Expanding Search Accessibility using Cosine Similarity Measure

Jaswinder Singh
Assistant Professor, Department of Computer Science & Engineering,
Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India

Abstract: With the increase in the content on the internet it is difficult for the user to get the relevant information when a query of two or three words is usually typed by the user for searching any information of interest from the web world. These short queries and the incompatibility between the terms of query and the pages affect the relevancy of retrieved pages. When user enters request in the form of query then the matching mechanism of the search system delivers the ranked list of documents to the user using the similarity measures. In this paper Cosine similarity measure is used as fitness function and Genetic Algorithm is applied for enpanding the accesibility of search. The training data of retrieved documents for formulated queries was prepared using the Google search Engine.

Keywords: Cosine Similarity Measure, Genetic Algorithm, Query expansion.

I. Introduction

The similarity measurement between the different objects is the fundamental function of any information retrieval application and there are varieties of ways to compute the similarity among the different object representations. Textual similarity functions play a vital role in tasks and applications of information retrieval i.e. document clustering, topic detection, question answering, text classification and others. Textual similarity can be measured lexically and semantically. For the words have which similar character order then they are said to be lexically similar but for the words which are used in the same perspective, then they are said as semantically similar. In general the character based similarity measures are used to compare the short strings and it is because of this reason it is too expensive to apply them for the large documents so the token base similarity functions or term-based similarity functions avoids these problems by viewing as the bag of terms or tokens. The basic and ultimate goal of the IRS is to deliver the relevant documents that have the capability to satisfy the user's need and the success of IRS depends on capability to assess the significance of objects in its repository i.e. information units, documents, functions, commands etc. to the given user's request [1] . With the increase in the content on the internet it is difficult for the user to get the relevant information when a query of two or three words is usually typed by the user for searching any information of interest from the web world. These short queries and the incompatibility between the terms of query and the pages affect relevancy of retrieved pages. When user enters request in the form of query then the matching method of the search system delivers the ranked list of documents to the user using the similarity measures. The database containing pages, query system and matching method are three fundamental components of IRS [2], [3], [4]. If the user is not fulfilled with the results returned by search system then user reformulates query there by increasing the retrieval effectiveness iteratively and incrementally [3]. The user evaluates the results on the basis of retrieved documents and provides the relevant feedback for the expansion of terms of initial query. Query expansion is a technique used to increase the effectiveness of the information retrieval [2]. It is the process of adding some more terms or phrases to the existing query to improve relevancy of the retrieved documents. The reformulated query contains more terms so the probability of matching them with terms in relevant documents is therefore enhanced. This paper contains six sections. The first section of explains the introduction about similarity measure and its role in information retrieval system. The second section of paper describes the work related to similarity measure and expansion of query. The third section describes the methodology followed to achieve the results. The fourth section of paper describes the experimentation and the results obtained are described in section five of paper. Section six of paper describes the conclusion.

II. Related Work

Similarity Measures: Similarity of the text can be computed with the string similarity Measures. Similarity between the strings can be measured using the string similarity measures and these are categorized as the similarity coefficients based on sequence and similarity measures based on tokens. The similarity coefficients based on sequence measures similarity between the strings by viewing the strings as the adjoining sequences which differ at the individual character level and the token similarity measure based on tokens measures the similarity between the strings by viewing the stings as the unordered set of tokens [5].Categorization of string based similarity measures is shown in fig.1.

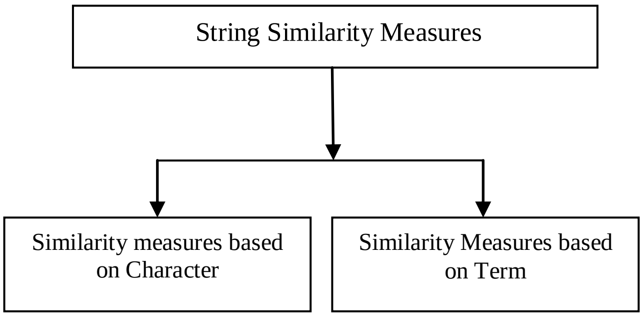
String Similarity Measures
Similarity measures based on Character
Similarity Measures based on Term

Fig. 1: String based similarity measures

Generally the similarity measures based on characters are used to compare the short strings. In such type of similarity measure, the character of the one set is compared with the other set of characters. The similarity measures based on characters include longest common substring, Jaro, Jaro Wrinkler, Damerau-Levenshtein, Needleman-Wunsch, Smith-Waterman and n-gram [6]. Another category of string similarity measures is the similarity measures based on terms. These include Jaccard, Cosine, Dice and Overlap similarity measures [6]. In general similarity measures based on characters are used to compare the short strings and it is because of this reason it is too expensive to apply them for the large documents so similarity measures based on terms avoids these problems by viewing as the bag of terms.

Query Expansion: From the literature related to the query expansion it was found that the relevancy of the retrieved documents can be increased by adding the terms in the query as with the addition of more terms in the query there is more probability of retrieving the relevant documents. From the literature it was found that the local analysis and the global analysis are two methods for expanding query and local analysis is better method to expand the query in which the ranked documents are retrieved first and then important terms from the pages are extracted and then added to query, this will enhance relevancy of the retrieved documents. Different authors used the different methods to optimize the query and by studying the literature it was also found that the genetic algorithm is the good method for the optimization.

III. Methodology followed

As the web is increasing day by day and there is variety of data but the major content is the text .When the user enter his or her query then only two or three words of text are written and the search engine returns the pages related to the text written in the formulated query. In this paper, only text is considered for studying its impact on the accessibility of search system. The methodology includes following steps.

Step1: Preparation of Training Data

The experimentation starts with the preparation of training data. Queries were chosen for retrieving the web pages from the web by using Google search engine. In the experiment ten queries were chosen which are described in the table1. Additionally, it is also required to choose the structure of data for the experimentation.

Table 1: Queries used in experiment

Query No.	Query
Q1	Terrorist Attack Mumbai
Q2	Cloud Burst India
Q3	Moist Attack India
Q4	Corruption Cricket India
Q5	Pollution River Ganga
Q6	Power Generation India
Q7	Sand Mining India
Q8	Mid Day Meal India
Q9	Sikh Riots India
Q10	Moist Attack Train

Step 2: Analysis of similarity function

In the information retrieval system the similarity as well as the relevancy of the retrieved pages relies on the similarity measures. Therefore the importance of the results relies on selection of similarity measures. The selection of similarity measures further rely upon category, class or the family of similarity measure. After the preparation of training data the similarity functions are chosen from the literatures which are used in the information retrieval. This section of paper explains that how the similarity of the retrieved documents is measured using the chosen similarity functions and how it is implemented on the training data. The performance analysis of the similarity functions i.e. Cosine is done with the help of the training data in the experiments. In the case of information retrieval when query is formulated by the user the documents are retrieved from the data base of retrieval system. For example, Similarity measure measures the degree of similarity between two sub sets X and Y of the entire data base of the documents in the repository i.e.

“X is defined, a set of all terms occurring in document X
Y is set of all terms occurring in document Y.

|X| = Numbers of terms that occur in set X.
|Y| = Number of terms that occur in set Y.
| X ⋂Y | =Number of terms occur in both X and Y.”

For X and Y subsets of documents retrieved from the entire repository of documents. The formula for the Cosine similarity function was defined in [7], [2], [12]. Cosine similarity measure is classified as similarity measure based on term [6], [8]. Cosine similarity between the set of terms of first document set i.e. X and the set of terms of second document set i.e. Y is defined as follows.

$$\mathrm{Cos}(X, Y) = \frac{|X \cap Y|}{\sqrt{|X|}\sqrt{|Y|}}$$

Step3: Query Expansion

Several techniques are there in literature for the expansion of query and after studying the literature it was concluded that the local feedback is good technique for the expansion of query and this technique is implemented by applying the genetic algorithm. While applying the genetic algorithm it is required to have the training data in the proper form of strings of zeros and ones so as to form the population to apply the genetic algorithm operators.

IV. Experimentation

This experimentation is done by formulating ten queries by using Google search engine. Search system provide outcome in form of pages. The first ten pages are taken in the experiment. After collecting pages, keywords are extracted from documents by Textalyser tool [9]. Chromosomes are encoded in form of binary [10], [11]. All keywords of documents are arranged in ascending order in form of a set as it was described in [12]. These chromosomes are called initial populations that are fed into genetic operators. The code for fitness evaluation and implementation of genetic algorithm is done using MATLAB. The keyword of a set which is present in a document is assigned one otherwise zero. Similarity of documents retrieved from search engine is calculated by Cosine coefficient fitness function and results are shown in table 3. After evaluating population’s fitness, the next step is chromosome selection. Selection operator selects only those chromosomes which have higher fitness value. Here roulette wheel selection was used for this purpose. After this cross over and mutation operator were applied. The experiment was done for 500 generations with probability of crossover i.e. 0.5 and mutation probability .001. The experiment was repeated with different probability of crossover and mutation rates. New keyword was chosen by selecting the bit position which has value one in the mutated chromosome. Only one keyword is added to the original query. Average similarity with new keyword is calculated and the results are shown in table 3.

V. Results

The experimentation of Cosine similarity measure used as the fitness function in the Genetic Algorithm is performed. Similarity between the retrieved documents is measured using the Cosine similarity function for the newly formulated query i.e. Q1’ which is formed by adding the keyword “ Headly” i.e. “Headly Terrorist Attack Mumbai”. Doc1’, Doc2’……Doc10’ are the documents represented with the presence of terms in the text of documents retrieved for newly formulated query.

Doc1’= 0,1,1,0,0,0,0,0,1,0,0,1,0,0,1,0,1,1,0,0,1,0,0,0,0;
Doc1’= 0,1,1,0,0,0,0,0,1,0,0,1,0,0,1,0,1,1,0,0,1,0,0,0,0;

$$\text{Cosine1'} = \mathrm{Cosine}(\mathrm{Doc1'}, \mathrm{Doc1'}) = \frac{|8|}{\sqrt{|8|}\sqrt{|8|}} = 1$$

Doc1’= 0,1,1,0,0,0,0,0,1,0,0,1,0,0,1,0,1,1,0,0,1,0,0,0,0;
Doc2’= 0,1,0,0,1,0,0,0,1,0,0,1,0,0,1,0,1,0,1,0,0,0,0,0,0;

$$\text{Cosine2'} = \mathrm{Cos}(\mathrm{Doc1'}, \mathrm{Doc2'}) = \frac{|5|}{\sqrt{|8|}\sqrt{|7|}} = 0.6681$$

In the similar way the values of Cosine were obtained as Cosine3’, Cosine4’, Cosin5’, Cosin6’, Cosine7’, Cosine8’, Cosine9’ and Cosine10’ respectively. The average of all the values of cosine similarity is calculated as shown in table 2.

Table 2: Cosine similarity for query “Headly Terrorist Attack Mumbai”

Docs	**Similarity with Cosine Similarity Measure**										**Avg.**
	Doc1’	Doc2’	Doc3’	Doc4’	Doc5’	Doc6’	Doc7’	Doc8’	Doc9’	Doc10’	
Doc1’	1	0.6681	0.5345	0.5345	0.6681	0.5773	0.5345	0.4330	0.2672	0.4330	0.5651
Doc2’	0.6681	1	0.4285	0.42857	0.5714	0.4629	0.4285	0.3086	0.2857	0.4629	0.5045
Doc3’	0.5345	0.4285	1	0.8571	0.5714	0.4629	0.7142	0.6172	0.2857	0.3086	0.578
Doc4’	0.5345	0.4285	0.8571	1	0.5714	0.4629	0.7142	0.6172	0.2857	0.3086	0.578
Doc5’	0.6681	0.5714	0.5714	0.5714	1	0.6172	0.7142	0.6172	0.2857	0.4629	0.608
Doc6’	0.5773	0.4629	0.46291	0.4629	0.6172	1	0.6172	0.3333	0.3086	0.3333	0.5176
Doc7’	0.5345	0.4285	0.7142	0.7142	0.7142	0.6172	1	0.4629	0.2857	0.3086	0.578
Doc8’	0.4330	0.3086	0.6172	0.6172	0.6172	0.3333	0.46291	1	0.3086	0.5	0.5198
Doc9’	0.2672	0.2857	0.2857	0.2857	0.2857	0.3086	0.2857	0.3086	1	0.4629	0.3776
Doc10’	0.4330	0.4629	0.3086	0.3086	0.4629	0.3333	0.3086	0.5	0.4629	1	0.4581

In the experiment similarity using Cosine similarity measure is measured with the new query i.e. Q1’ and average value of similarity is obtained which is 0.5285.This value is compared with the value of previous query i.e. Q1 which is 0.4280 and percentage improvement in similarity is calculated which is 10 %. The process was repeated with the other queries i.e. Q2, Q3,… Q10, the results are summarized in the table 3 and results shows that there is improvement in the relevancy or similarity of the retrieved documents when the term retrieved from experiment using genetic algorithm was added into the original query.

Table 3: Similarity using Cosine similarity measure with the added term

Query No.	**Query**	**Similarity of pages with original query**	**New Added term in Query**	**Similarity of pages with added term**	**Percentage Improvement**
Q1	Terrorist Attack Mumbai	0.4280	Headly	0.5285	10.05
Q2	Cloud Burst India	0.3112	Uttarakhand	0.4699	15.87
Q3	Moist Attack India	0.3345	Train	0.5116	17.71
Q4	Corruption Cricket India	0.4093	Fixing	0.4809	7.16
Q5	Pollution River Ganga	0.5969	Industrial	0.6247	2.78
Q6	Power Generation India	0.3823	Thermal	0.4868	10.45
Q7	Sand Mining India	0.5210	Illegal	0.5629	4.19
Q8	Mid Day Meal India	0.4278	Bihar	0.4738	4.6
Q9	Sikh Riots India	0.4784	Sajjan	0.6274	14.9
Q10	Moist Attack Train	0.5116	People	0.5614	4.98

VI. Conclusion

Genetic Algorithm based approach was implemented and it was concluded that the similarity of retrieved documents is improved by using the Cosine similarity measure as the fitness function in Genetic Algorithm and percentage increase in the similarity is measured. With the expansion of the terms of query, it was found that with the accessibility of more terms the accessibility of search has been expanded.

References

[1] W. P. Jones and G. W. Furnas, “Pictures of relevance: A geometric analysis of similarity measures,” Journal of the American Society for Information Science, vol. 38, no. 6, pp. 420–442, 1987.

[2] R. Baeza-Yates and B. Ribiero-Neto, Modern Information Retrieval. Addison Wesley, New York, 1999.

[3] V. N. Gudivada, V. V. Raghavan, W. I. Grosky, and R. Kasanagottu, “Information retrieval on the world wide web,” IEEE Internet Computing, no. 5, pp. 58–68, 1997.

[4] Michael Gordon, “Probabilistic and genetic algorithms in document retrieval,” Communications of ACM, vol.31, no. 10, pages. 1208-1218, 1988.

[5] M. Bilenko and R. J. Mooney, “Adaptive duplicate detection using learnable string similarity measures,” Proc. 9th ACM SIGKDD, Int. Conf. Knowledge Discovery and Data Mining, KDD-2003, Washington DC, USA, 2003 pp. 39-48.

[6] W. H. Gomaa and A. A. Fahmy, “A survey of text similarity approaches,” International Journal of Computer Applications, vol. 68, no. 13, pp. 13–18, 2013.

[7] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval. McGraw-Hill, New York, USA, 1986.

[8] M.C. Kim and K. S. Choi, “A comparison of collocation-based similarity measures in query expansion,” Information Processing & Management, vol. 35, no. 1, pp. 19–30, 1999.

[9] http://textalyser.net.

[10] Jaswinder Singh, Parvinder Singh, Yogesh Chaba,“ Increasing the visibility of search using Genetic Algorithm,” Journal of Computer Engineering, vol.17, issue 5,ver.1,pp.7-17, 2015.

[11] Z. Michalewicz, Genetic Algorithm + Data structure = Evolution programs. Springer, 1996.

[12] Jaswinder Singh, Parvinder Singh, Yogesh Chaba,“ Performance Modelling of Information Retrieval Techniques Using Similarity Functions in Wide Area Networks,” International Journal of Advanced Research in Computer Science and Software Engineering, vol.4, issue 12, pp.786-793, 2014.