Evaluating the Performance of Similarity Measures Used in Document Clustering and Information Retrieval

R.Subhashini[1] and V.Jawahar Senthil Kumar[2]
[1] Research Scholar, Sathyabama University, Chennai-119, India
[2] Assistant Professor, Anna University, Chennai, India
E-mail: [1]subhaagopi@gmail.com

***Abstract*-This paper presents the results of an experimental study of some similarity measures used for both Information Retrieval and Document Clustering. Our results indicate that the cosine similarity measure is superior than the other measures such as Jaccard measure, Euclidean measure that we tested. Cosine Similarity measure is particularly better for text documents. Previously these measures are compared with the conventional text datasets but the proposed system collects the datasets with the help of API and it returns the collection of XML pages. These XML pages are parsed and filtered to get the web document datasets. In this paper, we compare and analyze the effectiveness of these measures for these web document datasets.**

Keywords-Document clustering; Web mining; similarity measure, Information Retrieval

I. INTRODUCTION

Similarity measure is an important parameter in text mining. Similarity measure enables us to find the similarity or commonality existing between the text documents. Traditionally Term-Frequency (**TF**) approach has been used for representation of text document where the document is considered as a vector with a set of terms occurring in the document and their frequencies. There are two problems synonymy and polysemy, associated with the approach. Synonyms are terms with the same meaning with one another and two terms that can be interchanged in a given context are said to be synonymous relative to the context. For example car, automobile, and vehicle may be considered synonyms in a particular document. Using thesaurus[16] or similar facility synonymous terms can be grouped together. Polysemy refers to the lexical ambiguity when the same word is used to express two or more meanings in different contexts. To correctly distinguish between the occurrences of the same term in different contexts with different meanings, knowledge rich approach using Natural Language Processing (**NLP**)[17] is a necessity.

Text document clustering groups similar documents that to form a coherent cluster, while documents that are different have separated apart into different clusters. However, the definition of a pair of documents being similar or different is not always clear and normally varies with the actual problem setting. For example, when clustering research papers, two documents are regarded as similar if they share similar thematic topics. When clustering is employed on web sites, we are usually more interested in clustering the component pages according to the type of information that is presented in the page. For instance, when dealing with universities web sites, we may want to separate professors home pages from students' home pages, and pages for courses from pages for research projects. This kind of clustering can benefit further analysis and utilize of the dataset such as information retrieval and information extraction, by grouping similar types of information sources together. Accurate clustering requires a precise definition of the closeness between a pair of objects, in terms of either the pair-wise similarity or distance. A variety of similarity or distance measures have been proposed and widely applied, such as cosine similarity and the Jaccard correlation coefficient. Meanwhile, similarity is often conceived in terms of dissimilarity or distance as well [1]. Measures such as Euclidean distance and relative entropy have been applied in clustering to calculate the pair-wise distances.

Traditional methods used text datasets but in this system, web documents are collected using yahoo API[17] and similarity measures are implemented on it.BOSS (Build your Own Search Service) is Yahoo! & apos;s open search web services platform. The goal of BOSS is simple: to faster innovation in the search industry. Developers, start-ups, and large Internet companies can use BOSS to build and launch web-scale search products that utilize the entire Yahoo! Search index. BOSS gives you access to Yahoo! & apos;s investments in crawling and indexing, ranking and relevancy algorithms, and powerful infrastructure. BOSS is a platform for the next generation of search innovation, serving hundreds of millions of users across the Web. This API returns the XML document that dump the search results.XML files listing URLs for a site along with additional metadata about each URL so that search engines can more intelligently crawl the site.

In order to achieve this goal, The system is designed by the following scenario:

1) The application takes the user query.
2) Submits it to the search engine API and gets its response as XML file.

978-0-7695-4152-5/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIIC.2010.42

3) XML file is parsed and downloaded to get the web documents .
4) Similarity measures are performed on these web documents.

Today, the information in the internet is growing enormously and it is very difficult for users to get the relevant documents .While surfing, it takes a lot of time to find the relevant information. Actually, it is a challenge in the field of information retrieval. Hence a lot of research papers are coming in this area.

This paper is organized as follows. The next section describes the Feature selection of the documents. Section 3 discusses about the Document Clustering. Section 4 presents the Information Retrieval and Query similarity measures and Section 5 explains experiment settings, evaluation approaches, results and analysis. And Section 7 concludes and discusses future work.

II. FEATURE SELECTION

All methods of information retrieval require several steps of preprocessing of the data. First, any non-textual information such as HTML-tags and punctuation is removed from the documents. Then, stop words such as "I", "am", "and" etc. are also removed. A *term* is any sequence of characters separated from other terms by some delimiter. Note that a term may either be a single word or consist of several words. Typically, the terms are reduced to their basic stem applying a stemming algorithm[15]. Most of the information retrieval methods rely on the so-called *vector-space model*. In this model, each text document *d* is represented by a vector of frequencies of the remaining *m* terms:

$$d = (tf_1, tf_2, \ldots\ldots\ldots\ldots tf_m) \quad (1)$$

Often, the document vectors are normalized to unit length to allow comparison of documents of different lengths. Note that the vector-space has a very high dimensionality since even after preprocessing there are typically still several thousands of terms, in many text databases you have to deal with approximately 10,000 terms. Due to the high dimensionality, most frequencies are zero for any single document. The important technology of retrieval and index based on keywords is TFIDF (Term Frequency-Inverse Frequency), TFIDF is one popular technology in information retrieval and exploration. TFIDF is a statistical method to evaluate the importance of one word or term in one of the files of documents sets or knowledge warehouse. The traditional TFIDF algorithm, by Gerald Sahon and McGill, is a method to describe the features of document for vector space information retrieval paradigm. The inverse document frequency IDFi of a term ti can be calculated as follows :

$$IDFi = \log_{10}(N/ni) \quad (2)$$

where IDFi denotes the inverse document frequency of term ti; N denotes the number of documents in the document set; ni denotes the number of documents containing term ti. The weight wik of term ti in document dk can be calculated as follows:

$$W_{ik} = tf_{ik} / \max_j tf_{jk} \; X \; IDF_i \quad (3)$$

where tfik denotes the frequency of term ti in document dk and maxtfjk denotes the maximal frequency of terms in document dk.

III. DOCUMENT CLUSTERING

Document clustering [2] is a technology that puts all the related documents in to groups and is useful for categorizing, organizing, refining search results. Most of the current document clustering methods are based on Vector Space Model (VSD) model [11].In which , documents are represented as feature vector of words. To achieve a more accurate document clustering, a more informative feature term, phrase-orderded sequence of one or more words is used in Suffix Tree Document (STD) model and was proposed by Zamir et al.[12]

Before clustering, a similarity/distance measure must be determined. The measure reflects the degree of closeness or separation of the target objects and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand, and there is no measure that is universally best for all kinds of clustering problems. Moreover, choosing an appropriate similarity measure is also crucial for cluster analysis, especially for a particular type of clustering algorithms Therefore, understanding the effectiveness of different measures is of great importance in helping to choose the best one.

A. Euclidean Distance

Euclidean distance is a standard metric for geometrical problems. It is the ordinary distance between two points and can be easily measured with a ruler in two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text. It satisfies all the above four conditions and therefore is a true metric. It is also the default distance measure used with the K-means algorithm .Measuring distance between text documents, given two documents da and db represented by their term vectors p,q respectively, the Euclidean distance of the two documents is defined as

$$d(p, q) = ((p1-q1)^2 + (p2-q2)^2 + \ldots\ldots+(pn-qn)^2)^{1/2} \quad (4)$$

where the term set is p = {p1, . . . , pn} and q = {q1,….,qn}. As mentioned previously, we use the tfidf value as term weights, that is wp= tfidf(da, p).

B. Cosine Similarity

When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity. Cosine similarity is one of the most popular similarity measure applied to text documents, such as in numerous information retrieval applications [4] and clustering too [3]. Given two documents X, Y and their cosine similarity is

$$COS(x, y) = X * Y / |X| |Y| \quad (5)$$

Where X,Y are m-dimensional vectors over the term set T = {t1, . . . , tm}. Each dimension represents a term with its weight in the document, which is non-negative. As a result, the cosine similarity is non-negative and bounded between [0,1].An important property of the cosine similarity is its independence of document length. For example, combining two identical copies of a document d to get a new pseudo document d0, the cosine similarity between d and d0 is 1, which means that these two documents are regarded to be identical. In other words, documents with the same composition but different totals will be treated identically. Strictly speaking, this does not satisfy the second condition of a metric, because after all the combination of two copies is a different object from the original document. However, in practice, when the term vectors are normalized to a unit length such as 1, and in this case the representation of d and d0 is the same.

C. Jaccard Coefficient

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms. The formal definition is:

$$J(A,B) = |A \cap B| / |A \cup B| \quad (6)$$

Where A & B represents the documents and the Jaccard coefficient is a similarity measure and ranges between 0 and 1. It is 1 when A = B and 0 when A and B are disjoint, where 1 means the two objects are the same and 0 means they are completely different. The corresponding distance measure is DJ = 1 − SIMJ and we will use DJ instead in subsequent experiments.

IV. INFORMATION RETRIEVAL - QUERY SIMILARITY MEASURES

As discussed, our approach to query clustering uses a hybrid method based on the analysis of query terms and query results. Here, two queries are similar when (1) they contain one or more terms in common; or (2) they have results that contain one or more items in common. The remainder of this section provides definitions of different query similarity measures used in our experiments. Our method of constructing query clusters based on different query similarity measures is also presented.

A. Content-based Similarity Measure

Similarity Measures also plays an important role in the field of Information Retrieval. It used to retrieve the relevant document from the web. Actually it is a challenging task in IR. It is also used to rank the results. This measure uses a hybrid method based on the analysis of query terms and query results. Here, two queries are similar when (1) they contain one or more terms in common; or (2) they have results that contain one or more items in common. In this paper we are going to calculate the similarity only for the common terms and not for the common results.

We borrow concepts from information retrieval [4] and define a set of queries as Q={Q1, Q2…Qi, Qj…Qn}. A query Qj is converted to a term and weight vector shown in (7), where qi is an index term of Qj and wiQj represents the weight of the ith term in query Qj. To compute the term weight, we define the term frequency, tfiQj, as the number of occurrences of term i in query Qj and the query frequency, qfi, as the number of queries in a collection of n queries that contains the term i. Next, the inverse query frequency, iqfi, is expressed as (8), in which n represents the total number of queries in the query collection. We then compute wiQj based on (9):

$$Qj = \{ <q1,W1q>;<q2,W2q>;\ldots\ldots qi,Wiq> \} \quad (7)$$

$$iqf_i = \log(n/qf_i) \quad (8)$$

$$wi\ Qj = tfi\ Qj * iqf\ i \quad (9)$$

Taking the term weights into consideration, we can use any one of the standard similarity measures [9]. Here, we only present the cosine similarity measure since it is most frequently used in information retrieval.
For any query q and a document d,

$$Sim(d,q) = cos(d,q) = d * q \quad (10)$$

In other words, let the *n*-dimensional vector of query-document similarities be:

$$Sim(A,q) = A*q \quad (11)$$

Where **A** is our *n* X *p* term document matrix, and **q** is a *p*-dimensional query vector.

B. Result URLs-based Similarity Measure

The results returned by search engines usually contain a variety of information such as the title, abstract, topic, etc. This information can be used to compare the similarity between queries. In our work, taking the cost of processing query results into consideration, we consider the query results' unique identifiers (e.g. URLs) in determining the similarity between

queries as in [8,16]. Let U(Qj) represent a set of query result URLs to query Qj:

$$U(Q) \{u, u2, \ldots\ldots\ldots u\} j = i i \quad (12)$$

where ui represents the ith result URL for query Qj. We then define Rij as (13), which represents the common query results URL vector between Qi and Qj. Here u refers to the URLs that belong to both U(Qi) and U(Qj).

$$R_{ij} = \{ u: u \in U(Q_i) \cap U(Q_j) \} \quad (13)$$

Next, the similarity definition based on query result URLs can be stated as:

A query Qi is similar to query Qj if |Rij|>0, where the |Rij| is the number of common result URLs in both queries. The similarity measure can be expressed as (14):

$$SIM_{Result}(Q_i, Q_j) = |R_{ij}| \div Max(|UQ_i|, |UQ_j|) \quad (14)$$

where the |U(Qi)| is the number of result URLs in U(Qi).Note that this is only one possible method for calculating similarity using result URLs. Other measures such as using the overlaps of document titles or domain names in the result URLs may be used.

V. EXPERIMENTAL RESULTS

A. Data Preparation

Web documents dataset are collected with the help of Yahoo API. Yahoo! Search BOSS (Build your Own Search Service) is an initiative in Yahoo! Search to open up Yahoo!'s search infrastructure and enable third parties to build revolutionary search products leveraging their own data, content, technology, social graph, or other assets. This release includes Web, News, and Image Search as well as Spelling Suggestions. Developers have expressed interest in altering the result order, removing results they do not want, and blending in their own data. All of these activities are allowed and encouraged with BOSS but not in the existing search API.

SYNTAX
http://boss.yahooapis.com/ysearch/web/v1/{query}?appid={yourBOSSappid}[&param1=val1&param2=val2&etc]

EX
http://boss.yahooapis.com/ysearch/web/v1/animals?appid=12345&format=xml&start=1&count=10

Where the parameter start represents the ordinal position of first result where first position is zero. The parameter count represents the total number of results to return; maximum value is 100.

B. Evaluation Measures

The number of clusters is set as the same with the number of pre-assigned categories in the data set. The quality of a clustering result was evaluated using two evaluation measures—purity and entropy[10], which are widely used to evaluate the performance of unsupervised learning algorithms [5,6]. To begin with, each cluster is labeled with the majority category that appears in that cluster. Moreover, if a category label has been assigned to a cluster, it still can be assigned to other clusters if it is the dominant category in that cluster. Based on the cluster labels, the purity and entropy measures are computed as follows. The purity measure evaluates the coherence of a cluster, that is, the degree to which a cluster contains documents from a single category. Given a particular cluster Ci of size ni, the purity of Ci is formally defined as

$$P(C_i) = 1/n_i \max_h (n_i^h) \quad (15)$$

where $\max_h(n_i^h)$ is the number of documents that are from the dominant category in cluster Ci and n_i^h represents the number of documents from cluster Ci assigned to category h. Purity can be interpreted as the classification rate under the assumption that all samples of the cluster are predicted to be members of the actual dominant class for the cluster. For an ideal cluster, which only contains documents from a single category, its purity value is 1. In general, the higher the purity value, the better the quality of the cluster is. As shown in Table 1, Euclidean distance performs worst while the performance of the other measures are quite similar. On average, Cosine measure is better than Jaccard in terms of purity, which means the clusters have higher purity scores.

Table.1.Result of Evaluation in terms of Purity

Data	Euclidean	Cosine	Jaccard
20 News	0.32	**0.64**	0.61
reo	0.48	**0.79**	0.72
wap	0.38	0.63	**0.65**

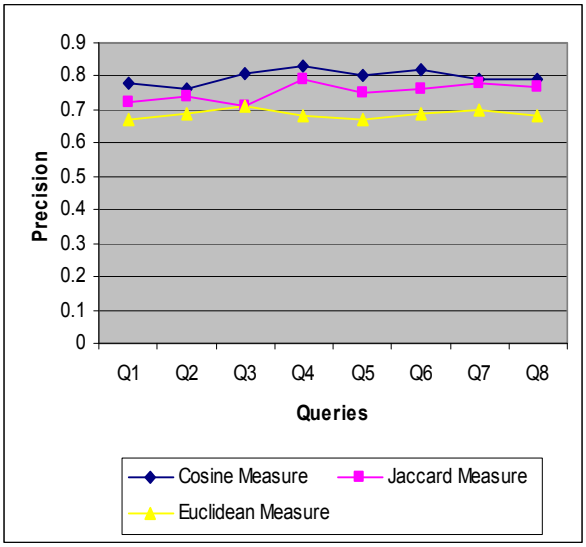

Figure.1. Performance of Similarity Measures

For the purpose of experiment in Information Retrieval model, only content based similarity measure is used. we selected 100 queries from the Google AdWords [18] and then these queries are used to retrieve the 2000 documents from the web using the Yahoo API [17] . This API returns the XML, that dump the search results and it contains the URL, Title, Snippet, Description. From which, web documents are collected by parsing the XML document. After that we calculated the similarity measures for these 100 queries and the documents. More similar documents are considered as relevant and accordingly they are ranked.. The similarity value for each query with the original documents was used to get relavent documents. In Fig.1. the three similarity measures are compared with the sample of Eight queries. In this the precision value of cosine measure is comparatively better for all the queries.

VI. CONCLUSION

This Paper proves that the similarity measures plays an important role in retrieving the relevant document and also to rank and cluster the documents. Metric distances (such as Euclidean distance) are not appropriate for high-dimensional, sparse domains. Cosine, correlation and extended Jaccard measures are successful and perform equivalently in capturing the similarities implicitly indicated by manual categorizations. To conclude, this investigation found that except for the Euclidean distance measure, the other measures have comparable effectiveness for the partitional text document clustering task. We can see that there are three components that affect the final results representation of the objects, distance or similarity measures, and the clustering algorithm itself. My future plan is to compare these measures in query clustering and web search result clustering. Further, content based and result-URL's based similarity measures should be combined to provide accurate results.

REFERENCES

[1] G. Salton. Automatic Text Processing. Addison-Wesley, New York, 1989.
[2] M. Steinbach, G. Karypis, and V. Kumar. A Comparison of document clustering techniques. In KDD Workshop on Text Mining, 2000.
[3] R. B. Yates and B. R. Neto. Modern Information Retrieval. ADDISON-WESLEY, New York, 1999.
[4] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In Proceedings of the International Conference on Information and Knowledge Management, 2002.
[5] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. Machine Learning, 55(3), 2004.
[6] Milne, O. Medelyan, and I. H. Witten. Mining domain-specific thesauri from wikipedia: A case study In Proc. of the International Conference on Web Intelligence (IEEE/WIC/ACM WI'2006), 2006.
[7] N.S. Glance, Community search assistant ,Proceedings of the 6th ACM International Conference on Intelligent User Interfaces (Santa Fe, January 2001), 91-96.
[8] R.Z. Osmar & S. Alexander, Finding similar queries to satisfy searches based on query traces,Workshops of the 8th International Conference on Object-Oriented Information Systems (Montpellier, September 2002), 207- 216.
[9] N.S. Glance, Community search assistant,Proceedings of the 6th ACM International Conference on Intelligent User Interfaces (Santa Fe, January 2001),91-96.
[10] J. van Rijsbergen. Information Retrieval. Second Edition, utterworths, London, 1979.
[11] A Vector Space Model For Automatic Indexing,G. Salton, A. Wong and C. S. Yang, Cornell University.
[12] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In Proceedings of SIGIR'98, University of Washington, Seattle, USA, 1998.
[13] Mehmet Ali Salahli, AN APPROACH FOR MEASURING SEMANTIC RELATEDNESS BETWEEN WORDS VIA RELATED TERMS, Mathematical and Computational Applications, Vol. 14, No. 1, pp. 55-63, 2009.
[14] Mrs. K. P. Supreethi Dr. E.V.Prasad ,Web Document Clustering Technique Using Case Grammar Structure, International Conference on Computational Intelligence and Multimedia Applications 2007
[15] M. F. Porter. An algorithm for suffix stripping.Program, 14(3):130–137, 1980.
[16] R.Z. Osmar & S. Alexander, Finding similar queries to satisfy searches based on query traces,Workshops of the 8th International Conference on Object-Oriented Information Systems (Montpellier, September 2002), 207- 216.
[17] http://developer.yahoo.com/search/boss/
[18] https://adwords.google.co.in/o/Targeting/Explorer?__u=1000000000&__c=1000000000&stylePrefOverride=2