ISSN: 0067-2904

Finding the Similarity between Two Arabic Texts

Suhad Malallah kadhem, Aseel Qassim Abd Alameer*
Department of Computer Sciences, University of Technology, Baghdad, Iraq

Abstract

Calculating similarities between texts that have been written in one language or multiple languages still one of the most important challenges facing the natural language processing. This work offers many approaches that used for the texts similarity. The proposed system will find the similarity between two Arabic texts by using hybrid similarity measures techniques: Semantic similarity measure, Cosine similarity measure and N-gram (using the Dice similarity measure). In our proposed system we will design Arabic SemanticNet that store the keywords for a specific field(computer science), by this network we can find semantic similarity between words according to specific equations. Cosine and N-gram similarity measures are used in order to find the similar characters sequences. The proposed system was executed by using Visual Basic 2012, and after testing it, it proved to be a worthy for finding the similarity between two Arabic texts (From the viewpoint of accuracy and search time).

Keywords: Arabic Text Similarity, Semantic Similarity, Keyword Extraction, N-Gram, Cosine Similarity Measure, Dice's Similarity Measure.

ايجاد نسبة التشابة مابين نصين عربيين

سهاد مال الله كاظم ، اسيل قاسم عبد الامير*
قسم علوم الحاسوب ، جامعة التكنولوجيا ، بغداد ، العراق.

الخلاصة

ايجاد نسبة تشابه بين نصوص مكتوبة بلغة واحدة أو عدة لغات تعتبر من أهم التحديات التي تواجه معالجة اللغة الطبيعية. هذا العمل يقدم عدة طرق لتشابه النصوص .وفي هذه البحث سوف نقوم بايجاد نسبة التشابه بين نصين عربيين من خلال دمج عدة طرق لقياس التشابه : مقياس التشابة المعنوي ومقياس تشابهة Cosineوتقنية(N-gram باستخدام مقياس تشابهة Dice). في نظامنا المقترح تم تصميم SemanticNetلخزن الكلمات المفتاحية لمجال معين(علوم الحاسوب) ومن خلال هذة الشبكة نستطيع ايجاد التشابة المعنوي بين الكلمات وفق معادلات معينة .استخدام مقياس التشابهة Cosine وتقنية N-gramلغرض ايجاد سلسلة الحروف المتشابهة .ولقد تم تنفيذ النظام المقترح باستخدام اللغة البرمجية فيجول بيسك 2012. بعد ان اثبت اختبار النظام المقترح بانة قيم في ايجاد نسبة التشابة مابين نصين عربيين(من وجة نظر الدقة ووقت البحث) .

Introduction

The text similarity measures are playing an important role in the text related applications and research in assignments such as text summarization, text classification, information retrieval (IR),

*Email: aseelqasim30@yahoo.com

topic detection, document clustering, questions generation, topic tracking, essay scoring, question answering, machine translation, short answer scoring, and others[1].

The text similarity Measures have used in natural language processing (NLP) applications for a long time. The text similarity measure according to Ethan which is to know how two documents alike are, or how alike a document and a query are[2].

Many research themes in the field of Artificial Intelligence (AI) are emerging under this environment, for example, IR, information extraction, information filtering, machine translation, question answering, and so on. And one of the key problems of these themes is the similarity which has close relationship with psychology and cognitive science [3].

The characteristics of polysemy and synonymy that exist in words of natural language have always been a challenge in the fields of NLP and IR [4].

NLP is the heart of AI and has text classification as an important problem area to process different texts by finding out their grammatical syntax and semantics and representing them in the fully structured form [5].

Keywords are a set of significant words in an article that gives high-level narrative of its contents to readers. To produce a short summary of news articles identifying the keyword from a large number of online news data is very useful. Keyword extraction technique is used to extract main features in studies such as text classification, text categorization, IR, topic detection and document summarization [6].

The fundamental part of text similarity is finding the similarity between words which have been used as primary stage in document similarities, paragraph and sentence. There are two ways of similar words: it can be lexically or semantically. If the words have the same sequence character then the words are similar lexically. If the words have used in the same way, or same thing, or opposite of each other, or used in the same context or one of them is a type of another one then the words similar semantically[1].

In our proposed system we will try to find the similarity degree between two Arabic texts semantically and lexically similarity. Semantic similarity measure depend on keyword extraction by using Arabic SemanticNet for a specific field (computer science). The lexical similarity measure will use the cosine similarity based on characters sequence, while the Dice similarity is based on N2gram.

Semantic Net [7]:

The semantic network is a predicate logic as a form of knowledge representation. The idea of semantic networks is it can be used to store the knowledge in form of graph, with objects representing nodes in the world, and arches representing relationships between those objects as shown in Figure-1.

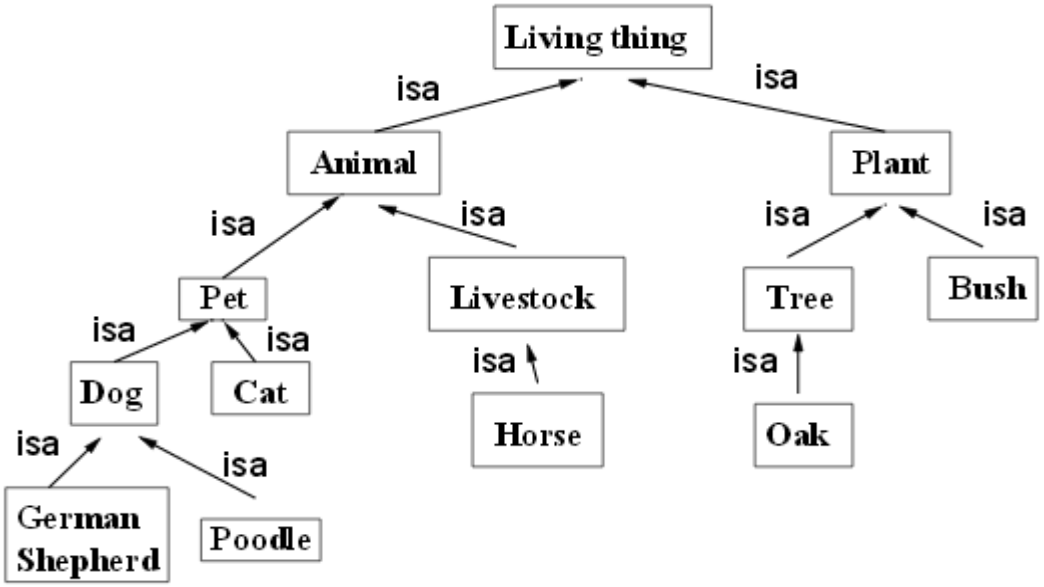

Figure 1-Simple Form of Semantic Network is an is-a Hierarchy Related Theories and Studies:

This work deals with the measure between words of sentences and paragraph in documents using computer science domain. Similarity measure in this domain is an algorithm that determines the degree of agreement between entities. There are many measures which were used to identify the similarity, e.g Semantic similarity, Cosine similarity, Euclidean distance, Jaccard similarity, Overlap coefficient, Matching coefficient, Dice's coefficient and so on, as describe below.

- **Semantic Similarity** [8].

The formula semantic similarity is proposed by Wu & Palmer that takes into account the measure of both path length and depth of the least common sub-summer (LCS) as given in equation (1):

Where

s and **t**: denote the source and target words being compared.

Depth(s): is the shortest distance from root node to a node S.

LCS: denotes the least common sub-submer is the share parent of s and t.

- **Cosine Similarity Measure**

Cosine similarity is a measure of similarity between two vectors of n dimensions by finding the cosine of the angle between them. Given two vectors of attributes, A and B, the cosine similarity is represented using a dot product and magnitude as in equation (2)[9].

For example to find the cosine similarity between "استمرار" and "استمراية" , the first task is taking the union of the characters "استمرار" and "استمرارية", we get the vector:

(ة,ي , ر, م , ت, س , ا).

Converting our previous words " استمرار " and " استمرارية " into frequency of occurrence vectors:

(0 0 2 1 1 1 2) = استمرار

(1 1 2 1 1 1 2) = استمرارية

We can now perform the cosine similarity calculation:

- Dot Product = ((2*2)+(1*1)+(1*1)+(1*1)+(2*2)+(0*1)+(0*1))= 11
- Magnitude of "استمرار" = $\sqrt{(2)^2+(1)^2+(1)^2+(1)^2+(2)^2}= \sqrt{11}$
- Magnitude of "استمرارية" =$\sqrt{(2)^2+(1)^2+(1)^2+(1)^2+(2)^2+(1)^2+(1)^2}=\sqrt{13}$
- Products of magnitudes A & B = $\sqrt{11}*\sqrt{13}=11.9582607431014$
- Divide the dot product of A & B by the product of magnitude of

A & B = 11/11.9582607431014 = 0.9198662110077997 (or about 92% similar).

C. N-gram

Third method is N-gram, it is used to calculate the probability of character sequence that occurs as a word or probability of word sequence that occurs. The length of character are different (it can be 2, 3 and 4 Grams) as in equation(3)[10].

$$Ngram(2,X) = \{x_0x_1, x_1x_2, x_2x_3, \ldots x_{n-1}x_n\} \quad (2)$$

$$Ngram(3,X) = \{x_0x_1x_2, x_1x_2x_3, x_2x_3x_4, \ldots, x_{n-2}x_{n-1}x_n\}$$

The N-gram is performed by converting a word into a sequence of N2grams. The similarity between two words in this study was computed by dividing the number of unique identical N-grams between them by total number of unique N-grams in the two words (Dice's similarity coefficient) as equation (4)[2].

Where

Sim is the value of similarity,

A and **B** are the respective numbers of unique N2grams in word one and word two,

C represents the total number of unique N2grams that are common for both words being compared.

For example**:** the similarity between the two words (تكامل) and (تكامل), the N2grams of these words would be as listed in Table-1. So the similarity is (2*4) / (4+4) =1 (or similarity 100%), and between the two words (تكامل) and (محدود) would be (2*0) / (4+4) =0 (or similarity 0%) and so on.

Table 1- Bi-grams for words

Word	Bi-grams
تكامل	مل , ام , كا , تك
محدود	ود , دو , حد , مح

The proposed system:

The main steps of the proposed system are illustrated as in Figure-2:

- Text Prepossessing
- Keyword Extraction
- Similarity Measures
- Preprocessing

The preprocessing is the first step of the proposed system, includes three stages, these stages are:

- Sentences Cutter.
- Tokenization.
- Stop words removal.
- Normalization process.

The input to the proposed system is an Arabic text, it consists of sentences (a sentence is a set of words, that separated by a stop mark such as "،", ".", "؟" or "!").

In tokenization part, the sentence is converted to a list of tokens, according to the spaces between Arabic words or stop marks.

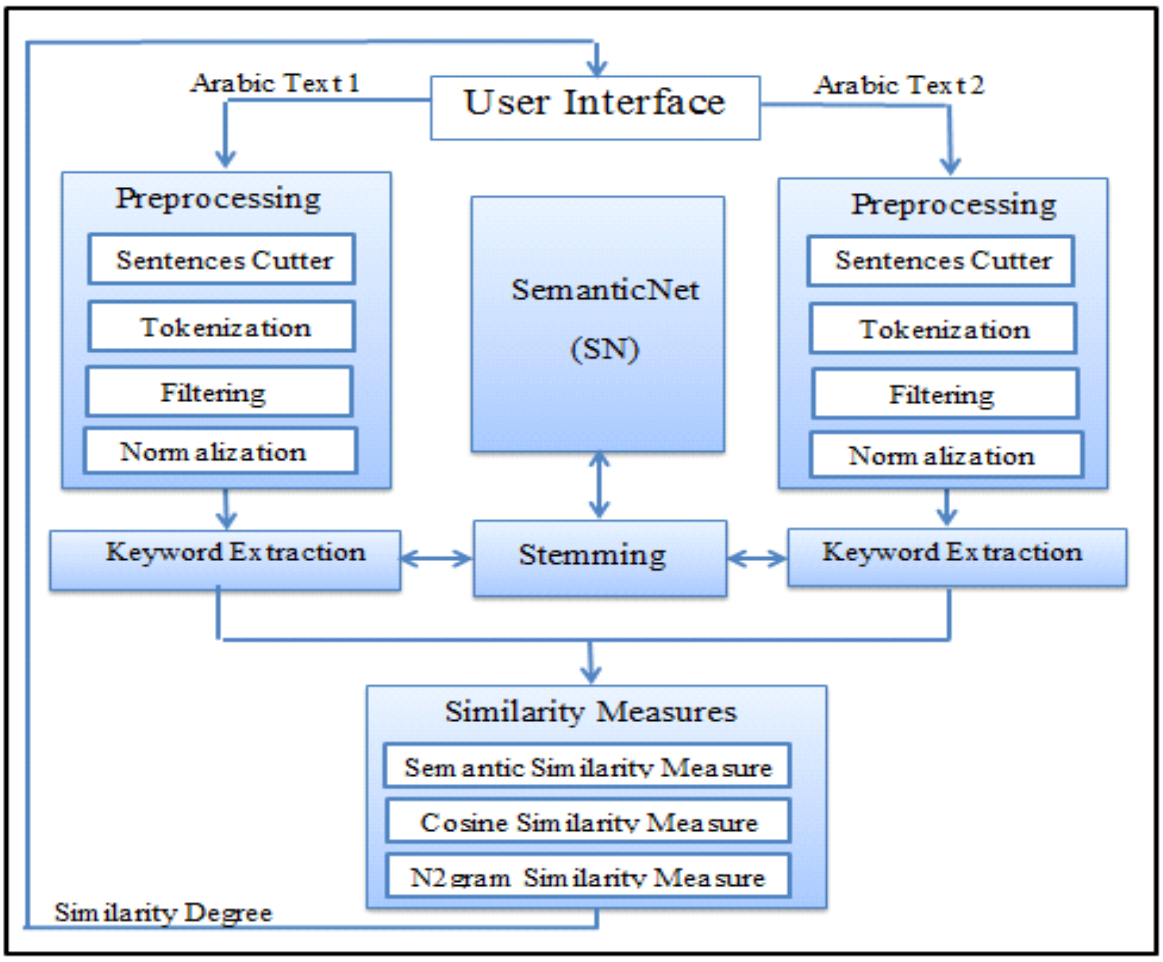

Figure 2- The Block Diagram of the Proposed System

After converting the input Arabic text to a list of tokens, the stop words will be removed. The stop words can be defined as words that don't have any remarkable importance, or any word that don't give any importance in finding the similarity could be considered as a stop word. For example "من", "على", "في", "فوق", "تحت", "هذا", "هذة", "الذي", "التي", "نحن", "اولئك", "عادة" , "حيث" , "طالما" , "كلما" and so on. In our proposed system we will replace the stop word with a flag.

Finally normalization is performed for the characters of two Arabic texts. For example:

أمنيه	امنية
شجره	شجرة
أكثر	اكثر

- Keyword Extraction

Keyword Extraction is a process to identify set of words, keyphrases, keywords that describe the meaning of the input text1 and text2. Extraction is implemented after preprocessing of input texts.

Algorithm 1- illustrate the main steps of extracting the keyphrases and keywords from the input text.

Algorithm(1) : Proposed Keyword Extraction Algorithm

Input: L**:** List of tokens,
SN: SemanticNet of Arabic keywords and keyphrases.

Output: LK: List of keywords,
LR: List of reference**s** of these keywords**.**

Begin
Step1: i=1,
j=1,
Let N= length of L.
Step2: While i <= N do
Begin
2.1: If L[i] != flag and L[i+1] != flag and L[i+2] != flag then
Begin
Concatenate L[i], L[i+1] and L[i+2] to be S
Call prefix algorithm that take S and return S1.
Call prefix_suffix algorithm that take S1 and return S2.
If S2 is found in SN then
Begin
LK[j] = S2
LR[j] = reference of S2
i=i+3

j=j+1
End
End

2.2 If L[i] != flag and L[i+1] != flag then
Begin
Concatenate L[i] and L[i+1] to be S.
Call prefix algorithm that take S and return S1.
Call prefix_suffix algorithm that take S1 and return S2.
If S2 is found in SN then
Begin
LK[j] = S2
LR[j] = reference of S2
i=i+2
j=j+1
End
End

2.3: If L[i] != flag then
Let L[i] to be S
Call prefix algorithm that take S and return S1.
Call prefix_suffix algorithm that take S1 and return S2.
Begin
If S2 is found in SN then

Begin
LK[j] = S2
LR[j] = reference of S2
i=i+1
j=j+1
End
Else // L[i] is a word, we keep it
LK[j] = L[i]
LR[j] = 0
i=i+1
End
End while
Step3: Return LK.
Step4: Return LR.
End

Stemming is a process for reducing words to their stems. Extract the stem of Arabic token is done by removing affixes. Affixes in our work deal with: suffixes and its prefixes. For example the stem of "شبكات","الشبكات","الشبكة" is "شبكة". Algorithm(2) illustrates the process of remove the prefix addition from the token.

Algorithm(2) : Prefix Algorithm
Input: W: Token, PL**:** Prefix list.
Output: WS: Stem of Arabic Word.

Begin
Step1: Let WS="" ,
Found=false,
i=1,
N=length of PL.
Step2: While i<=N and not(found) do
Begin
If PL[i] is the prefix of W and length of PL[i] < length of W then
Begin
Cut the prefix PL[i] from W to be WS.
If WS is found in SN then
Found=true.
End
i=i+1.
End while
Step3: Return WS**.**
End.

Some token of input text have prefix and suffix addition. Algorithm- 3 illustrates the process of remove the suffix addition with or without prefix addition.

Algorithm(3) : Prefix_Suffix Algorithm
Input: W: Token, PL**:** Prefix list, SL: Suffix list.
Output: WS: Stem of Arabic Word.
Begin **Step1:** Let WS="" ,

```
Found=false,
i=1,
N=size of SL,
M=size of PL.
Step2: While i<=N and not(found) do
  Begin
2.1: If SL[i] is suffix of W and length of SL[i] < length of  W then
      Begin
Cut the suffix SL[i] from W to be WS.
If WS is found in SN then
Found=true.
      End
                  //Check if S1 contain a prefix also
j=1
2.2: While j<M and not(found) do
Begin
              If PL[j] is prefix of WS and length of PL[j] < length of WS then
Begin
Cut the prefix PL[j] from WS to be new WS
If WS if found in SN then
Found= true
j=j+1
End
End
3.3: i=i+1
End
 Step4: Return WS.
End.
```

- Finding Similarity
- The most important step in this paper is to find the similarity between two Arabic texts. After the preprocessing process of text1 and text2, we will find similarity degree between them by using the following three techniques:
- Semantic Similarity,
- Cosine Similarity,
- N2gram word base by using Dice's similarity.

- After extracting the keywords depends on the proposed SemanticNet (that stored the keywords of computer science), each keyword is represented as a node, and these nodes are connected to each other by edges which is represented as semantic relations between nodes. The relation of connection is known as is_a, as illustrated in Figure-3. We use semantic similarity in order to check if two words have the same meaning (belong to the same parent). In this paper, the path length used to measure semantic similarity between two keywords, using edge-counting techniques as given in equation(1) with threshold value 0.70. If the value of similarity is less than 0.70, then the second technique is checked.
- The second technique measure the similarity between two words using cosine similarity. It has been used to count the number of characters that shared between them with threshold value 0.70. If the value of similarity is less than 0.70, then the third technique is checked.
- The third similarity measure technique uses the N-gram to measure the similarity between two words. It is used to count the number of terms the characters are shared between them (as shown in algorithm(4)).

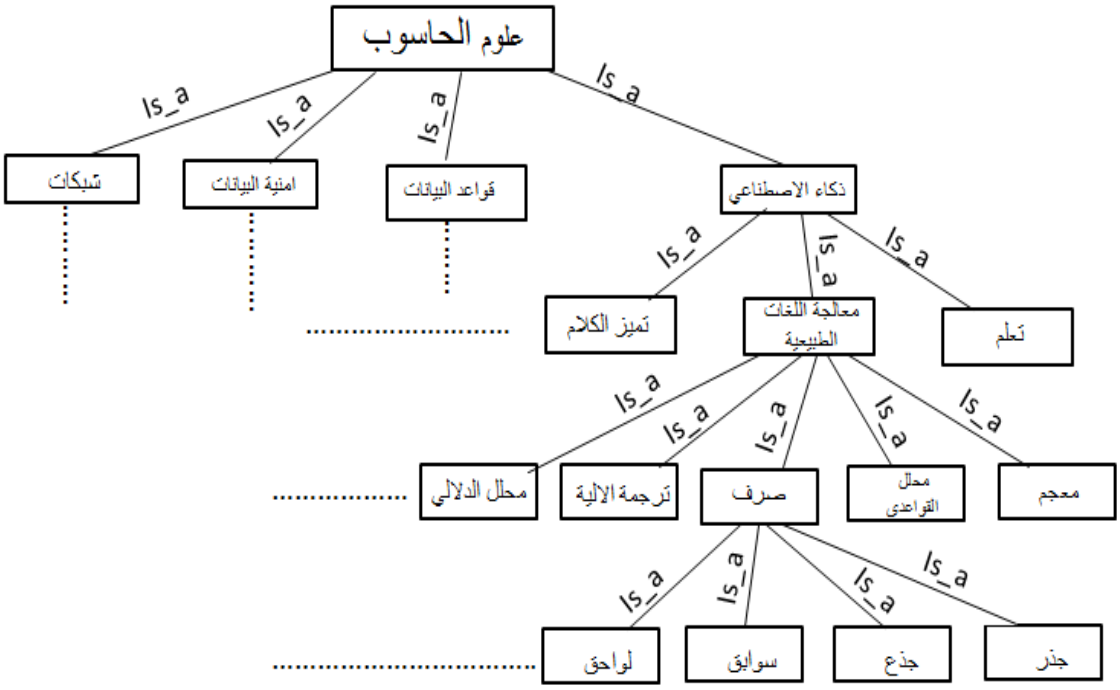

Figure 3- An Example of Keywords that Stored in the Proposed Arabic Semantic Net

Algorithm(4): Proposed System for Finding Similarity between Two Arabic Texts
Input: T1:Arabic Text1 , T2: Arabic Text2.
Output: R: Relevant degree between (T1) and (T2).

Begin
Step1: Preprocessing (T1) and return a list of tokens L1.
Step2: Preprocessing (T2) and return a list of tokens L2.
Step3: Call keyword extraction algorithm (algorithm1) that take (L1) and return a list of keywords (KL1) with list of its paths(LR1),
Compute the length of (KL1) to be (N1).
Step4: Call keyword extraction algorithm (algorithm1) that take (L2) and return a list of keywords (KL2) with list of its paths(LR2),
Compute the length of (KL2) to be (N2).
Step5: Let Sum=0**.**
Step6: For i=1 to N1
For j=1 to N2
Begin
If KL1[i] = KL2[j] then
Sum= Sum + **1**
Else
If **(**LR1[i] != 0) OR **(**LR2[j**]** != 0) then compute the semantic similarity as in equation(1) and put the result to S.
If S >= 0.7 then
Sum=Sum + S
Next j
Else
Compute cosine equation(2) for pair of words as following steps
Begin
Convert token KL1[i] to list of characters C1
Convert token KL2[j] to list of character C2

Taking the union of character in C1 and C2
Converting KL1[i] into frequency of occurrence vectors A
Converting KL2[j] into frequency of occurrence vectors B
Compute the cosine equation=
Then put the result to S2
End
If S2 >= 0.7 then
Sum= Sum + S2
Next j
Else
Compute N2gram for pair of words as the following steps:
Begin
Convert token KL1[i] to list of pair characters N1
Convert token KL2[j] to list of pair characters N2
Count the union of pair characters in N1 and N2 put in C
Counting the number pair characters N1put in A
Counting the number pair characters N2 put in B
Compute Die equation = 2*C / A+B
Then put the result to S3
End
If S3 >= 0.7 then
Sum= Sum + S3
Else
M= max (S1, S2, S3).
Sum= Sum + M
End for
Step7: Av=Sum/(N1*N2**).**
Step8: R=Av*100**.**
Step9: Return**(R).**
End.

Evaluation Metrics [10]:

To evaluate the performance of the proposed system, there are three metrics: Precision, Recall and F-measure.

- The following formula is used to find the precision:

Where
TP (True Positive) = keywords presented and accepted by experts.
FP (False Positive) = keywords presented but not accepted by experts.

- The following formula is used to evaluates the recall:

Where
TP (True Positive) = keywords presented and accepted by experts.
FN (False Negative) = keywords not presented and not accepted by experts.

- Or can be evaluated by the *F-measure* using the following formula:

As illustrated in Figure- 4, the values that are calculated to identify similarity measuring efficiency with criteria of precision, recall and F-measure that have explained previsedin equations (5), (6) and (7).

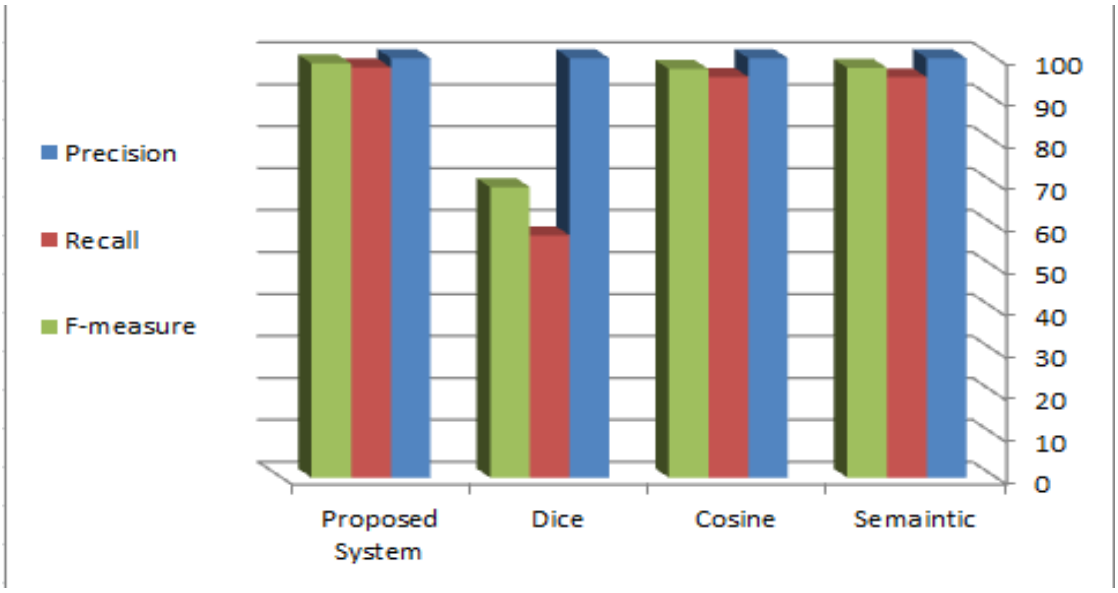

Figure 4- Evaluation of the Similarity Measures

Result and Discussions

- In this work, a new method is developed by building a SemanticNet that storing the Arabic keywords for specific field(computer science), using same concept of WordNet.
- In the proposed system we have used N2gram, because is give best results and the results indicated in this work are that the method of digram offers a better execution than trigram with respect to precision/recall ratios.
- In this work can use jaccard[10] similarity measure instead of the Dice similarity measure, but after testing (using the same texts), it has been appeared that dice similarity return better results.

Conclusion

From the proposed system we conclude the following:

- Using the proposed hybrid technique that combines semantic, cosine, and N2gram similarity techniques produce results in far superior performance than any method alone.
- Using the threshold value in the proposed method and computing the similarity for recurrent words one time will be useful for reducing the problem of time consuming.
- Using the proposed semantic similarity technique is useful to check if two words are belong to same subset, and using the cosine and N2gram similarity measures are useful to count the number of the share characters.
- The proposed system solved the problem of error writing such as repeated the character twice or mistake one or two character or lack of character by using cosine and n2gram similarity measures.

References

1. Wael, H. Gomaa, Aly, A. Fahmy. **2013**. A Survey of Text Similarity Approaches. *International Journal of Computer Applications,* 68(13), pp: 0975 – 8887.
2. Mohammad A. Al-Ramahi , Suleiman H. Mustafa, .**2012**. N-Gram-Based Techniques for Arabic Text Document Matching; Case Study: Courses Accreditation. *Abhath AL-Yarmouk: "Basic Sci. & Eng,* 21(1), pp: 85-105.
3. Jin, F., Yiming, Z. and Trevor, M. **2008**. Sentence Similarity based on Relevance. In Proceedings of PMU'08, pp: 832-839, *Torremolinos (Malaga),* June 22-27**.**
4. Jiang, J. and Conrath, D., **1997.** Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In Proceedings of International Conference Research on Computational Linguistics , pp: 19-33, Taiwan.
5. Shalini, P., **2011**. A Fuzzy Similarity Based Concept Mining Model for Text Classification**.** *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2(11), pp: 115-121.

6. Chakraborty, Rakhi , **2013**. Domain Keyword Extraction Technique: A New Weighting Method Based On Frequency Analysis**.** *Computer Science & Information Technology*.vol pp:109–118.
7. Simmons, Robert F., **1972**. Semantic networks: their computation and use for understanding English sentences. *IEEE.* Second edition. John Wiley & Sons.
8. **[8]** Dao, Thanh Ngoc. **2005**. An improvement on capturing similarity between strings [online]". *Available from:http://www.codeproject.com/KB/recipes/improvestringsimilarity.aspx [cited 2008-06-03]*.
9. Satya, sree K.P.N.V. , Murthy Dr.J V. R. **2012.** Clustering based on Cosine Similarity Measure. *International Journal of Engineering Science & Advanced Technology*, 2(3), pp: 508 – 512
10. Singthongchai, Jatsada and Niwattanakul, Suphakit **. 2013.** A Method for Measuring Keywords Similarity by Applying Jaccard's, N-Gram and Vector Space**".** *Lecture Notes on Information Theory*, 1(4).