International Journal of Intelligent Information Systems
2014; 3(6-1): 61-66
Published online October 28, 2014 (http://www.sciencepublishinggroup.com/j/ijiis)
doi: 10.11648/j.ijiis.s.2014030601.21
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

Software development for identifying Persian text similarity

Elham Mahdipour, Rahele Shojaeian Razavi, Zahra Gheibi

Computer Engineering Department, Khavaran Institute of Higher Education, Mashhad, Iran

Email address:
Mahdipour@khi.ac.ir (E. Mahdipour), rahil.razavi@gmail.com (R. S. Razavi), zgdayana184@yahoo.com (Z. Gheibi)

To cite this article:
Elham Mahdipour, Rahele Shojaeian Razavi, Zahra Gheibi. Software Development for Identifying Persian Text Similarity. *International Journal of Intelligent Information Systems.* Special Issue: Research and Practices in Information Systems and Technologies in Developing Countries. Vol. 3, No. 6-1, 2014, pp. 61-66. doi: 10.11648/j.ijiis.s.2014030601.21

Abstract: The vast span of nouns, words and verbs in Persian language and the availability of information in all fields in the form of paper, book and internet arises the need of a system to compare texts and evaluate their similarities. In this paper a system has been presented for comparing the text and determining the degree of Persian (Farsi) text similarities. This system uses TF-IDF method to give weight to sentences. Moreover, the roots of the nouns have been found and identical score has been given to synonyms and word families. The results gained from implementation indicate that the proposed system has a desired efficiency in comparing short texts.

Keywords: Text Similarity, TF-IDF, Semantic Similarity, Stemming

1. Introduction

Nowadays the information is growing and persons have collision problems with unauthorized or unrelated use of information. One of the problems in this regard is the deficiency of effective methods for evaluating the degree of similarities of the texts. Text mining- the extraction of the words features and comparing them with each other is the basic technology to respond to this problem. One of its applications is the evaluation of text similarity which has gained lots of attention in various applications nowadays. For instance, comparing the similarity of one paper to other papers and determining whether it is repetitive or not is one of the most usages of assessing similarity of the text in conference and journal publications.

Text comparison is the process of studying the degree of similarities and differences of the texts with each other by a computer program. Evaluating the similarity between two pieces of short texts is a highly significant task in applied researches and programs like: text-mining, text extraction, information retrieval in web and search engines. In this case, evaluating the similarity or differences between two short texts or two sentences is a main step for the system's better function [1]. For instance, in an interactive question and answer system, evaluating the similarity between two short texts like two questions, is a basic step in classifying questions as well as the suggested questions. In the case of documents retrieval in web, it has been proved that evaluating the similarity between two texts holds a great significance. For instance, when the page headings are used to display documents in the page using the same name for finding a special task [2]. In text mining, evaluating the similarity in short texts is a helpful method to discover the hidden knowledge from the database [3].Studying the similarities of short texts is also applicable in wide range of programs like formulas search [4].

Generally, different methods exists for text mining and evaluating the text similarity including: Information retrieval, clustering, graphs theory, machine learning, latin semantic analysis (LSA), N-gram, part of speech tagging (POS), singular values decomposition (SVD), machine translation and TF-IDF [2, 5, and 13].

For this purpose, the manner the natural language processor functions can be applied to evaluate text similarity in which the system receives a full sentence (preferably ended with a punctuation mark) and then processes the word through stages [5]. For instance the following general steps can be considered:

First, the words which have been separated by a space are recognized using database and the functions for recognizing combined words and provide the required data for the processor. In this stage, the phonemic form of the words is also formed.

In this stage the surveyor recognizes the type of sentence

and determines the structural features of the words and the phonemically form in adjective and noun modifiers.

Stemming is the process of weighting the words and sentences, computing the scores and creating the similarity matrix.

The results evaluated the text similarity matrix for the two related texts.

Using graph-based methods and TF-IDF, a software system has been created in this paper to compare the similarity of Persian texts. Accordingly, the structure of the paper is in this way: Section 2 has a review on primary topics of text similarity evaluation. Section 3 investigates what has been done in the field of comparing text similarity, Section 4 studies the suggested software of text similarity called as ”Iranian Persian Text Similarity System”. In section 5, the experimental result by performing the software is assessed.

2. The Primary Concepts of Evaluating Text Similarity

The semantic word similarity is used to introduce a degree of similarity between the words used in unique information of a big structure. In order to calculate the semantic similarity two measuring ways can be used: 1) Mutual point to point information [6] and 2) latent semantic analysis [7].

One of the simple ways to find the similarity between two parts of the text is to use lexical adaptation. That is, the similarity is determined based on the number of lexical units which exists in both parts of the text. Aas and Eikvil [8], made some changes in the stages of this simple method as: Stemming, omitting the stop words, marking a part of speech, the longest sequence adaptation as different weights and factors normalization [8]. The text-based semantic similarity which is widely used is in fact an estimation of some inquiries made as information retrieval or using latent semantic analysis which gains the text similarity by operating the relations of second rank words which have been automatically gained from the big collections. The procedure includes a method for formalizing the translation and interpretation which is normally used for aligning the sentences in case of sudden changes or an interpretation of a generation which uses distributive similarity in the route of dependency trees [9].

The evaluations related to semantic similarity has traditionally been defined between the words or concepts and textual parts consist of two or some words. One of the indices of word to word similarity is the accessibility of the resources which encode the relation between words and concepts. In addition to this derivation, measuring the text to text similarity begins with a word based on semantic similarity may have no step forward. Mainly, the most of what has been done in the field is the applied programs of the traditional model of vector space which sometimes develops to N-gram language model. Considering the two parts of input text, a score indicating similarity in the semantic level is automatically determined and as a result, simple lexical adaptation method is applied for this purpose. The fact is that a comprehensive index of the text semantic similarity should be considered in its structure. To solve this problem, first a piece of the text is chosen and for modeling, the semantic similarity of the text is regarded as a function of semantic similarity of part of the word. This is done with the indices of word to word similarity and that group of a word features which are considered as a potential good formula for semantic similarity of two input texts [10].

What can be concluded so far is that first of all, we should act to divide the words and their meaning. This demands using stemming algorithms.

2.1. Stemming

Words in each language are divided into two groups of simple and derivational. The words which are derived from other words are called derivational. Simple words are those which have not been derived from any other word. Finding the root of the derivational words is called stemming. Due to developments in natural language processing, stemming has found lots of applications. Generally, there are two main applications for stemming of the words.

Stemming in Machine Translators: It is clear that words accompanied with their derivations give significant variety to the words which practically makes sentence translation difficult. In this method by using stemming, the complexity of the translation is decreased.

Stemming in Information Retrieval Systems: Information retrieval and text process is regarded as one of the growing applications of the recent. Processing and classifying the news, processing scientific texts and alike, are regular today. In information retrieval systems, there is usually a very huge database on which information retrieval and process has to be done. The more precise and developed the semantic networks extracted from these information are, the more convenient will be to access the extracted information. One of the applications of stemming is to provide more developed semantic networks in text process system and information retrieval.

In spite the fact that the problems in two above-mentioned applications are similar, words stemming in them has different demands. In translator systems, we often try to find the roots of the words whose derivation does not make any change in its type of that word (verb, noun, etc.).When the type of the word changes, its equivalent in the target language changes a lot and this practically ends to a translation of low quality. While in text process systems, discovering all relations is very important. Therefore, in machine translations, the emphasis is mostly on the cases where there is no change in the function of the words like verbs conjugation .Of course, this does not mean that stemming has no applicability in cases of translation systems, rather, by considering current means of translation systems which are mostly applicable in the level of words structure and not in the level of concept, stemming has to be directed in the same direction.

One of the algorithms of stemming, whose Persian version has been used in this paper, is Krovetz algorithm [11]. This algorithm applies morphological methods and a dictionary for trying found roots. This algorithm has shown a desirable efficiency in languages in which compound words structure is

rule–governed. Hungarian and Hebrew languages are placed in this category. Krovetz algorithm studies the prefixes and suffixes of the words and has shown an acceptable efficiency in translator machines. In Persian language in which word derivation is systematic, stemming is well capable to become mechanized. As noted before, for languages that have more morphological derivations, the capabilities of Krovetz algorithm are more evident. Persian and Arabic languages are placed in this category. Similar methods have only used linguistic structures .As a result, their results can be improved. Krovetz algorithm and Krovetz2 have been developed for verbs stemming in Persian language [12].

After stemming the words, considering the frequency and abundance of each word, a weight is assigned to it. In this paper TF-IDF method [13] is used to give weight to words.

2.2. TF-IDF Weighting

TF-IDF method equals the index of term frequency – inverse document frequency in the method of information retrieval. In this method, weight giving tf-isf is calculated for each sentence, in a way that $tf_{i,j}$ is said to the frequency of i^{th} word in j^{th} sentence and isf_i is the inverse of document frequency of i^{th} word. Where N is the number of all sentences and n_1 the number of sentences containing i^{th} word [13] (Equation 1).

$$isf_i = \log\frac{N}{n_i} \quad tf_{i,j} = \frac{freq_{i,j}}{\max_i freq_{i,j}} \tag{1}$$

Thus the weight of the words is calculated in the following way (Equation 2):

$$W_{i,j}=tf_{i,j}*isf_i \tag{2}$$

3. Related Works

The current available text similarity systems are for other languages except Persian and there has been no comprehensive language for evaluating Persian language text similarities. Among available systems for other languages, the following tasks have been studied.

In Rada et al. [14] knowledge-based methods have been used to measure semantic similarity of the texts. Since a great part of the information available today include short texts (scientific papers abstracts, notes on the pictures, descriptions of the products), their paper studies the semantic similarity of short texts. It offers methods for measuring semantic similarity of the texts using information and trivial similarity between them, as well as methods for extracting similarity as text to text and semantic similarity in knowledge-based method. Their results indicate that the semantic similarity methods based on simple lexical adaptation has caused 13% error reduction to the evaluation methods based on vector space.

In Toral et, al, [15] the ambiguous and unambiguous relationship between the nouns in Word net and Wikipedia have been evaluated based on text similarity methods. They consider a combination of supervised and unsupervised methods. The gold standard with disambiguated links is publicly available. The results range from 64.7% for the first sense heuristic, 68% for an unsupervised combination, and up to 77.74% for a supervised combination.

In [16] a new method has been presented for measuring the similarity between two short texts by comparing each of them with probable subjects. Their goal is to find discrimination between two short texts and compare them with series of probable subject's extracted using Gibbs sampling method. The conditions of short text discrimination are gained by studying their probabilities under subjects that have been discovered in that field. The similarity between two textual short abstracts is gained based on their normal conditions as well as relationship between their differences. Extensive tests in the ground of questions interpretation and categorizations indicate that the suggested method can perform a more precise computation for evaluating the degree of similarity compared to other methods that use TF-IDF.

4. Persian Text Similarity System

The programming language of Persian text similarity system is C#.net and uses Microsoft Access database. The reason why it uses Access database is that the program is easily used in each system with no need to SQL Server. Moreover, since during the performance, the database of the program do not face any changes including inserting, deleting and editing, the speed of performing the operation in SQL database has no benefit for the system, Access database was used.

Generally, a text similarity system is made of segmentation, stemming and scoring sections. Persian text similarity system holds two actors of user and text similarity method and two units of initializing and scoring. The initializing unit includes pre-process and segmentation. The scoring unit includes weight giving and creating the matrix of similarity. Accordingly, the stages of implementing the Persian text similarity system are as follows.

First, a collection of general knowledge on natural languages (NLP) has to be presented in order to facilitate text segmentation to the desired extracted unites. In a coherent text a word may usually appear in several different forms. These forms of derivation if in the form of plural or singular are controlled by the text. After the process of stemming, each word is shown with its root. In most cases, the different forms of the word have a similar semantic interpretation and hence can be acted as synonyms for a large number of information management. Thus in the first stage, using database, the synonyms, special and redundant and the algorithm of stemming all words and verbs are root-found and prepositions, plural markers and unimportant words are omitted.

In the second stage, the frequency of the words is gained using TF-IDF, Equation (1). Then the weight of each word in the sentence is calculated by Equation (2). The third stage is making matrix of similarity using Equation 3 in which the similarity of two sentences with each other is calculated.

$$sim(s_m, s_n) = \frac{\sum_{i=1}^{t} w_{i,m} * w_{i,n}}{\sqrt{\sum_{i=1}^{t} w_{i,m}^2} * \sqrt{\sum_{i=1}^{t} w_{i,n}^2}} \quad (3)$$

Where *m* refers to the sentences of the first text and n refers to the sentences of the second text. $W_{i,m}$ is the weight of i^{th} word in the first text in m^{th} sentence of the first text. Similarly, $W_{i,n}$ is the weight of i^{th} word in the first text in n^{th} sentence of the second text . Using Equation (3), the similarity matrix is formed. It is a m*n matrix in which m refers to the number of sentences in the first sentence equal to matrix rows and n is the number of sentences of the second text or the columns of the matrix. After making the similarity matrix, the weight graph is formed in which the weight of each edge for two joint vectors is the degree of similarity of two sentences to each other.

Furthermore, in order to get familiar with the manner of implementing the Persian text similarity system, its algorithm is defined below:

1. Receiving the first text.
2. Receiving the second text.
3. Segmenting the sentences of the first and second texts.
4. Separating the words of the first and second texts.
5. Stemming of the first and second texts.
6. Calculating the frequency of the words of the first text based on TF-IDF weight giving system.
7. Calculating the frequency of the words of the first text in the second text based on TF-IDF weighting system.
8. Scoring the sentences of the first and second texts.
9. Making similarity matrix according to Equation (3).
10. After making the similarity matrix, each matrix element refers to the degree of similarity of each sentence of the first text with each sentence of the second text. Now, in order to calculate the percentage of similarity of the two texts, primarily, for each sentence of the first text, the average of its similarity with all sentences of the second text is gained that is, the average of each row of similarity matrix is calculated. Then again an average is taken from all the averages in each row so that the percentage of total similarity is gained and announced to the user.

Figure 1 shows the flowchart of Persian text similarity system.

5. The Experimental Results

Figure 2 shows the software environment of the Persian text similarity system. Working with this software is very simple and facile. In order to work with this software, first click on the icon of open and choose the desired text. Do the same for the second text as well. The click "run" button to execute the software and see the result (Figures 2-4).

In order to evaluate the degree of accuracy of the text similarity declared by the software, manpower has been used in this research. Five volunteers studied the degree of textual similarities for 20 sample texts compared mutually. The satisfaction degree of each volunteer of the software outputs was recorded and finally the average of the volunteers' satisfaction was measured. The findings showed that the average of man's satisfaction of text similarity announced by the software is 64.31%. This criterion indicates the preciseness of Persian text similarity system.

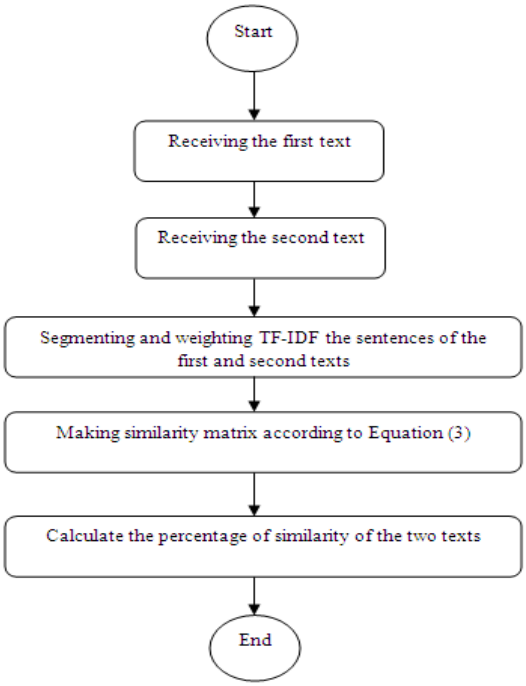

Figure 1. *Flowchart of Persian text similarity software steps.*

Figure 2. *Two different texts with some similar words.*

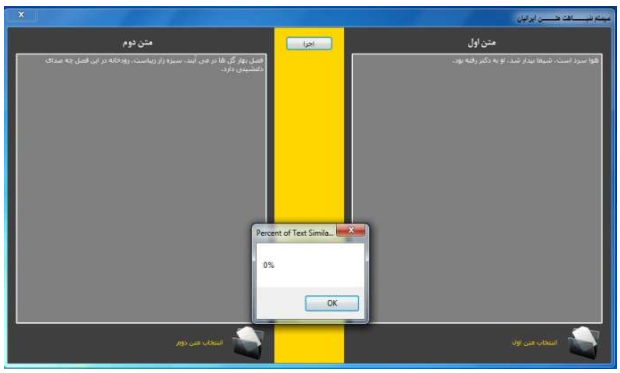

Figure 3. *Two complete different texts (No Similarity).*

Figure 4. *Two Same texts (complete similarity).*

6. Discussion, Conclusion and Future Work

The methods used in evaluating text similarity can be classified into two general groups. The first group is the statistical methods based on information retrieval (IR) which acts in lexical level and puts the statistical characteristics into consideration such as the frequency of the word due to neglecting the semantic relation between sentences, this method affects the text readability. The other approach existing in this evaluation gets benefit from natural language process and information extraction, thus tries to understand the subject and the relations between different parts of the text. The methods that use this approach, generally use syntactic-semantic analysis like LSA, lexical chain, random indexing and so on in order to discover the relations between entities. These methods use the word features of concurrence, co-reference, lexical similarity and semantic analysis. The results gained from the methods following this approach are usually of a higher quality. Usage of weight giving TF-IDF system is also a highly efficient way to gain frequency and other features of the words.

In the present paper, the similarities between two Persian texts were discovered using TF-IDF method. In order to calculate text similarity, the words were root found and the synonyms were accurately recognized and identical scores were assigned to them. The results gained by implementing this software indicate that by developing the database of this software, it can be used for larger texts as well.

The Persian text similarity system gained the human satisfaction average degree of 64.31% for evaluating the similarity of short texts and abstracts in all fields which indicates the preciseness of the offered system. Comparing the text of web pages including photos and link also needs a program to understand the layers and frames of the web pages so that it can extract the words and their features. The authors plan to upgrade the Persian text similarity system in order to compare long texts and the texts of web pages. Hence, it is planned to put the related information extensively in database.

References

[1] WenyinL, Hao TY, ChenW, FengM "A web-based platform for user interactive question answering". World Wide Web: Internet Web Inform Syst (2009) 12(2):107–124, 2009.

[2] Park EK, Ra DY, Jang MG, "Techniques for improving web retrieval effectiveness". Inform Process Manag 41:1207–1223, 2005.

[3] Atkinson-Abutridy J, Mellish C, Aitken S, "Combining information extraction with genetic algorithms for text mining", IEEE Intelligent Systems, pp: 22-30, 2004, Available on: http://homepages.abdn.ac.uk/c.mellish/pages/papers/atkinsoni eee.pdf.

[4] K Metzler D, Dumais S, Meek C, "Similarity measures for short segments of text". In: Proceedings of the 29th European conference on information retrieval (ECIR 2007). Lecture notes in computer science,vol 4425, Springer, Berlin , pp 16–27, 2007.

[5] Hassel, M., Resource Lean and Portable "Automatic Text Summarization", Stockholm, Sweden. p. 144, 2007.

[6] Turney, P. "Mining the web for synonyms: PMI-IR versus LSA on TOEFL". In Proceedings of the Twelfth European Conference on Machine Learning, 2001, Available on: http://www.extractor.com/turney-ecml2001.pdf.

[7] Landauer T. K., Foltz P., and Laham D, "Introduction to latent semantic analysis". Discourse Processes 25, 1998.

[8] K. Aas and L. Eikvil, "Text Categorisation: A Survey", 1999, Available on: http://citeseer.nj.nec.com/aas99text.html.

[9] Wu Z., Palmer M., "Verb semantics and lexical selection". ACL' 94 Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp: 133-138, 1994. Available on: http://dl.acm.org/citation.cfm?id=981751.

[10] Voorhees E., "Using WordNet to disambiguate word senses for text retrieval", SIGIR '93 Proceedings of the 16th annual international ACM SIGIR conference on research and development information retrieval, pp: 171-180, 1993, Available on: http://dl.acm.org/citation.cfm?id=160715.

[11] R. Krovetz, "Viewing morphology as an inference process", Proc. 16th ACM SIGIR Conference, Pittsburgh, June 27-July 1, pp. 191-202, 1993.

[12] Hessami Fard Reza, Ghasem sany Gholamreza, "Design of a stemming algorithm for Persian", 11th Annual Conference of Computer Society of Iran, Tehran, 2006. (Persian) Available on: http://www.civilica.com/Paper-ACCSI11-ACCSI11_066.html

[13] Qazvinian,Vahed.,SharifHassnabadi,Leila., Halavati, Ramin.,"Summarizing Text With a Genetic Algorithm-Based Sentence Extraction", Int. J. Knowledge Management Studies, Vol. 2, No. 4, pp:426-444, 2008, Available on: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130 .2201&rep=rep1&type=pdf.

[14] Rada Mihalcea, Courtney Corley, Carlo Strapparava, "Corpus-based and Knowledge-based measures of text semantic similarity", AAAI '06 Proceeding of the 21st national conference on Artificial intelligence, Vol. 1, pp: 775-780, 2006.

[15] Antonio Toral, Oscar Ferrandez, Eneko Agirre, Rafael Munoz, "A study on linking Wikipedia categories to Wordnet synsets using text similarity", International Conference RANLP 2009, Borovets, Bolgaria, pp: 449-454, 2009.

[16] Xiaojun Quan, Gang Liu, Zhi Lu, Xingliang Ni, Liu Wenyin, "Short text similarity based on probabilistic topics", Knowl Inf Syst, 25, pp:473-491, DOI:10.1007/s10115-009-0250-y, 2010.