Text Similarity Based on Modified LSA Technique
Ahmed T. Sadiq, Khudhair J. Kadhim
Computer Science Department, University of Technology
Computer Science Department, Mustansaryiah University
drahmaed_tark@yahoo.com, khudhair_jassim78@yahoo.com

ISSN: 2231-8852

Abstract
The most applications or practice areas which are interrelated with text mining such as information retriever (IR), clustering, text summarization, automatic answer grading, machine translation and automatic essay scoring and other. All of them are depends on how find similarity distance between a pair or more of texts as a major process. This paper proposes two approaches are focuses on the problem of the semantic similarities between texts in English language by using Latent Semantic Analysis (LSA) technique. It's trying to enhance the process of finding the semantic similarity distance between texts and making it more adaptable for both long documents and short sentences. The two proposed approaches are using the same style that used in Knowledge-based measures, where derived semantic relationship on terms level from the semantic space, and thus calculate the similarity between the two texts fully. Evaluation results on three different data sets show that the two approaches gives good results comparison to human judgment equal to 76% and outperforms several competing methods which are used for detecting Plagiarism in texts, where the proposed system achieves 92%.

Keywords: Text Similarity, Latent Semantic Similarity, Semantic Similarity, Corpus-Based Similarity.

1- Introduction
The measure of similarity between texts is one of the critical tasks and have high-impact in many applications which give interest to texts similarity or text-related, such as: text summarization, information retrieval, clustering, text classification, machine translation etc. Text similarity process is not a simple task, due to variability in using natural language expressions, for example, the expression to the same thing in different ways. The techniques of text similarity measures are used early in several NLP applications and the fields associated with text mining. These techniques takes two texts as input and automatically returns numeric score that quantifies how much are similar based on their contents.

The traditional techniques for detecting similarity between texts are based on simple lexical matching method, it focus on analyzing shared words (co-occurrence) between them. This approach can find the similarity degree between texts from the number of the words which are matched in lexical form and occur in both texts. These techniques measures are simple in implement, but they are weak in reflecting the relatedness between words having similar meanings whether to have the same root form or synonym. Many tasks in NLP often need to determine the semantic similarity or semantic relation between words. By the concept of semantic similarity can be allocated set of different words into similarities lists depending on the their meaning. Many numbers of semantic similarity measures that were previously proposed, approach based on using
external data sources (dictionary) or derived information from either semantic networks such as WordNet, these techniques are called Knowledge-Based similarity or from words distribution in collection of texts, these techniques are called corpus-based, and hybrid technique. In this paper, we present two approaches for measuring the semantic similarity of texts (document, sentence) based on Latent Semantic Analysis (LSA). LSA is a method for capturing and analyzing the similarity of words and text passages by statistical computations applied to a large corpus of text (Landauer & Dumais, 1997), it one of corpus-based measure. LSA theory is based on the aggregate of all contexts words, These contexts in turn, are explains when a given word is appeared or not, thus available set of mutual restrictions, from which LSA can determine the similarity between words (Landauer & Dumais, 1997). The most widely use of the LSA technique in information retrieval applications (IR), where it's measure the similarity between user queries and corpus documents only. The LSA technique applies a set of mathematical algorithms on texts corpus, these algorithms are comprises: set of pre-processed algorithms, terms weights algorithm, Singular Value Decomposition (SVD) algorithm and measure similarity algorithms (Cosma, 2008). The rest of this paper is organized as follows. Section 2 describes the texts similarity. Section 3 explores the related work. Section 4 describes our approaches in detail. Section 5 shows the results and Section 6 shows conclude.

2- Text Similarity

The assessment of similarities between documents which are computer-based a key problem in the organization and analysis process on large documents repositories, that is one of great challenges to machine learning (Hofmann, 1999). All application that combine the artificial intelligence (AI) and NLP appear increasingly important role for measure the text similarity, and calculation the similarity between a pair of documents is represents a major burden (Gomaa & Fahmy, 2013). Similarity measure is a function that returns a real number to a pair of documents (often between 0 and 1) represent the degree of similarity or distance between them. When distance value is equal to one means that two texts are identical whereas a value is zero the two texts are dissimilar. Mainly, text can be similar in two ways lexically and semantically.

The lexically model is use traditional matching for measure the distance between two texts. Increasing the similarity of two texts when are have a similar character sequence. While in semantic similarity, texts are similar if they have the same thing, used in the same way, used in the same context or one is a type of another. It can take advantage of the semantic analysis for texts and knowledge representations, which support their meaning or describe their nature to mimic human ability to compare texts. The semantic relatedness or semantic similarity is a distance measure over a set of words or texts and it determined based on their semantic content or similarity in their meaning, the semantic relatedness concept is more general from semantic similarity, where the similarity is more specifically and represented the special case from relatedness, the meronymy and antonymy relation can find in semantic relatedness while the semantic similarity not include them. The measure of similarities between texts has important role in many applications which belong to NLP and areas interrelated with text mining such as information retrieval (IR), clustering, documents classification, web mining, information extraction and concept extraction, in software engineering (SE) field through program analysis to support understanding software Andrian Marcus (Marcus, 2004), and biomedical - Gene Ontology (GO) using similarity degree between genes and proteins (Couto, 2003).
3- Related Works

The importance of measuring similarity between texts increases over time, therefore, many researchers started to investigate and several approaches were introduced. There is extensive literature on measuring the similarity between texts, where varying between measure on long texts (documents) (Hatzivassiloglou et al. 1999; Landauer & Dumais 1997) and short texts (sentences) (Foltz et al. 1998). From the point of view the size of text, while for the techniques contain many methods lexical, semantic, hybrid between them and hybrid between semantic similarity techniques itself.

The semantic text similarity method (STS) in Islam & Inkpen (2008) proposed a hybrid method that derives similarity degree between short texts from syntactic and semantic information. The authors suggested using the word order a string similarity as a lexical techniques, by combining three different results for string similarity, where it used three versions from longest common subsequence (LCS) method with some modification and then calculate a weighted sum for them. By using corpus-based measures as a semantic techniques, simple measure Pointwise Mutual Information-Information Retrieval (PMI-IR) to determine the semantic similarity degree of words. Both similarity (semantic, string) They are calculated at first, and the similarity measuring of word order in the two texts is optional to make his method more generic (Islam & Inkpen, 2008).

Mihalcea & Corley (2006) used hybrid methods which mixed two approaches of measures similarity knowledge-based and corpus-based to find the semantic similarity degree between short texts, use two corpus-based measures, PMIIR and LSA (Latent Semantic Analysis) and six knowledge-based measures to measure the word-to-word similarity and to find the word weight via using the inverse document frequency function (idf ). The proposed define for each word appear in text1 the corresponding word in text 2 which has highest degree of semantic similarity (max-Similar(w,T2)) using one of the word-to-word similarity methods, repeat the same process on words in text2 with text1. The score of words similarities are then weighted. Finally, the resulting similarity scores are combined using a simple average (Mihalcea & Corley, 2006).

Bollegala et al. (2007) proposed method is exploit the search engine to providing information on the web uses in semantic similarity measure between words. This method is exploits the what are returned by search engine, it using text snippets from web, and used a traditional measure the page counts. Two words are given to define similarity score between (P and Q), and then apply two types of queries on them to obtain the page counts are returned from (P OR Q) query, (P AND Q)query, the result will using with degree of semantic similarity which calculates from text snippets. And integrated these different degree of similarity to find the optimal similarity by combination using support vector machines.

In Aggarwal et al. (2012) They implemented System semantic textual similarity (STS) by two approaches to calculate degree of similarity between two sentences. First approach it is referred as TunedESA (combines the corpus-based with knowledge-based ) Which includes A-Calculate the Explicit Semantic Analysis (ESA) relatedness score between the whole sentences, B- Calculate degrees of semantic similarity to words which are falling under the same syntactic roles (subjects, actions and objects) in both the sentences using WordNet- Lin measure C- final score combine from these four scores. The second approach is based on the WordNet based Lin measure between the whole sentences with any modification.

Gomaa & Fahmy (2012) presented the Automatic Scoring (AS) system that is using the texts similarity to evaluate student’s answer. The proposed model calculates the similarity degree between each word that appears in typical answer with all student’s answer words. This model
suggests combining two measures of similarity, the lexical-basis with a corpus-based to find the automatic score. It starts with string sequences (answers) to find lexical similarity via using term-based measures. Then it tries to define the semantic similarity to words in both the typical answer and each student’s answer via using corpus-based similarity. Where AS system is constructing the similarity matrix where the words of model answer are represented by rows, while columns represents student’s answer words, last two columns in similarity matrix contain the average and max similarity degree between each words in student’s answer and model answer, then to find average for them (Max, Average). Final student’s mark is defined by combining obtained values from both similarity methods.

4. Proposed System

The proposed system determines the similarity of two texts from semantic information. It suggests two approaches, one uses cosine similarity to determine the degree of similarity with no regard to the length of the text, and the other approach uses typical average by taking the length of texts in to account. The proposed system exploits the mathematical properties of SVD especially the terms matrix in semantic space to find the distance between two texts. The two approaches are sharing the same principle to find the semantic similarity for the different words between texts from terms semantic space that generated from SVD. The main difference between the two approaches is the method of calculating the final similarity of texts. In the first approach a semantic vector is created for each text from

- The weights words that appear in both text without change.
- The words that appear in only one of the two texts, The max of semantic similarity is calculated and multiplied by the weight of corresponding word which has max of semantic similarity.
- Final score of the similarity is computed by using the cosine similarity function between the semantic vectors. This approach doesn't care about the length of texts.

In the second approach, only one semantic vector is created from

- The value equal 1 for words that appear in both text depending on the words index.
- The words that appear in only one of the two texts, The max of semantic similarity is calculated and multiplied by the average weights of corresponding words which has the max of semantic similarity.
- Final score of the similarity is computed from the average of sum values of semantic vector divided on average length of the two texts. This approach takes into account the length of the two texts as shown in equation (1).

\[
SIM (T1, T2) = \frac{\sum_{i=1}^{n} sem \_ v(wi)}{(L(T1) + L(T2) / 2)} ...(1)
\]

The general proposed system is depicted in Figure (1). The figure illustrates that there are two phases. The first phase creates semantic space from corpus. Usually this has to be done only once, or when significant changes to the corpus. The second phase is analysing and calculating semantic similarity of the two texts, that need to finding the similarity score. Create semantic space it's the main part for all corpus-based semantic similarity measures or the methods that use statistical computations. The corpus passes in a set of successive operations (pre-processing the stop word removing and stemming uses the Porter stemming algorithm, creating the terms-documents matrix (TDM), create weight matrix by using (TF×IDF) and apply the SVD algorithm) to extract semantic
space, which will be used in next phase. The phase of analysing and calculating semantic similarity receives two texts as inputs which is required to measure the distance between them and re-apply the same processes which are applied on corpus except the SVD algorithm to formation three vectors, two vectors one for each text used in the first approach and the third average vector used in the second approach. Eventually the system through the two phases is application for improved LSA technique to measure semantic similarity between texts which are input to system.

The proposed approach tries to improve the performance of measuring semantic similarity process, it makes the vector length that is used to represent text depends on the texts lengths instead of depending on the size of semantic space, which may be a multiplication of the real size. The proposed approach looks in the first text to any term that does not appear in the second text, when the system finds this case it begins to calculate the semantic similarity for each term in the second text by uses cosine similarity function between each two terms vector, where each term is represented by vector (row) in terms space and selects the highest value, then the proposed system will rely on the result if it exceeded the specified threshold, it repeats this process but in reverse between the first and second text. The final output of this part, two vectors to the first approach and one vector to second approach. Finally, the system takes a weighted sum for the score of the similarity for two approaches to find the final similarity distance depended on the weight of each approach (the default w1 = 0.5) as shown in equation (2).

\[
\text{Sim}(T1,T2) = w1 \cdot \text{sim}(p1) + (1-w2) \cdot \text{sim}(p2) \quad \ldots \quad (2)
\]

![Figure (1): Block diagram general structure of proposed system](image)
5. Experimental Results

The proposed system are implemented by C# programming language from Microsoft Visual Studio .NET.2008. The tests are conducted on the environment: Windows 7 Home Premium 64-bit operating system; HP Laptop of AMD Triple-Core processor 2.10 GHz speed. RAM 4.00 GHz.

To evaluate the proposed system as a text similarity measure, we use three different types of unstructured corpus. The first, it contains 128 documents for technical descriptions of the published patents (128 PATENT CORPUS), the second training dataset from Microsoft Research Paraphrase Corpus (MRPC), and the third corpus of Plagiarised Short Answers. In our first experiment, we compute the similarity score for 100 sentence pairs from test dataset in MRPC and find the correlation with human judges, repeat the same experiment with 128-patents corpus to experience the change corpus impact on the results and used two systems of texts similarity measure are free online with same sentences (www.tools4noobs.com/online_tools/string_similarity, and http://lsa.colorado.edu/). The percentage of results that have been obtained are 76% and 73% respectively, compare with human judgment, when threshold = 0.5 for semantic similarity of terms and normalize the score similarity result with binary human judgment at 0.5 consider is 1 and less than is 0. Table (1) show the results for the first ten sentences. In the second experiment, used proposed system as the application for detecting Plagiarism in texts. we compute the similarity score for 95 text files represent answers to five questions in computer science to compare with the results of global tools used for the purpose of detecting plagiarism in texts. The system proposed generates the semantic space from Plagiarized Short Answers Corpus. The best percentage of results that have been obtained 92% when threshold = 0.5 for semantic similarity of terms and consider is Plagiarism when the score similarity >= 0.4. Table (2) show the results for the first ten cases.

Table (1) Similarity result with human judgment

<table>
<thead>
<tr>
<th>TEXT ID1</th>
<th>TEXT ID2</th>
<th>human judgment</th>
<th>PATENT CORPUS</th>
<th>MRPC CORPUS</th>
<th>Tools4noobs</th>
<th>lsa.colorado</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1089874</td>
<td>1089925</td>
<td>1</td>
<td>0.98</td>
<td>0.89</td>
<td>0.69</td>
</tr>
<tr>
<td>2</td>
<td>3019446</td>
<td>3019327</td>
<td>1</td>
<td>0.66</td>
<td>0.6</td>
<td>0.62</td>
</tr>
<tr>
<td>3</td>
<td>1945605</td>
<td>1945824</td>
<td>1</td>
<td>0.65</td>
<td>0.77</td>
<td>0.84</td>
</tr>
<tr>
<td>4</td>
<td>2594779</td>
<td>2595060</td>
<td>0</td>
<td>0.55</td>
<td>0.45</td>
<td>0.51</td>
</tr>
<tr>
<td>5</td>
<td>3354381</td>
<td>3354396</td>
<td>0</td>
<td>0.33</td>
<td>0.4</td>
<td>0.59</td>
</tr>
<tr>
<td>6</td>
<td>1390995</td>
<td>1391183</td>
<td>1</td>
<td>0.86</td>
<td>0.81</td>
<td>0.81</td>
</tr>
<tr>
<td>7</td>
<td>2201401</td>
<td>2201285</td>
<td>0</td>
<td>0.45</td>
<td>0.43</td>
<td>0.54</td>
</tr>
<tr>
<td>8</td>
<td>2453843</td>
<td>2453998</td>
<td>1</td>
<td>0.56</td>
<td>0.62</td>
<td>0.52</td>
</tr>
<tr>
<td>9</td>
<td>1756630</td>
<td>1756502</td>
<td>1</td>
<td>0.83</td>
<td>0.66</td>
<td>0.69</td>
</tr>
<tr>
<td>10</td>
<td>938878</td>
<td>938896</td>
<td>0</td>
<td>0.27</td>
<td>0.2</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Table (2) Similarity to detecting plagiarism

<table>
<thead>
<tr>
<th>Person</th>
<th>Group</th>
<th>Task</th>
<th>Category</th>
<th>Proposed Method</th>
<th>Clough-Stevenson</th>
<th>Ferret</th>
<th>Sherlock</th>
<th>Turnitin</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>a</td>
<td>Non</td>
<td>0.20</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>b</td>
<td>Cut</td>
<td>Subset 0.70</td>
<td>0.85</td>
<td>0.38</td>
<td>0.27</td>
<td>1.00</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>c</td>
<td>Light</td>
<td>83</td>
<td>0.56</td>
<td>0.42</td>
<td>0.25</td>
<td>0.85</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>d</td>
<td>Heavy</td>
<td>0.43</td>
<td>0.34</td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>e</td>
<td>Non</td>
<td>0.29</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>a</td>
<td>Non</td>
<td>0.22</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>b</td>
<td>Non</td>
<td>0.39</td>
<td>0.05</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>c</td>
<td>Cut</td>
<td>0.90</td>
<td>0.85</td>
<td>0.60</td>
<td>0.71</td>
<td>0.74</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>d</td>
<td>Light</td>
<td>0.60</td>
<td>0.56</td>
<td>0.22</td>
<td>0.16</td>
<td>0.58</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>e</td>
<td>Heavy</td>
<td>0.60</td>
<td>0.34</td>
<td>0.11</td>
<td>0.15</td>
<td>0.49</td>
</tr>
</tbody>
</table>
6. Conclusion

The paper is proposed two approaches to determined the semantic similarity depended on information derived from corpus-based measures techniques, through apply LSA technique. The main advantage of our system is can apply on both long texts (documents) or short texts (sentences) and application to the machine learning, where the meaning of concepts is derived from the sentences contexts. Therefore it gives result can outperforms on other techniques if the corpus create from texts selected carefully to cover the certain language or specified domain. It dynamic with change the corpus contents. it has suitable complexity time for applications. It suffers from weaknesses that can overcome some of them through using small database to expansion the corpus to cover conjugation irregular verbs and long abbreviations to improve the performance. Cosine similarity mathematically easy and giving good results for different tasks but in tasks which need similarity like human judgment can become weakness, through use small number of highly representative terms (has high weight) gives high similarity.

References


Discourse Processes, 25, 259-284.


