International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 6 ǁ June. 2013 ǁ PP.75-78

Algorithm for Semantic Based Similarity Measure

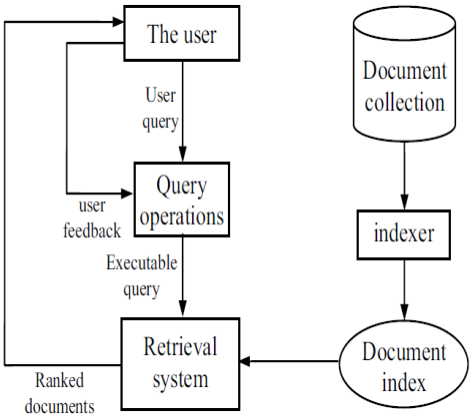
Sapna Chauhan[1,] Pridhi Arora[2] ,Pawan Bhadana[3]
[1]*M.Tech Scholar of computer science & Engineering, BSAITM, Faridabad*
[2]*Department of computer science & Engineering, BSAITM, Faridabad*
[3]*Department of computer science & Engineering,BSAITM, Faridabad*

ABSTRACT: *In a document representation model the Semanti based Similarity Measure (SBSM), is proposed. This model combines phrases analysis as well as words analysis with the use of propbank notation as background knowledge to explore better ways of documents representation for clustering. The SBSM assigns semantic weights to both document words and phrases. The new weights reflect the semantic relatedness between documents terms and capture the semantic information in the documents. The SBSM finds similarity between documents based on matching terms (phrases and words) and their semantic weights. Experimental results show that the semantic based similarity Measure (SBSM) in conjunction with Propbank Notation has a promising performance improvement for text clustering.*

KEYWORDS: *Click-through data, semantic similarity measure, marginalized kernel, event detection, evolution pattern*

I. INTRODUCTION

Information retrieval (IR) is the study of helping users to find information that matches their information needs. Technically, IR studies the acquisition, organization, storage, retrieval, and distribution of information. Historically, IR is about document retrieval, emphasizing document as the basic unit. Fig. 2.1 gives a general architecture of an IR system. In Figure 2.1, the user with information need issues a query (**user query**) to the **retrieval system** through the **query operations** module. The retrieval module uses the **document index** to retrieve those documents that contain some query terms (such documents are likely to be relevant to the query), compute relevance scores for them, and then rank the retrieved documents according to the scores..The ranked documents are then presented to the user. The **document collection** is also called the **text database**, which is indexed by the **indexer** for efficient retrieval



• **Fig. 2.1.** A general IR system architecture

II. SIMILARITY MEASURE TECHNIQUES

There is various type of similarity measures such as:
1Cosine similarity measure
2 Jacard similarity measure
3 Euclidean Distance measure
4 Metric similarity measure

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 [14].

Given two documents $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ their cosine similarity is.

$SIM_c(\overrightarrow{t_a}, \overrightarrow{t_b}) = = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{\overrightarrow{t_a} \times \overrightarrow{t_b}}$

Where $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ are m-dimensional vectors over the term set T = {t_1,……t_m}. 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. Meanwhile, given another document l, d' and d0 will.

Have the same similarity value to l, that is, sim($\overrightarrow{t_d}$, $\overrightarrow{t_l}$)= sim($\overrightarrow{t_d}$, $\overrightarrow{t_l}$) 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.

Jacard similarity: 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 documents but are not the shared terms. The formal definition is [14].

$SIM_j(\overrightarrow{t_a}, \overrightarrow{t_b}) = = \frac{\overrightarrow{t_a} \cdot \overrightarrow{t_b}}{|\overrightarrow{t_a}| + |\overrightarrow{t_b}| - \overrightarrow{t_a} \cdot \overrightarrow{t_b}}$

The Jaccard coefficient is a similarity measure and ranges between 0 and 1. It is 1 When $\overrightarrow{t_a}$ =$\overrightarrow{t_b}$ and 0 when $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ 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 – SIM_j and we will use D_j instead in subsequent experiments.

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 d_a and d_b represented by their term vectors $\overrightarrow{t_a}$ and $\overrightarrow{t_b}$ respectively, the Euclidean distance of the two documents is defined as [14].

$$D_E(\overrightarrow{t_a}, \overrightarrow{t_b}) = (\sum_{t=1}^{m} |w_{t,a} - w_{t,b}|^2)^{1/2}$$

Where the term set is T = {t1, . . . , tm}. As mentioned previously, we use the tfidf value as term weights, that is $w_{t,a}$ = tfidf(d_a, t).

Metric similarity: To qualify as a metric, a measure d must satisfy the following four conditions:
Let x and y be any two objects in a set and d(x, y) be the distance between x and y [14].

- The distance between any two points must be nonnegative, that is, d(x, y) ≥ 0.
- The distance between two objects must be zero if and only if the two objects are identical, that is, d(x, y) = 0 if and only if x = y.
- Distance must be symmetric, that is, distance from x to y is the same as the distance from y to x, ie. d(x, y) = d(y, x).
- The measure must satisfy the triangle inequality, which is d(x, z) ≤ d(x, y) + d(y, z

III. RELATED WORK

Phrases convey local context information, which is essential in determining an accurate similarity between documents. Toward this end, we devised a similarity measure based on matching phrases rather than individual terms. This measure exploits the information extracted from the previous phrase matching algorithm to better judge the similarity between the documents. This is related to the work of Isaacs and used a pair-wise

probabilistic document similarity measure based on Information Theory. Although, they showed it could improve on traditional similarity measures, but it is still fundamentally based on the vector space model representation.The phrase similarity between two documents is calculated based on the list of matching phrases between the two documents. From an information theoretic point of view, the similarity between two objects is regarded as how much they share in common. The cosine and the Jaccard measures are indeed of such nature, but they are essentially used as single-term based similarity measures.In Clustering of large collections of text documents is a key process in providing a higher level of knowledge about the underlying inherent classification of the documents. Web documents, in particular, are of great interest since managing, accessing, searching, and browsing large repositories of web content requires efficient organization. Incremental clustering algorithms are always preferred to traditional clustering techniques, since they can be applied in a dynamic environment such as the Web. An incremental document clustering algorithm is introduced in this paper, which relies only on pair-wise document similarity information. Clusters are represented using a Cluster Similarity Histogram, a concise statistical representation of the distribution of similarities within each cluster, which provides a measure of cohesiveness. The measure guides the incremental clustering process. Complexity analysis and experimental results are discussed and show that the algorithm requires less computational time than standard methods while achieving a comparable or better clustering quality

IV. PROPOSED WORK

There have been various attempts to label the sentence using semantic term labeler. Labeling the thematic role in a sentence is known as thematic role analysis [29, 30]. In our approach we have used PropBank [31] notation for labeling the each sentence of each document. Using the PropBank notation the sentence can be labeled in verb argument structure in more than one way if a term used as a argument with different verbs in the same sentence. Then it means the term has more significant semantic importance rather than others which has been used less number of times. So the weight assigned to each term which can be a single word or phrase will be based upon the count of how many times a term is used as an argument in the whole document in every verb argument structure of sentences.

For example consider the following:

"We have noted, how some soft computing techniques, developed for optimization, have eventually been used in data mining and others related fields."

By using the PropBank notation the above sentence can be represented in three ways in verb argument structure.

- [ARG0 We] [verb noted] [ARG1 how some soft computing techniques, developed for optimization, have eventually been used in data mining and others related fields]

-we have noted how [ARG1 some soft computing techniques][verb developed][ARGM_PNG for optimization] have eventually been used in data mining and others related fields.

-We have noted how [ARG1 some soft computing techniques, developed for optimization] have [ARGM-TMP eventually] been [verb used] [ARGM-LOC in data mining and other related fields].

After labeling the sentences some preprocessing is required which we have done using Porter Stemmer Algorithm [32]. After performing the stemming we end up having some labeled terms. The same process we have to do for query as well to get the labeled terms.

Now the algorithm given below is used to get the semantic similarity between the query and document. In the algorithm below Di is a document, and Qi is query where i=1, 2, 3…..k; and k is a positive finite integer. LDi and LQi are the list corresponding to document to document Di and query Qi to hold their labeled terms. A node of the list contains labeled term as data, weight as the count of labeled term and link to next node.

Algorithm: Semantic based similarity measure

```
1.  Di is a new document
2.  LDi  is  empty list
3.  for each sentence S in Di do
4.      for each labeled term in S do
5.        if(labeled term already in the list LDi)
6.           Increase labeled-term count by 1;
7.        else
8.          {
9.           Add a new node in the list
10.          Node->data=labeled-term;
11.           Labeled-term count =1
12.          }
```

```
13.        End for
14. End for
15. SQ is a temporary variable.
16.  For each labeled term in LQi do
17.        If(labeled-term in LQi==labeled-term in LDi)
18.          {
19.             SQ= SQ + Labeled-term count in LDi *          Labeled-term count  in   LQi;
20.          }
21. End for
22. Semantic similarity=SQ/sum of count of all labeled terms in LDi;
```

If we use the above algorithm to compute the weight of each labeled term then we found the count for labeled term "soft-computing", "developed" and "optimization" are highest. This shows that these terms are having more semantic significance rather than others labeled terms.

V. EXPERIMENTAL RESULT

The document collection we have used to test our algorithm is cisi dataset. The dataset has 1414 documents and 35 user queries. We have implemented the algorithm using MATLAB software. For finding cosine and jaccard similarity we have used TMG:A MATLAB TOOLBOX. TMG is basically text to matrix generator. We have used f-score as a fitness function. Overall fitness we have calculated in terms of f-score. We have taken a population of random weights in which each individual represent the weights for each similarity measure. We have run the algorithm upto 40 generations and got the optimized weight 0.932, 0.767, 0.621 respectfully. Fig. 5.1 below has shown the f-score over generations. Fig. 5.2 and Fig. 5.3 have shown the precision on various level of recall for cosine and jaccard respectively. While Figure 5.4 has shown the precision recall curve for our proposed semantic-based-combined-similarity- measure.

CONCLUSION

In our work we have combined various similarity measures to generate an effective matching function. Effectiveness of the matching function depends upon all similarity measures based on weight given by genetic algorithm. So to have an effective matching function both semantic and syntactic aspects should be taken into consideration while choosing similarity measures. We observed that no significant improvement has been seen in average fitness (f- score) value of overall generation after 40-50 iterations. The effect of crossover operator beyond this stage becomes insignificant due to very small variation in individual for particular generation. Applying fuzzy theory in our approach can control genetic algorithm and may lead to better results.

REFERENCES

[1.] Bing Liu, Web Data Mining, Springer, ISBN-10 3-540-37881-2

[2.] J. R. Quinlan. C4.5: Program for Machine Learning. Morgan Kaufmann, 1992

[3.] B. Liu, C. W. Chin, and H. T. Ng. Mining Topic-Specific Concepts and Definitions on the Web. In Proc. of the 12th Intl. World Wide Web Conf. (WWW'03), pp. 251– 260, 2003

[4.] J. L. Klavans, and S. Muresan. DEFINDER: Rule-Based Methods for the Extraction of Medical Terminology and Their Associated Definitions from On-line Text. In Proc. of American Medical Informatics Assoc., 2000

[5.] R. A. Baeza-Yates and B. A. Ribeiro-Neto. Modern Information Retrieval. ACM Press / Addison-Wesley, 1999

[6.] G. Bordogna and G. Pasi. Modeling vagueness in information retrieval. Lectures on information retrieval, pages 207–241, 2001

[7.] J. N. K. Liu. An intelligent system integrated with fuzzy ontology for product recommendation and retrieval. In FS'07: Proceedings of the 8th Conference on 8th WSEAS International Conference on Fuzzy Systems, pages 180–185, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[8.] R. Pereira, I. Ricarte, and F. Gomide. Fuzzy relational ontological model in information search systems. In Elie Sanchez. (Org.). Fuzzy Logic and The Semantic Web, pages 395–412, Amsterdan, 2006. Elsevier B. V

[9.] M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3), pp 130-137, 1980

[10.] Brin, S. and L. Page (1998). The anatomy of a large-scale hyper textual Web search engine. Computer Networks and ISDN Systems 30 (1-7), 107-117.