based on measuring the association degree that may exist between a pair of adjacent words in corpus. No previous knowledge about the form of the collocation is assumed. Second, extraction of collocations with patterns will assume any pair of word that matches this pattern to be a collocation. This neglects a crucial requirement for a collocation, which is frequent adjacent occurrence of the pair of words that forms the collocation.

This paper is structured as follows: In section 2, the algorithms that were designed to find Quranic collocations are presented. Section 3 introduces the three statistical measures that were used to find collocations. Section 3 presents the results and observations that were found when the approach was implemented. The paper ends with a conclusion and a set of references.

2. PROPOSED ALGORITHMS TO EXTRACT QURANIC COLLOCATIONS

In this section, the algorithms that were designed to find Quranic collocations are presented. Firstly, the basic algorithms are presented: building *QuranicWordsVector*, the merge locations and finding collocation instances for a given pair of words. Secondly, the main algorithm that uses these algorithms is presented. It finds the collocation instances vector for every pair of adjacent Quranic words and uses these vectors to calculate the three statistical measures for these pairs.

2.1 Building Quranic Words Vector

QuranicWordsVector is a vector of records for the distinct words in Quran. A record contains the *string*, *frequency* and *locations* of a given word in Quran. For example, the word (الرحمن) has the frequency 157 and *locations* vector is 2,8,.. etc. This means that this word is located in the third, ninth,.... in the Quranic text (the rank of the first word is 0). Figure-1 shows BUILD-QURANIC-WORDS-VECTOR algorithm that calculates *QuranicWordsVector* from the Quran text.

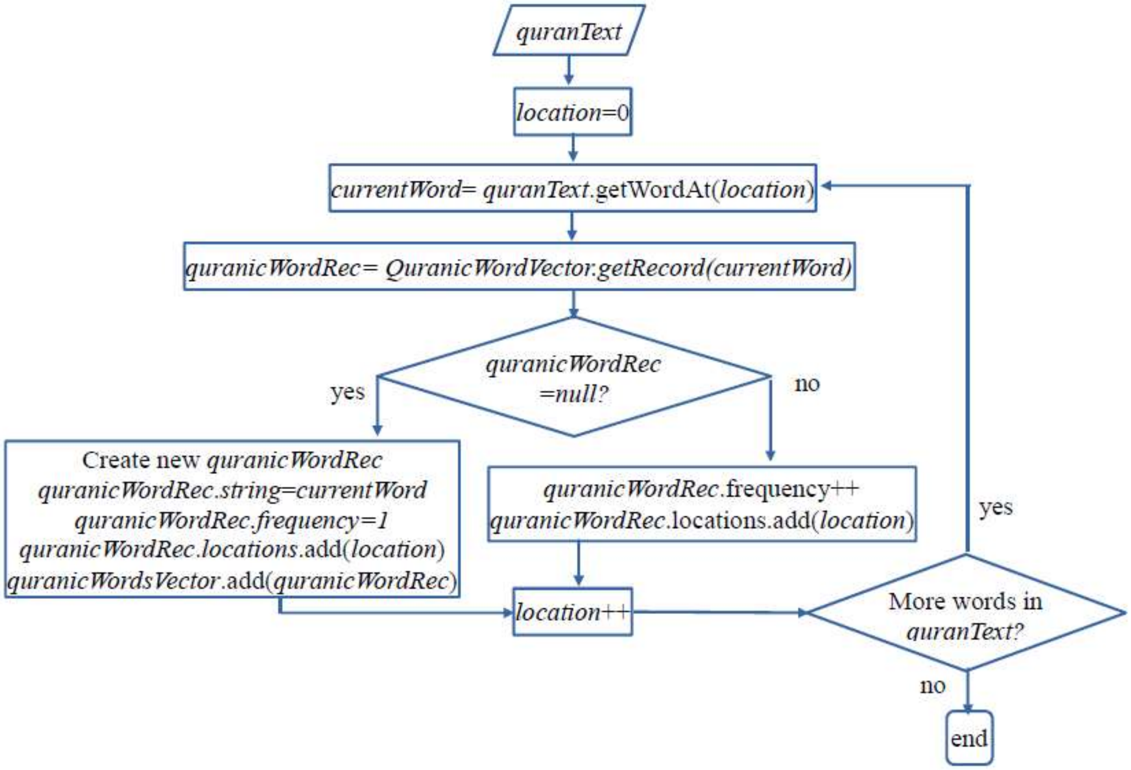
quranText
location=0
currentWord= quranText.getWordAt(location)
quranicWordRec= QuranicWordVector.getRecord(currentWord)
quranicWordRec =null?
yes
no
Create new quranicWordRec
quranicWordRec.string=currentWord
quranicWordRec.frequency=1
quranicWordRec.locations.add(location)
quranicWordsVector.add(quranicWordRec)
quranicWordRec.frequency++
quranicWordRec.locations.add(location)
yes
location++
More words in quranText?
no
end

FIGURE-1 BUILD-QURANIC-WORDS-VECTOR ALGORITHM

In this algorithm, the words of Quranic text are taken a word by word. The *currentWord* is checked if it already has a record in *QuranicWordsVector*. If not, a new record is created by setting the *frequency* field to 1 and its *location* will be added to the *locations* vector field. However, if the *currentWord* already has a record – meaning that it was seen before - the corresponding record is updated by incrementing the *frequency* field and adding its *location* to *locations* vector field. After the *currentWord* is processed, *location* will be incremented and *currentWord* is set to be the next word. The algorithm terminates when all the words in Quran text are processed.

2.2 Merge Locations Vectors

The MERGE-LOCATIONS (Figure-2) algorithm merges the *locations* vector of a given pair of Quranic word records: w_1 and w_2 into one vector. This merged vector is needed to find the occurrences – in corpus – where the corresponding pair of words is adjacent. It is very similar to the merge algorithm that is defined in [8]. Two pointers p_1 and p_2 to the *locations* vector of the two input words are maintained. Initially, they are set to zero. In the loop, the locations (i.e. l_1 and l_2) that are pointed to by these two pointers are retrieved from the corresponding *locations* vectors. If l_1 is less than or equal to l_2 then l_1 is added to the *mergedLocations* vector and p_1 is incremented. Otherwise, l_2 is added and p_2 is incremented. This loop iterates until one of the two *locations* fields is exhausted. In this case, the rest of the *locations* vector of the other word is copied to the *mergedLocations* vector. Figure-3 shows an example to illustrate the algorithm.

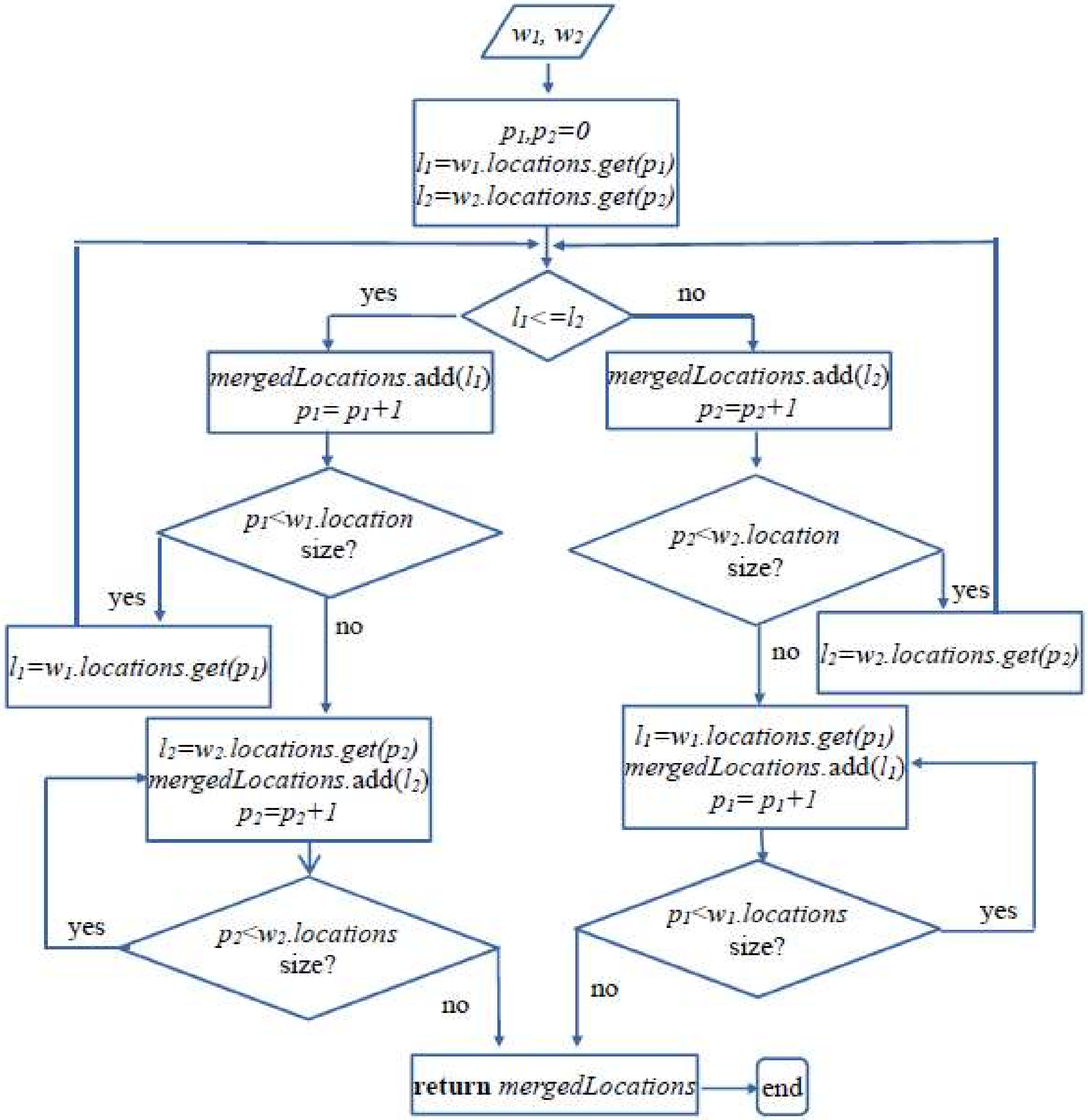
w_1, w_2
p_1,p_2=0
l_1=w_1.locations.get(p_1)
l_2=w_2.locations.get(p_2)
l_1<=l_2
yes
no
mergedLocations.add(l_1)
p_1= p_1+1
mergedLocations.add(l_2)
p_2=p_2+1
p_1<w_1.location size?
yes
no
l_1=w_1.locations.get(p_1)
p_2<w_2.location size?
yes
no
l_2=w_2.locations.get(p_2)
l_2=w_2.locations.get(p_2)
mergedLocations.add(l_2)
p_2=p_2+1
l_1=w_1.locations.get(p_1)
mergedLocations.add(l_1)
p_1= p_1+1
yes
p_2<w_2.locations size?
no
p_1<w_1.locations size?
yes
no
return mergedLocations
end

Figure-2 Merge-Locations Algorithm

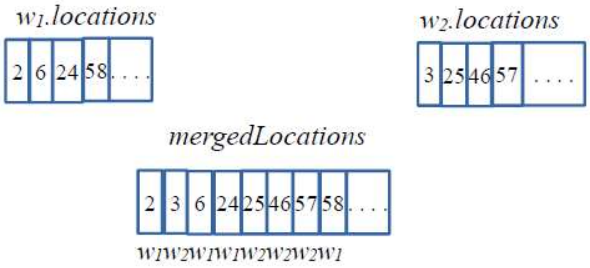

Figure-3 Mergedlocations Returned By Merge-Locations(w_1,w_2)

2.3 Finding Collocation Instances for a Pair of Words

Once the *mergedLocations* for a pair of Quranic words records (w_1, w_2) is produced, it may be scanned to find *collocationInstances* vector. A *collocation instance* is a record that documents the locations for *an* occurrence of w_1*.string* immediately before w_2*.string* in the Quranic text. FIND-COLLOCATION-INSTANCES (Figure-4) is the algorithm that is used to build *collocationInstances* vector for a given pair of words (w_1, w_2). The loop takes the consecutive pairs of elements (l_1 and l_2) within *mergedLocations* one after another. Next, l_1 and l_2 are tested for two conditions: first, they are tested if they are adjacent (i.e. l_2-l_1=1?), second, l_1 is tested if it is a location for w_1 and l_2 is tested if it is a location for w_2. If both conditions are satisfied, a *collocationInstance* is created and added to the *collocationInstances* vector of the pair (w_1, w_2). If one of the two conditions is not satisfied, then the current iteration of the loop is skipped and the next two elements (locations) in *mergedLocations* are taken. Note that there is a *collocationInstances* vector for every pair of words. If the w_1*.string* never existed immediately before w_2*.string* in Quranic text, then their *collocationInstances* vector is empty. Moreover, the *collocationInstances* vector of the pair (w_1,w_2) is not the same as the *collocationInstances* vector for the pair (w_2,w_1). To show how this algorithm works, refer to the example in Figure-3. The *collocationInstances* vector for the pair (w_1,w_2) collocation instances for locations (2,3) and (24,25) while the *collocationInstances* for the pair (w_2,w_1) will contain a collocation instance for locations (57,58).

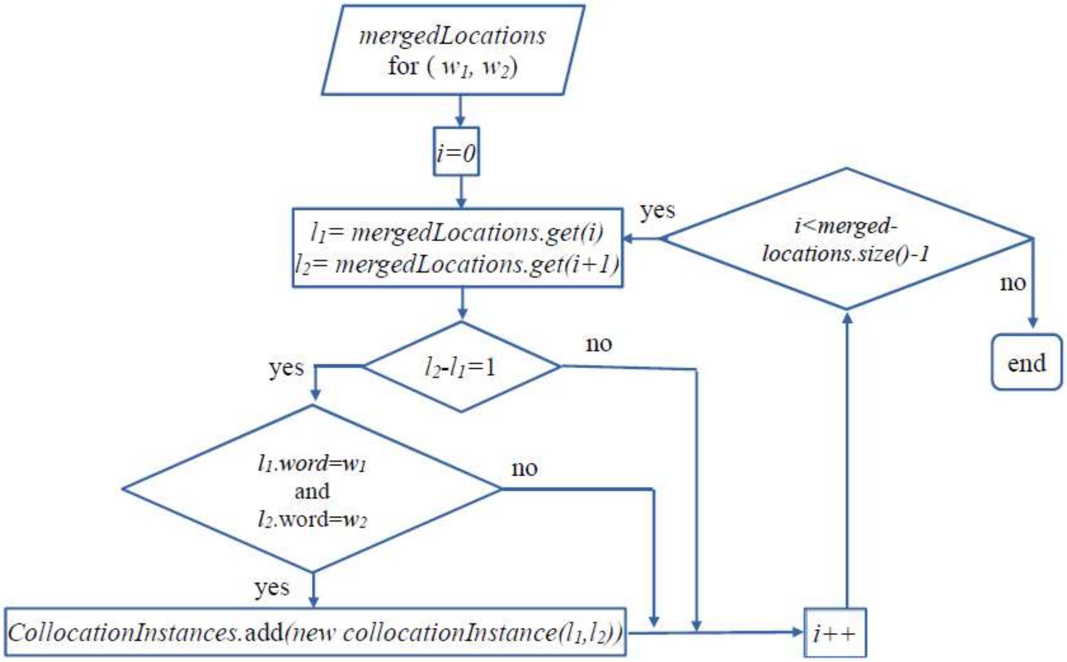
mergedLocations
for (w_1, w_2)
i=0
l_1= mergedLocations.get(i)
l_2= mergedLocations.get(i+1)
yes
i<merged-locations.size()-1
no
end
yes
l_2-l_1=1
no
l_1.word=w_1
and
l_2.word=w_2
no
yes
CollocationInstances.add(new collocationInstance(l_1,l_2))
i++

Figure-4: FIND-COLLOCATION-INSTANCES Algorithm

2.4 Calculating Statistical Measures

The MAIN algorithm (Figure-5) combines the above mentioned algorithms to calculate the three statistical measures for every pair of Quranic words. It calls BUILD-QURANIC-WORDS-VECTOR to build *quranicWordsVector.* Next, two nested loops are executed: the outer loop - indexed with *i* – which iterates over the records in *quranicWordsVector* (excluding the last). The inner loop - indexed with *j* – which iterates over the words beyond *i*. Within the nested loop body, the words records w_i and w_j are taken. The *locations* vectors of these two words are merged to compute the corresponding *mergedLocations* vector. The FIND-COLLOCATION-INSTANCES algorithm is applied on *mergedLocations* vector to generate the corresponding *collocationInstances* vector. Then, *Chi-squared*, *t* , and *MI* values are calculated for this *collocationInstances* vector. The *collocationInstances* vector for w_i and w_j is stored in *collocationInstancesVectcor* vector. The *collocationInstancesVectcor* vector contains a *collocationInstnces* vector for every pair of adjacent words. If w_i word never existed immediately before w_j word in Quran text, then their *collocationInstances* vector will be empty and the statistical values calculation steps are skipped.

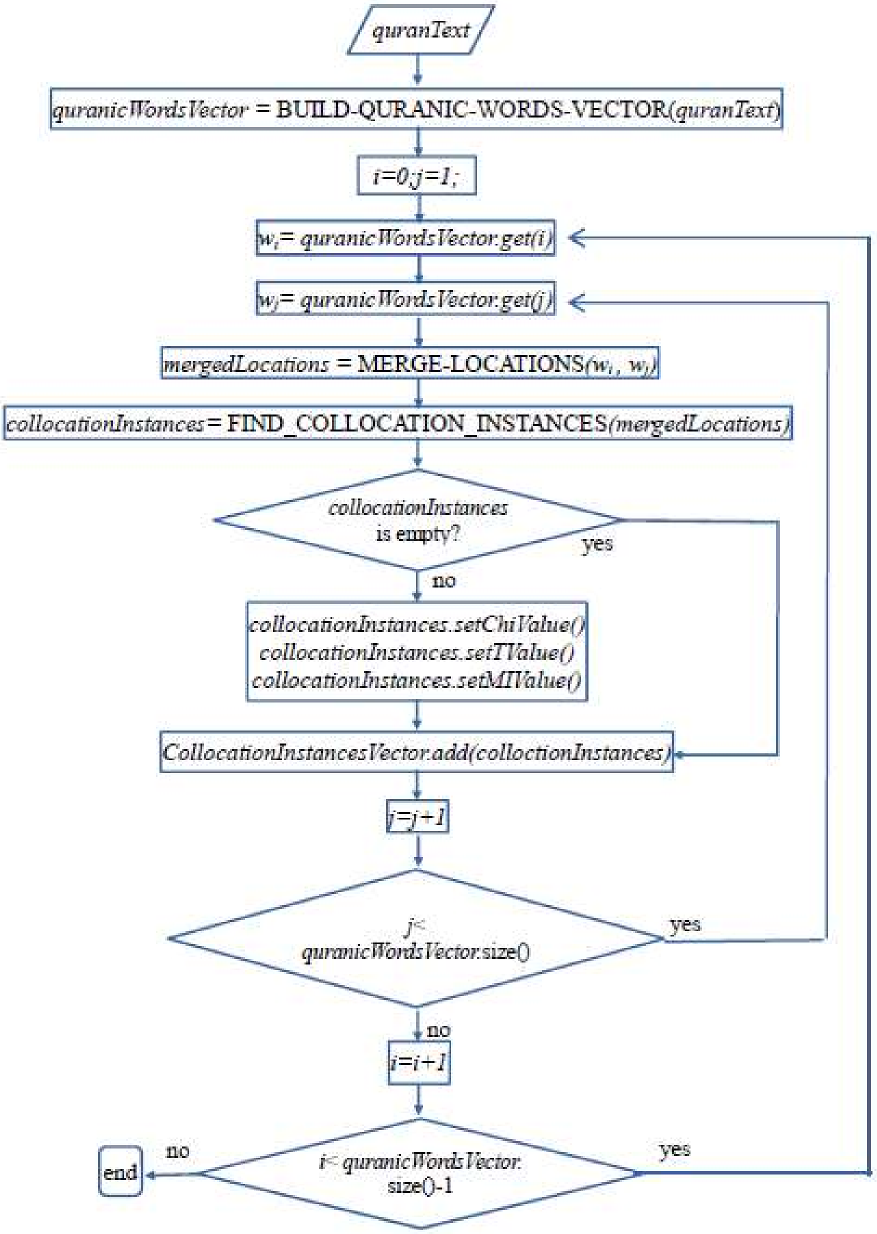
quranText
quranicWordsVector = BUILD-QURANIC-WORDS-VECTOR(quranText)
i=0;j=1;
wi= quranicWordsVector.get(i)
wj= quranicWordsVector.get(j)
mergedLocations = MERGE-LOCATIONS(wi, wj)
collocationInstances= FIND_COLLOCATION_INSTANCES(mergedLocations)
collocationInstances is empty?
yes
no
collocationInstances.setChiValue()
collocationInstances.setTValue()
collocationInstances.setMIValue()
CollocationInstancesVector.add(colloctionInstances)
j=j+1
j< quranicWordsVector.size()
yes
no
i=i+1
end
no
i< quranicWordsVector.size()-1
yes

Figure-5: MAIN algorithm

3. STATISTICAL MEASURES CALCULATIONS

This section shows how these statistical measures are calculated. These calculations are based on [1]. To clarify these calculations, we will show in detail how they are calculated for a given pair of adjacent words (الرحمن الرحيم) where w_1="الرحمن" and w_2="الرحيم". The word w_1 occurred 157 times and w_2 occurred 146 times in Quran. The total number of words is Quran (N) is 78245. This number is larger than the known number of words of Quran because in our count, every token is considered a word (for example: إن ، في ، ما).

3.1 Chi-Squared

To calculate the Chi-squared value for a pair of words w_1, w_2, the following O values are needed: (1) O_{11} is the frequency of w_1 to exist immediately before w_2 in the corpus, (2) O_{12} is the frequency of w_2 word to occur in the corpus without the word w_1 existing immediately before it, (3) O_{21} is the frequency of w_1 to occur without the word w_2 being immediately after it, (4) O_{22} is the frequency of two adjacent words where the first word is *not* w_1 and the second word is *not* w_2. This value equals to $N-(O_{12}+O_{21})$ where N is the total number of words in corpus. In our example, these values are summarized in the following table.

Table 1: O values for collocation (الرحمن الرحيم)

	w_1=الرحمن	w_1 !=الرحمن
w_2= الرحيم	O_{11}=118	O_{12}=28
w_2 != الرحيم	O_{21}=37	O_{22}=78126

We can use O values to calculate Chi-squared for (الرحمن الرحيم) using the following formula and its value will be 48288.22

$$Chi^2 = \frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11}+O_{12})(O_{11}+O_{21})(O_{12}+O_{22})(O_{21}+O_{22})}$$

3.2 *t* value

The *t* value for a pair of words w_1 and w_2 as a collocation is based on the following formula

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}$$

Where *x-bar* is the frequency of having w_1 immediately followed by w_2 in the corpus divided by the size of corpus, μ is the multiplication of frequencies of the two words divided by the size of corpus and s^2 is approximated to be equal to the value of *x-bar*. For our example, *x-bar* is the frequency of the occurrences of (الرحمن الرحيم) in the corpus, which is 118 times divided by the number of words in Quran:

$$\bar{x} = \frac{freq(\text{الرحمن الرحيم})}{N} = \frac{118}{78245} = 0.0015$$

the value of μ

$$\mu = \frac{freq(\text{الرحمن}) * freq(\text{الرحيم})}{N} = \frac{157 \times 146}{78245} = 0\text{،}000003744$$

and variance s^2 is approximated to be *x-bar*, so for the pair of words (الرحمن الرحيم) , *t* value will be

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}} = \frac{0\text{،}0015 - 0\text{،}000003744}{\sqrt{\frac{0\text{،}0015}{78245}}} = \frac{0\text{،}001496256}{0\text{،}0001384578} = 10\text{،}8$$

3.3 Mutual Information

The third statistical measure is the *mutual information*. This measure quantifies how far two words are related. For our example, to see how far the words (الرحمن) and (الرحيم) are related, the value *I(الرحمن , الرحيم)* need to be calculated as follows. These probabilities are calculated by the frequencies.

$$I(\text{الرحمن},\text{الرحيم}) = \log_2 \frac{P(\text{الرحمن الرحيم})}{P(\text{الرحمن}) \times P(\text{الرحيم})} = \frac{118}{157 \times 146} = 8.654$$

4. IMPLEMENTATION AND COMPARISON RESULTS

In this section, the implementation results of this research will be presented with a comparison between the three measures. Quranic collocations that are generated by *t* will also be presented to give the reader an idea about the collocation that may be extracted by this approach.

4.1 Experimental study

The proposed algorithms were implemented with a Java program. The goal was to rank the pairs of adjacent words in Quran according to the values they score for the three statistical measures. A pair of words that scored higher value is closer to be considered as a collocation than pairs with lower values. The Quranic text file was downloaded from *tanzil.net*. The simple clean copy was selected instead of the *Uthmani* copy to avoid the complexities in string matching due to diacritical marks and the specifics of *Uthmani* dictation. Table-

2 shows the top twenty pairs (total was 49558) of adjacent words that scored the highest values for the three statistical measures. These measures were rounded up to three decimal places.

Table 2: Collocations with the highest statistical values

rank	Chi-squared		t-value		Mutual Information (MI)	
	Collocation	value	Collocation	value	Collocation	value
1	وإليك أنبنا	104328.000	الذين آمنوا	13.364	وإياك نستعين	16.256
2	عاما ويحرمونه	104328.000	في الأرض	12.939	نستعين اهدنا	16.256
3	سماعون للكذب	88026.375	إن الله	12.365	ربحت تجارتهم	16.256
4	شهر ورواحها	78246.000	يا أيها	11.860	بنورهم وتركهم	16.256
5	اثنتين وأحييتنا	78246.000	السماوات والأرض	11.501	ورعد وبرق	16.256
6	مكرا ومكرنا	78246.000	الذين كفروا	10.840	ويسفك الدماء	16.256
7	سببا فأتبع	78246.000	الرحمن الرحيم	10.413	نسبح بحمدك	16.256
8	وإياك نستعين	78245 .000	بسم الله	10.251	بحمدك ونقدس	16..256
9	نستعين اهدنا	78245 .000	الله الرحمن	10.251	أنبئوني بأسماء	16.256
10	ربحت تجارتهم	78245 .000	أيها الذين	9.430	بعهدي أوف	16.256
11	بنورهم وتركهم	78245 .000	كل شيء	9.264	أوف بعهدكم	16.256
12	ورعد وبرق	78245 .000	من قبل	9.157	بقلها وقثائها	16.256
13	ميثاقه ويقطعون	78245 .000	من دون	8.758	وقثائها وفومها	16.256
14	ويسفك الدماء	78245 .000	إن كنتم	8.685	وفومها وعدسها	16.256
15	نسبح بحمدك	78245 .000	من بعد	8.603	وعدسها وبصلها	16.256
16	بحمدك ونقدس	78245 .000	دون الله	8.175	بكر عوان	16.256
17	أنبئوني بأسماء	78245 .000	في السماوات	8.099	صفراء فاقع	16.256
18	لآدم فسجدوا	78245 .000	إن الذين	8.074	تسر الناظرين	16.256
19	بعهدي أوف	78245 .000	سبيل الله	8.003	ذلول تثير	16.256
20	أوف بعهدكم	78245 .000	الحياة الدنيا	7.798	اضربوه ببعضها	16.256

By studying the implementation results, the following observations can easily be seen: (1) *Chi-squared* and *MI* measures are too biased towards adjacent words that are never repeated in Quran. However, *t* measure gave more reasonable collocations with less bias towards words that are unique. A person that is familiar with Quran can easily see that *t* value collocations are frequently repeated in Quran, (2) *t* values have varying values for all pairs of words. For the other two measures, a lot of pairs scored the same values and this is even more obvious in *MI* measure (the top 1434 pairs had the same *MI* value). In other words, *t* values measure is more discriminating than *Chi-squared* measure which is turn is more discriminating than *MI* measure, (3) some collocations included adjacent words that are not *first class citizen* words. Take for example (في) for the collocation (في الأرض). Many pairs scored high statistical values but they don't qualify to be collocations based on the conventional definition of a collocation. However, it reflects the frequent occurrence of these pairs in Quran. Other examples are (حتى إذا),(إن في), (تجري من) , (إله إلا).

4.2 Collocation Classes

To give the reader a notion of Quranic collocations in general and a notion of Quranic collocations that may be extracted by statistical measures in specific, the top 1600 collocations based on the *t* measure were studied. They were grouped into classes to facilitate their presentation and the comprehension of their variety. The *t* measure collocations were selected because they were the least biased towards words that are unique in Quran. Those pairs of words with *second class* words or those that do not comply to the conventional conception of collocations were excluded. These collocations are listed in Table-3. The categorization that is selected for these collocations is: (1) names of Allah (أسماء الله الحسنى), (2) adjectives, where the second word is a description for the first, (3) *edhafah*: where the first word is part or semi-part of the second word, (4) *atf:* the first is separated from the second with (و) or (like *and* in English) (5) some miscellaneous collocations.

Table 3: Extracted Collocations Classified

Category	Example
Allah Names	الرحمن الرحيم ، غفور رحيم ، العزيز الحكيم ، السميع العليم ، عليم حكيم ، العزيز الرحيم
Adjective	الحياة الدنيا ، عذاب أليم ، صراط مستقيم ، اليوم الآخر ، القوم الظالمين ، ضلال مبين ، عذاب عظيم ، المسجد الحرام ، أجل مسمى ، الفوز العظيم ، نذير مبين ، أجرا عظيما ، ثمنا قليلا ، أمة واحدة ، القوم الكافرين ، يوم عظيم ، القوم الفاسقين ، سحر مبين ، عذاب مهين ، البلاغ المبين ، عدو مبين ، قرآنا عربيا ، رزقا حسنا ، الفضل العظيم ، العذاب الأليم ، رسول أمين ، صيحة واحدة ، بسلطان مبين ، عذاب مقيم ، أجر عظيم ، كتاب مبين ،حلالا طيبا، صراطا مستقيما ، صبار شكور ، الأسماء الحسنى ، قولا معروفا ، عذابا مهينا ، مكان بعيد ، بأس شديد ، خلق جديد
Edhafah	بسم الله ، دون الله ، سبيل الله ، يوم القيامة ، رب العالمين ، بني إسرائيل ، أهل الكتاب ، خلق السماوات ، ابن مريم ، بآيات الله ، وعد الله ، أكثر الناس ، أصحاب النار ، ملك السماوات ، فضل الله ، آيات الله ، بإذن الله ، رسول الله ، شديد العقاب ، أصحاب الجنة ، رب السماوات ، بذات الصدور ، عالم الغيب ، جنات عدن ، حدود الله ، يوم الدين ، أساطير الأولين ، وبئس المصير ، نار جهنم ، سوء العذاب ، سريع الحساب ، وابن السبيل ، ذكر الله ، عذاب النار ، نعمة الله ، عاقبة الذين ، ستة أيام ، جنات النعيم ، آيات الكتاب ، كتاب الله ، عباد الله ، مثقال ذرة ، سواء السبيل ، غيب السماوات ، فاطر السماوات ، أصحاب الجحيم ، هدى الله ، كل الثمرات ، سبع سماوات ، لأولي الألباب ، خطوات الشيطان ، ولعذاب الآخرة ، خير الرازقين ، عذاب الحريق ، متاع الحياة ، كلمت ربك ، تنزيل الكتاب ، رب العرش ، سبحان الله ، علام الغيوب ، أرحم الراحمين ، لقاء يومكم ، عاقبة المكذبين ، أولي الألباب
Atf	لسماوات والأرض، الليل والنهار ، الدنيا والآخرة ، السماء والأرض ، الغيب والشهادة، يحيي ويميت ، الجن والإنس ، الكتاب والحكمة ، بأموالهم وأنفسهم ، الشمس والقمر ، اليتامى والمساكين ، البر والبحر ، القربى واليتامى ، سبحانه وتعالى ، التوراة والإنجيل ، فرعون وملئه ، جنات وعيون ، المؤمنين والمؤمنات ، موسى وهارون ، المشرق والمغرب ، ترابا وعظاما ، السمع والأبصار ، اسماعيل واسحاق ، مغفرة وأجر ، نوح وعاد ، خير وأبقى ، كفروا وصدوا ، العداوة والبغضاء ، خوفا وطمعا ، سرا وعلانية ، والأبصار والأفئدة ، بكرة وأصيلا ، الأعمى والبصير ، وهاجروا وجاهدوا ، يعقوب والأسباط ، حكما وعلما ، إسحاق ويعقوب ، المساكين وابن ، كلوا واشربوا ، سمعنا وأطعنا ، مغفرة ورزق ، بيننا وبينكم ، كذب وتولى ،
Miscellaneous	الذين كفروا ، وعملوا الصالحات ، آمنوا وعملوا ، خالدين فيها ، وكان الله ، تحتها الأنهار ، والذين آمنوا ، جنات تجري ، كنتم تعملون ، الذين ظلموا ، يا موسى ، الحمد لله ، أنزل الله ، أيها الناس ، الذين كذبوا ، كذبوا بآياتنا ، أوتوا الكتاب ، اعبدوا الله ، فاتقوا الله ، هذا القرآن ، لعلكم تفلحون ، وعمل صالحا ، أوحينا إليك ، آمنوا بالله ، يبسط الرزق

5. CONCLUSION

In this paper, three known statistical measures were used to extract the collocations in Quran. These measures give quantitative values of how far a given word is likely to exist adjacent to another word in a given corpus. These measures were calculated for every adjacent pair of words and then sorted accordingly. Those pairs with the highest values are assumed to form collocation. One main result of this research is that the *t* measure was the best measure to extract collocations. The other two measures are too biased towards pairs of adjacent words that are unique in the corpus. Moreover, It was found that although this computational measures helped in finding collocations, it was found that still some pairs of words had very high statistical measures but they don't define a collocation as defined in linguistics. So, these statistical approaches provide a raw data from which a human interfering is required to filter the most interesting collocations. One important extension of this research is to exclude these words in the corpus that are less likely to form a collocation. This requires an extra pre-processing on the corpus when running the algorithms that find collocations.

REFERENCES

[1] Manning, Christopher, and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[2] H. Badr, and K. Menacere, "Assessing the Translation Quality of Quranic Collocations: For better or for worse." International Journal of Linguistics, Literature and Translation 2, no. 2, 2019, pp 290243.

[3] Obeidat, Adham, and Tengku Sepora Binti Mahadi. "The English Translation of Idiomatic Collocations in The Noble Quran: Problem and Solutions." *Issues in Language Studies* 9, no. 2 (2020): 78-93.

[3] Hassan, Hassan Badr A. *Investigating the Challenges of Translating Arabic Collocations into English with Reference to the Quran*. Liverpool John Moores University (United Kingdom), 2019.

[4] Wardini, Elie. *The Quran: Key Word Collocations (Volume 6): Adjectives, Nouns, Proper Nouns And Verbs*. Gorgias Press, 2021.

[5] Obeidat, Adham Mousa, Ghada Rajeh Ayyad, and Tengku Sepora Tengku Mahadi. "A NEW VISION OF CLASSIFYING QURANIC COLLOCATIONS: A SYNTACTIC AND SEMANTIC PERSPECTIVE." *e-BANGI* 17, no. 7 (2020): 133-144.

[6] Zaidi, Soyara, Ahmed Abdelali, Fatiha Sadat, and Mohamed-Tayeb Laskri. "Hybrid Approach for Extracting Collocations from Arabic Quran Texts." In *Workshop Organizers*, p. 102. 2012.

[7] S. Zaidi, M. Laskri and A. Abdelali, "Arabic collocations extraction using Gate," *2010 International Conference on Machine and Web Intelligence*, 2010, pp. 473-475, doi: 10.1109/ICMWI.2010.5648038.

[8] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein *Introduction to algorithms*. MIT press, 2009.