Journal of Al-Nahrain University Science **Vol.١٧ (٤), December, ٢٠١٤, pp.٢٠٣-٢١٢**

Text Similarity Based on English Morphological Analyzer Approach

Abeer K. Al-Mashhadany[١*], Sura M. Ali[**] and Sawsan K. Thamer[*]
[*]Department of Computer Science, College of Science, Al-Nahrain University, Baghdad-Iraq.
[**] Department of Software Eng., Baghdad College of Economic Sciences University.
[*] E-mail: aabeeeeraa@yahoo.com.

Abstract

Nowadays many applications require text similarity. It becomes important for comparing texts on websites. Keywords are useful for a variety of purposes, including summarizing, indexing, labeling, information retrieval, text similarity, clustering, and searching. The objective of the proposed systemis achieving automatic test for text similarity and compute similarity ratio. The system based on several techniques especially English Morphological Analyzer (EMA). In this work, keyword extraction and text summarization are very useful to determine text similarity for long and very long texts. The proposed system solves the problem of text similarity through applying several statistics and linguistic approaches especially based on morphological rules. The linguistic approaches in this system also include synonym, word-frequencies, word position, and Part-Of-Speech (POS). It will be shown that keyword extraction and text summarization that are built on EMA approach and other statistics and linguistic approaches are very useful in building high accurate method for text similarity. The system was tested and the accuracy rates of results bounded from ٪٩٨,٨٥ to ٪١٠٠.

Keywords: words frequencies, EMA Approach, Keywords Extraction, text summarization, Text Similarity.

Introduction

The need for new software to solve the problems of texts became necessary. That is because the variety of purposes for which the computer and websites have been used. As one of such problems is text similarity. Text similarity is a common and basic issue to consider in many fields [١]. There are a growing number of tasks that require computing the similarity between texts. It is important to find an appropriate approach for comparing texts. It could be useful for checking answers' rightness, especially for questions that have long answers. Also it is useful for comparing texts on websites.

Similarity is a fundamental concept in the representation of vague knowledge and approximate reasoning. Goodman suggested that two objects "a and b are more alike than c and d if the cumulative importance of the properties shared by a and b is greater than that of the properties shared by c and d".

Many methods are used to calculate similarity. The traditional bag-of-words approaches treattext as unordered words and do not understand the grammatical roles of words, such as subjects or objects, or the part-of-speech roles of words, such as nouns or verbs. In Simfinder approach; different primitives (such as "words that are nouns" or "words that are verbs") are identified. Similarity is computed over all of these features. For two sentences, Simfinder will compute how similar those sentences are based on each feature, and it combines all the similarities into a single final similarity value representing the overall similarity of the two sentences [٤].

Text clustering technology has appeared for a long time. Many models such as *VSM* (*Vector Space Model*), DBScan (*Density-Based Scan*) and *SOM* (*Self-Organizing Map*) have been researched and improved repeatedly. It is a key step to calculate similarities, or distances, amount texts [٥]. Asymmetric word similarities could be used as a tool for automatically computing similarities between words on the basis of their contexts [٢].

Many methods are used for measuring similarity between short segments of texts. These measures include simple lexical

matching, stemming, and text representations, kernel function for semantic similarity [٦, ٧].

This work develops a method for text similarity. It based on previously improved techniques by Ahmad and Abeer (٢٠١٢) [٨] and Abeer (٢٠١٤) [٩]. It depends on EMA, keyword extraction, key phrase extraction [٨], and text summarization [٩].

The developed method is summarized by applying the English Morphological Analysis on the text. Then it applies statistics and linguistics approaches such as; synonym, word position, and *Part-Of-Speech* (*POS*), then it computes words frequencies. For short texts, frequency is enough to compare two texts and compute the ratio of similarity between them. For long text, the method uses additional linguistic approach. It applies keywords extraction then computes the similarity. While for very long texts, it applies text summarization then computes the similarity.

Related Works

Wei Li et al. (٢٠٠٥) [١٠] propose the *Critical Sentence Vector Model (CSVM)*, a novel model to measure text similarity. The CSVM accounts for the structural and semantic information of the document. Compared to existing methods based on keyword vector, e.g. *Vector Space Model (VSM)*, CSVM measures documents similarity by measuring similarity between critical sentence vectors extracted from documents.

James Lewis et al. (٢٠٠٦) [١١] create and optimize a new, hybrid search system for Medline that takes natural text as input and then delivers results with high precision and recall. The combination of a fast, low-sensitivity weighted keyword-based first pass algorithm to cast a wide net to gather an initial set of literature, followed by a unique sentence-alignment based similarity algorithm to rank order those results was developed that is sensitive, fast and easy to use. Several text similarity search algorithms, both standard and novel, were implemented and tested in order to determine which obtained the best results in information retrieval exercises.

A. Islam and D. Inkpen (٢٠٠٨) [١٢] present a method for measuring the semantic similarity of texts using a corpus-based measure of semantic word similarity and a normalized and modified version of the *Longest Common Subsequence(LCS)* string matching algorithm. Existing methods for computing text similarity have focused mainly on either large documents or individual words. We focus on computing the similarity between two sentences or two short paragraphs. The proposed method can be exploited in a variety of applications involving textual knowledge representation and knowledge discovery. Evaluation results on two different data sets show that the method outperforms several competing methods.

Lian Li, Ai Hong Zhu and Tao Su (٢٠١١) [١٣] give an improved text similarity calculation algorithm. The traditional text similarity calculation algorithm may lead to inaccurate results, because it does not consider the effect of same feature words between texts. This problem is solved in this work. Considering that the amount of same feature words reflects two texts' similarity in some extent, the improved algorithm adds in the coverage measured parameter, which effectively reduces the interference of texts with lower similarity. The simulation and experimental results verify the improved algorithm's correctness and effectiveness.

Andrzej Siemiński (٢٠١٢) [١٤] presents and evaluates an efficient algorithm for measuring semantic similarity of texts. Calculating the level of semantic similarity of texts is a very difficult task and the proposed up to now methods suffer from computational complexity. This substantially limits their application area. The proposed algorithm tries to reduce the problem by merging a computationally efficient statistical approach to text analysis with a semantic component. The semantic properties of text words are extracted from the Word Net lexical database. The approach was tested using Word Nets for two languages: English and Polish.

The Proposed System

The proposed system consists of five main components, see Fig. (١); interface module, morphology processing module, work memory module, keyword extraction module and compute similarity module. All modules will be explained one after another.

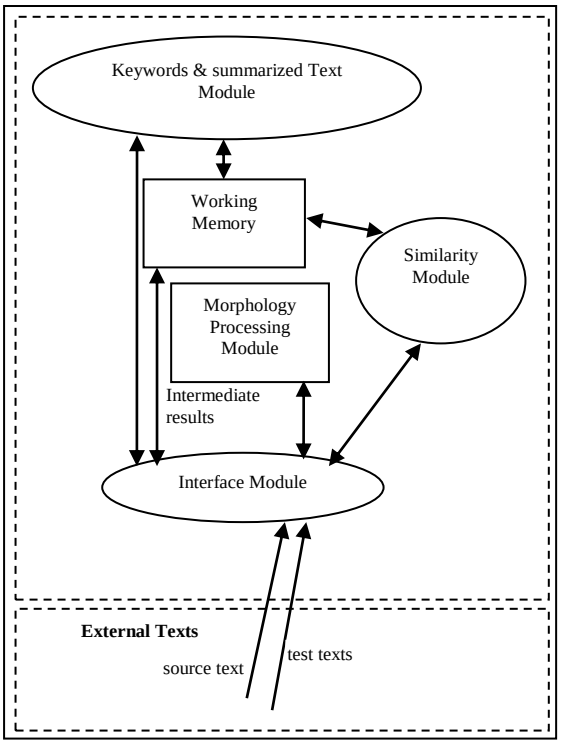

Fig.(١) The proposed system modules.

Interface Module

Name of this module refers to its task. It is the point of interaction between the system and external texts; also it is the control unit of the proposed system. Interface module has connections to all modules inside the system, and it controls their work. Interface module receives number of texts may be two or more. The first one is considered as source text while others are test texts. Each one of test texts must be compared with source text to calculate similarity ratio. In addition to task of interaction; this module performs the main processes of current approach, see Fig.(٢) and Fig.(٣). As shown in Fig.(٢); the processes done by interface module included; preprocessing, asking for word analysis, replace synonym, and constructing intermediate results and storing them in working memory.

Fig.(٣) shows how interface module controls the work of other modules. This module asks other modules to perform a specific task or to provide specific data. To perform each one of main processes tasks, interaction between interface module and morphology processing module is necessary. Preprocessing includes; convert text to lower case, divide text into tokens, and then replace abbreviations. Word analysis means use English morphological rules to find root or stem of word. Interface module asks morphology processing module for word analysis. At last each root or stem must be replaced with its synonym, and this will increase the accuracy rate of this approach. In case of long texts, interface module must connect with keywords and summarized text module to generate keywords. Also in case of very long text the same connection is necessary, but now to summarize text.

As a result of this module, new facts are constructed. This module computes frequency for each word (root) in text. Also, counters for words, sentences, negations and subject pronouns must be computed. Then the new facts with all previous results (intermediate results) must be stored in work memory to be used at other modules.

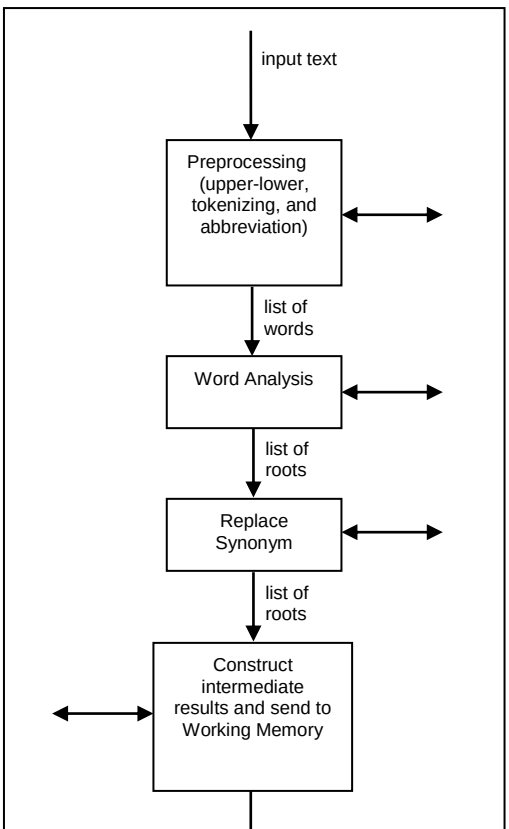

Fig.(٣) The processes done by interface module.

Morphology Processing Module

This module performs the task of word analysis. It receives source word from interface module, analyses the word, and then return its root to the interface module. Mainly this module decomposes into two types of components; dictionaries and morphological rules as shown in Fig. (٤). Dictionaries include; EMA dictionary, stemmer dictionary, abbreviations dictionary, and synonym dictionary. Morphological rules depended on rules that proved at the previous work "Using English Morphological Analyzer to Decrease the Dictionary Size in Keywords Extraction Techniques" (٢٠١٢) [٨]. At that work morphology processing was depends on EMA and stemmer. So, morphological rules in current work include EMA analyzer and English stemmer analyzer.

Algorithm for the main processes at current approach
Input: source text (ST), tested texts (TT).
Output: store facts on working memory.
Begin
Step١: Get text ST.
Step٢: Connect with morphology module to apply preprocessing.
Step٣: Ask morphology module to perform word analysis.
Step٤: Connect with morphology module to apply replace synonym.
Step٥: Construct facts (words frequencies, and counters for; words, sentences, negations and subject pronouns, then store them with other intermediate results in working memory.
Step٦: For each one of tested texts (TT_i) do: Repeat steps ٢, ٣, ٤, and ٥.
Step٧: If text is short then go to Step١٠.
Step٨: If text is long then asks keywords & summarized text module to construct keywords. Then go to Step١٠.
Step٩: If text is very long then:
٩,١: Asks keywords & summarized text module to summarize text.
٩,٢: Recalculate frequencies and store them in

Fig.(٣) Algorithm to follow the main processes at current approach.

EMA dictionary of current work designed to increase accuracy of its performance. It divides stop-words to three types, first type includes negation stop-words, second type includes subject pronouns, and all others will be at third type. This division helps interface module to describe texts clearly.

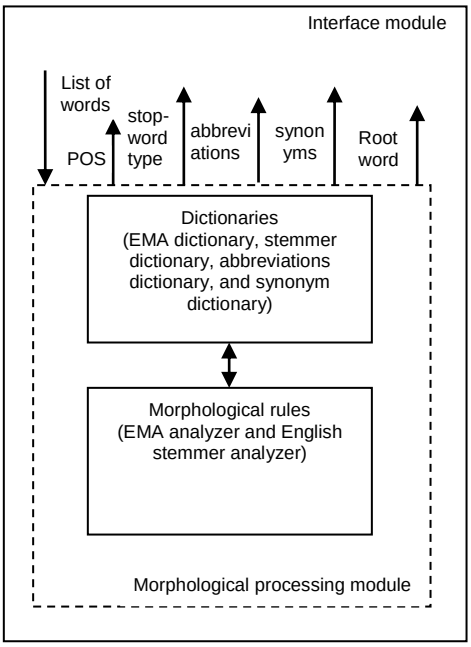

Fig.(٤) Morphological processing module interacts with interface module.

In interface module each one of its tasks needs a specific part of morphology processing module. Preprocessing task needs abbreviation dictionary. Word analysis needs EMA analyzer, EMA rules, EMA dictionary, stemmer analyzer, and stemmer dictionary. Replace synonym needs synonym dictionary. In addition to roots; this module provides

interface module with full description about each word, this will facilitate the construction of intermediate results. As example, this module provides POS, type of stop-word, abbreviations, and synonyms see Fig.(٤).

Working Memory Module

Working memory is the place (databases) in which all intermediate results must be stored. These results are constructed at interface module, and stored here to be used in constructing other intermediate results by other modules, until reaching the last required results.

Intermediate results were stored in working memory as facts. Facts that constructed by interface module interesting with; describing each sentence (its order, negation flag, list of POS, first word position, and last word position), describing each word (position, source, root-or-stem, POS), word frequency, counters (of sentences, negations, and subject pronouns). Then facts of keywords and summarization will be constructed by keywords and summarized text module. At last similarity module constructs facts to calculate the similarity ratio between texts.

Keywords & Summarized Text Module

The proposed system considered a received text as short text, long text, or very long text. In case of short text, the comparison will be implemented between facts that are stored as intermediate results in working memory (results that constructed by interface module).

It is difficult to check similarity between long texts. Because a long text contains unimportant information (noise), which could be different in two texts, while the important information may be the same. Facts that were constructed and stored on working memory are not enough to compare two long texts, but they are enough to extract keywords and key-phrases. So, keywords and key-phrases will be extracted and saved on working memory. Method that was proved at the previous work (٢٠١٢) [٨] is used at current work to extract keywords.

In case of very long text, text summarization will be needed. Text summarization means; reduces text without losing any one of diverse topics of the source text. After constructing keywords, facts at working memory became enough for summarizing text. Method that was proved at a previous work (٢٠١٤) [٩] is used at current work to construct the summarized text, then it will be saved in working memory too. Now interface module will manipulate the summarized text.

Similarity Module

Task of this module is calculating similarity between texts considering intermediate results that found in working memory. The proposed system provides detail description about sentences and about all words in texts. Surly, considering intermediate results will enforce the accuracy rate because the comparison will be closer to natural language understanding. The proposed system compares any two texts depending on their meaning not on the specific words inside texts.

In case of short texts, this module uses facts constructed by interface module (words frequencies, counters and POS) to calculate similarity ratio as described in Fig.(٥). In case of long texts, this module uses keywords and other facts that stored in working memory (counters and POS) to calculate similarity ratio as described in Fig.(٦). In case of very long text, this approach converts very long text to long text as described in Fig.(٣). So the calculation of similarity ratio will be as described in Fig.(٦).

Algorithm for calculating similarity ratio between two short texts

Input: all facts stored on working memory, (facts of source-text (ST), and facts of tested text (TT)).
Output: similarity ratio.
Begin
Step١: create: similarity-negation-counter (sn), similarity-counter (sc), and different-noun-counter (dnc). sn=٠, sc=٠, dnc=٠.
Step٢: for each two sentences if equal negation flag then sn=sn+١.
Step٣: compare all words frequencies in two texts as following:
٣,١: if they are equal then sc=sc+freq.
٣,٢: if they are not equal then add minimum (sc=sc+freqMin), and check POS for the missing word, if it is noun then dnc=dnc+missing.
Step٤: make sure that nc=negation counter of ST.
Step٥: if sc=frequencies of ST then similarity-ratio(sr)=١٠٠, goto End

Fig.(٥) Algorithm for computing similarity ratio of short texts.

Results and Discussions

Dictionary which is limited and dedicated for artificial intelligence domain had been designed and built, so it was called "Artificial Intelligence Ddictionary" (A.I.Dic). Its design was compatible with the structure of dictionaries in Morphological processing module of the proposed system. A.I.Dic is necessary to test the proposed system. Also the proposed system needs many real texts with variety longs to fallow its behavior and test its results. It is a good idea to achieve many samples of real texts from a real examination. So the right answer will be the source texts, and answers of students will be the tested texts. The similarity ratio will be compared with the real degree of the student to test the truthful of the proposed system.

Algorithm for calculating similarity ratio between two long texts

Input: all facts stored on working memory, (facts of source-text (ST), and facts of tested text (TT)).

Output: similarity ratio.

Begin

Step١: compute similarity-negation (sn) using negation counters (nc) of ST and TT:

If ncTT>=ncST then

sn=٢٥

else

sn=٢٥-(abs(ncST-ncTT))*(٢٥/ncST)

Step٢: create: source-frequency-counter (freq١), tested-frequency-counter (freq٢), and different-noun-counter (dnc). freq١=٠, freq٢=٠, dnc=٠.

Step٣: for each keyword at ST, add its frequency (sf) to counter, freq١=freq١+sf. Then do:

٣,١: if found the keyword at TT with the same frequency (tf) or more than sf; sf<= tf, then add to counter, freq٢=freq٢+sf.

٣,٢: else if tf<sf then check POS for the keyword. If it is noun then add the different to dnc, dnc=dnc+(sf-tf).

Step٤: if tf<sf and subject pronouns counter of ST (c١) less than subject pronounscounter of

Fig.(٦) Algorithm for computing similarity ratio of long texts.

The final examination of "Multi-Agent systems" course-٢٠١٣ at department of computer science/ Al-Nahrain University was chosen. Testing needs three types of source texts; short, long, and very long. For each type twenty samples of texts had been checked using the proposed system. In other word, for each type check the similarity of twenty students' answers with the right answer that is provided by the lecturer. The variety of tested samples was necessary, to achieve real texts with variety degrees of similarities, in order to ensure the performance of the proposed system and ensure the truthful of its results. So, sixty samples of real texts were used to perform the process of testing the proposed system.

Table (١) shows results of applying the proposed system on the twenty short real texts samples. As shown in table ١ the accuracy rate that achieved by this system for short texts was ٪٩٨,٨٥. The proposed system used root frequencies to compute the similarity ratio for short texts. As a trying to increase the accuracy rate for short texts, the proposed system went forward on its method and applied key-phrases extraction on samples of short texts, then it achieved accuracy rate ٪١٠٠.

Table (٢) shows results of applying the proposed system on the twenty long real texts samples. As shown in table ٢ the accuracy rate that achieved by this system for long texts was

٪٩٩,٤٥. The proposed system used keywords to compute the similarity ratio for long texts. It was clear that the accuracy rate of using keywords is higher than using root frequencies although the texts were longer in the second case.

Table (٣) shows results of applying the proposed system on the twenty very long real texts samples. As shown in table ٣ the accuracy rate that achieved by this system for very long texts was ٪٩٩,٣٥. First, the proposed system used keywords to compute the similarity ratio for very long texts. Then it went forward in its method and applied text summarization on the very long samples. And then it applied the keywords method on the summarized texts. It achieved high accuracy rate and it was ٪١٠٠.

The three previous tables show that see Fig.(٧) accuracy rate of current system starts at ٪٩٨,٨٥ for short texts, when the comparison dependent on frequencies, so it was the minimum accuracy. Then using key-phrases force the accuracy to ٪١٠٠. Also using text summarization can force the accuracy to ٪١٠٠.

Really all previous methods are built on frequencies; keywords and key-phrases extraction is built on frequencies and other linguistic approaches such as word position, and POS, and so on; text summarization is built on keywords and other linguistic approaches such as; title, first paragraph, and POS. So high accuracy of frequencies gives methods that are built in the proposed system high accuracy rates reach to ٪١٠٠.

Table (١)
Results of applying current approach on short texts.

Sample-No	*Mark /١٠٠*	*similarity results (/١٠٠) using frequencies*			*similarity results (/١٠٠) using key-phrases*		
		Similarity	*Accuracy rate*	*Error rate*	*Similarity*	*Accuracy rate*	*Error rate*
١	٠	٠	١٠٠	٠	٠	١٠٠	٠
٢	٠	٠	١٠٠	٠	٠	١٠٠	٠
٣	٠	٠	١٠٠	٠	٠	١٠٠	٠
٤	٠	٠	١٠٠	٠	٠	١٠٠	٠
٥	٠	١٥	٨٥	١٥	٠	١٠٠	٠
٦	٠	٨	٩٢	٨	٠	١٠٠	٠
٧	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
٨	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
٩	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
١٠	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
١١	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
١٢	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
١٣	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
١٤	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
١٥	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
١٦	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
١٧	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
١٨	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
١٩	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
٢٠	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
Average			٩٨,٨٥	١,١٥		١٠٠	٠

Table (٢)
Results of applying current approach on long texts.

Sample-No	Mark /١٠٠	Similarity results (/١٠٠) using keywords	Accuracy rate/١٠٠	Error rate/ ١٠٠
١	٠	٠	١٠٠	٠
٢	٠	٠	١٠٠	٠
٣	٠	٠	١٠٠	٠
٤	٠	٠	١٠٠	٠
٥	٠	٠	١٠٠	٠
٦	٠	٠	١٠٠	٠
٧	٠	٠	١٠٠	٠
٨	٥٠	٥٠	١٠٠	٠
٩	٥٠	٥٠	١٠٠	٠
١٠	٥٠	٥٠	١٠٠	٠
١١	٥٠	٥٠	١٠٠	٠
١٢	٥٠	٥٠	١٠٠	٠
١٣	٥٠	٥٤	٩٦	٤
١٤	٣٥	٣٧	٩٨	٢
١٥	٨٨	٨٣	٩٥	٥
١٦	٤٠	٤٠	١٠٠	٠
١٧	٨٥	٨٥	١٠٠	٠
١٨	١٠٠	١٠٠	١٠٠	٠

١٩	١٠٠	١٠٠	١٠٠	٠
٢٠	١٠٠	١٠٠	١٠٠	٠
Average			٩٩,٤٥	٠,٥٥

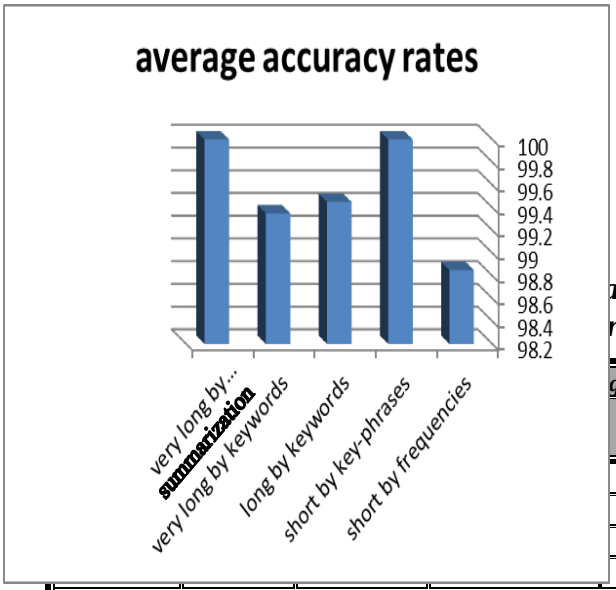

Fig.(٧) Ranges of accuracy rates for the proposed system.

ıble (٣)
nt approach on very long texts.

				g keywords	*similarity results (/١٠٠) after summarization*		
				Error rate/١٠٠	*Similarity/ ١٠٠*	*Accuracy rate/١٠٠*	*Error rate/١٠٠*
				٠	٠	١٠٠	٠
				٥	٠	١٠٠	٠
				٠	١٠	١٠٠	٠
				٠	١٠	١٠٠	٠
٥	٢٠	٢٠	١٠٠	٠	٢٠	١٠٠	٠
٦	٢٠	٢٠	١٠٠	٠	٢٠	١٠٠	٠
٧	٣٠	٣٠	١٠٠	٠	٣٠	١٠٠	٠
٨	٣٠	٣٠	١٠٠	٠	٣٠	١٠٠	٠
٩	٤٠	٤٠	١٠٠	٠	٤٠	١٠٠	٠
١٠	٥٠	٥٠	١٠٠	٠	٥٠	١٠٠	٠
١١	٥٠	٥٨	٩٢	٨	٥٠	١٠٠	٠
١٢	٠	٠	١٠٠	٠	٠	١٠٠	٠
١٣	٦٠	٦٠	١٠٠	٠	٦٠	١٠٠	٠
١٤	٦٠	٦٠	١٠٠	٠	٦٠	١٠٠	٠
١٥	٦٠	٦٠	١٠٠	٠	٦٠	١٠٠	٠
١٦	٧٠	٧٠	١٠٠	٠	٧٠	١٠٠	٠
١٧	٧٠	٧٠	١٠٠	٠	٧٠	١٠٠	٠
١٨	٧٠	٧٠	١٠٠	٠	٧٠	١٠٠	٠
١٩	٨٠	٨٠	١٠٠	٠	٨٠	١٠٠	٠
٢٠	١٠٠	١٠٠	١٠٠	٠	١٠٠	١٠٠	٠
Average			٩٩,٣٥	٠,٦٥		١٠٠	٠

The proposed system achieves the high accuracy of frequencies because of using linguistic approaches that force frequencies extraction. Firstly it uses EMA approach which based on root-stem analyzer. The second step that forces frequencies is replacing-synonyms, which means that frequency depends on meaning not only on English morphological rules. And before all processes, abbreviations are checked and replaced with full terms. The proposed system has all advantages of EMA approach; more flexibility, more accuracy and reducing dictionary size.

Structure of EMA dictionary in the proposed system has many details that make current method more flexible and closed to natural language understanding. Current EMA dictionary divides stop words into three types. It isolates stop words that refer to negation such that (not, no), then isolates stop words that refer to noun; subject pronouns such as (it, he, they), and put other stop word in one file. It is true that stop word could be neglected, but current system decided to make useful of their meaning, because meaning is necessary in computing similarity. Current system recognizes all negation tools in the received

text. Also current system recognizes that apronoun is used instead of a specific noun.

The proposed system provides high efficiency method that computes the similarity between two texts. It is more flexible because of studying all cases of texts (short, long, very long). It provides appropriate manipulation for each type of texts.

Conclusions

١. Current system successful in building text similarity with high efficiency, and computing similarity rate between any two texts.

٢. Current approach achieved high accuracy rate (٪٩٨,٨٥) for short texts when depending only on frequencies. Then increase the accuracy rate to (٪١٠٠) when depending on key-phrases.

٣. Current approach achieved accuracy rate (٪٩٩,٤٥) for long texts depending on keywords.

٤. Current approach achieved high accuracy rate (٪٩٩,٣٥) for very long texts when depending on keywords. Then increase accuracy rate to (٪١٠٠) when depending on text summarization then extracts keywords.

References

[١] `Yi Feng, "An Order-Based Taxonomy for Text Similarity", Procedings of the CICA ٢٠١١, Springer, Vol. ١٠٧, ١٦١٧-١٦٢٣, ٢٠١٢.
http://link.springer.com/chapter/١٠,١٠٠٧٪٢F٩٧٨-٩٤-٠٠٧-١٨٣٩-٥_١٧٤#page-١

[٢] Martin T. P. and Azmi-Murad M., "An Incremental Algorithm to find Asymmetric Word Similarities for Fuzzy Text Mining", Soft Computing as Transdisciplinary Science and Technology Advances in Soft Computing, Springer Berlin Heidelberg, Vol. ٢٩, ٨٣٨-٨٤٧, ٢٠٠٥.
http://www.ist-eperspace.org/Presentations/Martin_An_incremental_algorithm.pdf

[٣] Vasileios H., Judith L. Klavans, and Eleazar E., "Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning", Proceedings of the ١٩٩٩ Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, ٢٠٣-٢١٢, ١٩٩٩.
http://www.aclweb.org/anthology-new/W/W٩٩/W٩٩-٠٦٢٥.pdf

[٤] David K. Evans, "Identifying Similarity in Text: Multi-Lingual Analysis for Summarization", PhD. Thesis, Graduate School of Arts and Sciences, Columbia University, ٢٠٠٥.
http://www١.cs.columbia.edu/nlp/theses/dave_evans.pdf

[٥] Zuoguo L., and Xiaorong C., "A Graph-Based Text Similarity Algorithm", National Conference on Information Technology and Computer Science (CITCS ٢٠١٢), Atlantis Press, ٦١٤-٦١٧, ٢٠١٢.

[٦] Donald M., Susan D. and Ctopher M., "Similarity Measures for Short Segments of Text", Proceeding ECIR'٠٧ Proceedings of the ٢٩th European conference on IR research, Springer-Verlag Berlin, ١٦-٢٧, ٢٠٠٧.
http://research.microsoft.com/en-us/um/people/sdumais/ecir٠٧-metzlerdumaismeek-final.pdf

[٧] Mehran S., and Timothy D. Heilman, "A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets", Proceedings of the ١٥th international conference on World Wide Web, ٣٧٧-٣٨٦, ٢٠٠٦.
http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en//intl/en/pubs/archive/٣٢.pdf.

[٨] Ahmed T. and Abeer K., "Using English Morphological Analyzer to Decrease the Dictionary Size in Keywords Extraction Techniques", International Journal of Research and Reviews in Soft and Intelligent Computing (IJRRSIC), Vol. ٢, No. ١, March,١١٤-١١٨, ٢٠١٢.

[٩] Abeer K., "Text Summarization Based on Several Natural Language Techniques", Eng. & Tec. Journal /University of Technology, Vol.٣٢, Part (B), No.٢, ٢٠١٤.

[١٠] Wei Li, Kam-Fai W., Chunfa Y., Wenjie Li, and Yunqing X., "Improving Text Similarity Measurement by Critical Sentence Vector Model", Information retrieval technology, Springer, Volume ٣٦٨٩, ٥٢٢-٥٢٧, ٢٠٠٥.

بين النصوص الطويلة و الطويلة جدا. النظام المقدم يحل مشكلة مماثلة النص من خلال تطبيق بضعة طرق لغوية واحصائية تعتمد بالاخص على قواعد الصرف. الطرق اللغوية في هذا النظام ايضا تتضمن المرادفات، تكرار الكلمات، مواقع الكلمات، ومقاطع الكلام. سنرى من خلال البحث انه الكلمات المفتاحية وتلخيص النص المبنية على طريقة المحلل الصرفي الانكليزي وطرق احصائية ولغوية اخرى مفيد جدا في بناء طريقة ذات دقة عالية لاختبار مماثلة النص. تم اختبار النظام وكانت نسب دقة النتائج تتراوح بين ٩٨,٨٥% الى ١٠٠%.

http://link.springer.com/chapter/١٠,١٠٠٧٪٢F١١٥٦٢٣٨٢_٤٤#page-١

[١١] James L., Stephan O., Justin H., Mounir E. and Harold R., "Text similarity: an alternative way to search MEDLINE", Bioinformatics Original Paper, Vol. ٢٢, No. ١٨, ٢٢٩٨–٢٣٠٤, ٢٠٠٦.
http://scholar.google.com/scholar_url?hl=ar&q=http://citeseerx.ist.psu.edu/viewdoc/download٪٣Fdoi٪٣D١٠,١,١,١٠٥,١٤٩٪٢٦rep٪٣Drep١٪٢٦type٪٣Dpdf&sa=X&scisig=AAGBfm٢pvadcv٧O٠geXZmAdcbNWHWIFDPQ&oi=scholarr

[١٢] Aminul I. And Diana I., "Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity", ACM Transactions on Knowledge Discovery from Data, Vol. ٢, No. ٢, Article ١٠, ١-٢٥, ٢٠٠٨.
http://pdf.aminer.org/٠٠٠/٣٦٩/٤٣٩/classification_of_rss_formatted_documents_using_full_text_similarity_measures.pdf

[١٣] Lian Li, Ai Hong Zhu, and Tao Su, "An Improved Text Similarity Calculation Algorithm Based on VSM", Advanced Materials Research /Trans Tech Publications.Volx.٢٢٥-٢٢٦, ١١٠٥-١١٠٨, ٢٠١١.
http://www.scientific.net/AMR.٢٢٥-٢٢٦,١١٠٥

[١٤] Andrzej S., "Fast Algorithm for Assessing Semantic Similarity of Texts", International Journal of Intelligent Information and Database Systems, Inderscience Publishers, Vol. ٦, No. ٥, ٤٩٥-٥١٢, ٢٠١٢.
http://inderscience.metapress.com/content/f٣v٤٥r٦٧٤٤v٧٠j٦٥/

الخلاصة

هذه الايام العديد من التطبيقات تتطلب مماثلة النص. اصبحت مهمة في مقارنة النصوص على مواقع شبكة الانترنت. الكلمات المفتاحية مفيدة لاغراض مختلفة، تتضمن التلخيص، الترميز، استرجاع المعلومات، تشابه النص، التصنيف، والبحث. الهدف من النظام المقدم هو الحصول على اختبار تلقائي لمماثلة النص واحتساب نسبة المماثلة. النظام يعتمد عل بضعة تقنيات وبالاخص تعتمد على المحلل الصرفي للغة الانكليزية. في هذا البحث، استخراج الكلمات المفتاحية وتلخيص النصوص مفيدة جدا لتحديد مماثلة النص