Text classification Collation of TF-IDF and LSI on the basis of Information retrieval and Text categorization.

Manasi Desai
Thakur College of Engineering & Technology

Chetan Raman
Thakur College of Engineering & Technology

Karan Kadakia
Thakur College of Engineering & Technology

Harshali Patil
Thakur College of Engineering & Technology

Abstract

Text categorization is a significant concept in the field of text mining today. In this paper we will be studying two algorithms on the Text categorization, one being Term Frequency and Inverse Document Frequency (TF-IDF) and the other being the traditional method of text categorization based on indexing on Latent Semantics Indexing (LSI). TF-IDF algorithm compared to LSI algorithm on the basis of text classification which is based on the content information retrieval (IR) and Text categorization (TC). In this paper, we examine TF-IDF to determine what words in a corpus of documents might be more favorable to use in a query and LSI is used to get back the information that uses the method of singular value decomposition (SVD). Moreover, LSI being the traditional approach for text classification, TF-IDF relatively provides more relevant results and has great scope of future development in the field of text mining.

Keywords:
Term Frequency-Inverse Document frequency (TF-IDF), Latent Semantic Indexing (LSI),
Singular Value Decomposition (SVD),Text Mining.

1. Introduction

Based on the kind of task to be performed the text based system fulfillment are done by the representation of documents and appropriate representation. Except, some lexical and general grammar no other special requirements are used in the collection of unstructured documents which is dealt by text mining . In which text representation i.e. the transformation of text is into numerical data is one of the main themes supporting text mining. In information retrieval, usually the identification of the terms and keywords that are used to represent the document content collectively for stored documents and records. One of the most widely used models for representation is the Vector space model(VSM) mainly because of its conceptual simplicity and the very replacement of using spatial proximity for semantic proximity. Generally, there are two kinds of works involved in text representation: indexing and term weighting. The assignment of indexing term to the documents is done by Indexing. We should clarify here that in this paper, we will be discussing the problem as level of text representation and not discuss distinctively the efficiency of indexing and term weighting. Normally, the predefinition of the index terms is the fixed set which is controlled vocabulary indexing and any other words indexers regard them in relation to the topic document which is free indexing. The usage of natural indexing and computer selection of indexing terms has increased to a great level as more and more texts are being available. The measurement of the importance of terms in documents is done by Term weighting whose job is to assign the weights of terms. The different assumptions of terms characteristics or behaviors in texts derives various many term weighting methods even today. For example, IDF (inverse document frequency) holds the assumption that the significance of a term is inversely proportional to the frequency of occurrence of this term in all the documents and RIDF (residual inverse documents frequency) holds the supposition that the importance of a term should be measured by the difference between the frequency of actual occurrence in all the documents and the predicted frequency of occurrence by Poisson distribution (random occurrence). Fundamentally, the information retrieval (IR) and text categorization (TC) which are the inclusive part of text classification which are mainly concerned with two kind of properties of the indexing term: semantic quality and statistical quality [3]. To how much extent the index term represents the text content is semantic quality which is related to the meaning of the index term contain; statistical quality is related with the discriminative (resolving) power of the index term to discriminate the document it belongs to from other texts in the collection.

The motivation of this research is to study the accomplishment of text classification of different representation methods which are developed from different essential hypotheses concerning indexing and term weighting. Based on the insight for text representation, multi-word, which is a greater lexical unit than individual word and is anticipated that have both semantic quality and statistical quality, is proposed as a competitive index term.

2. Query Retrieval Problem

Nowadays retrieving data on the basis of user-defined query has become a very common task. Since it has a growing number of users who use a query retrieval .This has lead to a tremendous increase in research and development of algorithms which generates appropriate solutions to the problem. Informally , retrieving data based on user query can be described query retrieval can be described as the job of searching a collection of data ,let that be text documents ,databases, networks, etc., for specific instances of that data. First, we will only work on collection of English documents. The refined problem then becomes the task of searching this body for documents that the query retrieval system considers relevant to what the user entered as the query.

Let us describe this problem more formally. We have a set of documents D, with the user entering a query $q = w_1, w_2, ., w_n$ for a sequence of words w_i. Then we wish to return a subset D^* of D such that for each $d \in D^*$, we maximize the following probability :

$$P\ (d \mid q, D)\ (1)$$

(Berger & Lafferty, 1999). As the above notation suggests, numerous approaches to this problem involve probability and statistics, while others propose vector based models to enhance the retrieval.

3. LSI

LSI (Latent Semantic Indexing) [8] is regarded as one of the most popular linear document indexing methods which uses word co-occurrence which could be regarded to produce low dimensional representations between the terms. Document indexing process further raises its importance by adding in it Latent Semantic Indexing. Moreover to keeping the record of as to which keywords a document contains, the method investigates the document as a whole to check which other documents contain some of those words. LSI takes under consideration two types of documents that have ample of words in common to be semantically close and the other ones with less words in common to be semantically distant. The correlation of this simple method can be done to a human being who as to how after looking at the content, he would manually might classify the document. Even if the LSI algorithm doesn't comprehend anything about what the words *mean*, the patterns it notices can make it seem astoundingly intelligent.

When an LSI-indexed database is searched, before returning the document that it thinks its best fits the query the search engine looks at similarity values it has calculated for every content word. LSI does not require an exact match to return useful results as two documents can be semantically really close even though they do not share a particular keyword .LSI will often return all relevant documents that don't contain the keyword at all, wherever there is a failure in the plain keyword if there is no exact match.

To use an earlier instance, let's say we use LSI to index our collection of mathematical articles. If the words n-dimensional, numerous and topology appear together in enough articles, the search algorithm will notice that the three terms are semantically close. It is possible that a search for n-dimensional manifolds will hence retrieve a set of articles consisting that phrase (the same result we would get with a regular search), but also articles that consist just the word topology. As examining a sufficient number of documents teaches it that the three terms are related but search engine does not have any understanding or recognizes anything about mathematics. Further, It then uses that information to provide an expanded set of results with better recall than a plain keyword search.

In the sense of minimizing the global reconstruction error (the Euclidean distance between the original matrix and its approximation matrix) the LSI has its aim to find the best subspace approximation to the original document. It is significantly has its basis on SVD (Singular Value Decomposition) and projects the document vectors into the subspace so that cosine similarity can be represented accurately in semantic similarity.

Given a term-document matrix m X [x_1, x_2 ,..., x_n]$\in R^m$ and suppose the rank of X is r, LSI decomposes X using SVD as follows:

$$\boldsymbol{X= U\sum V^T}$$

where *diag* (σ_1 ,..., σ_r) and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ are the singular values of X. U [u_1 ,.., u_r] and u_i is called the left singular vector. V [v_1,..., v_r] and v_i is called the right singular vector. LSI uses the first k vectors in U as the transformation matrix to embed the original documents into a k-dimensional space.

There are also some disadvantages of LSI method. The first one is that there are some negative values in the reconstruction matrix we cannot give a reasonable explanation. It also has a huge computation as O(n^2r^3), where n is the smaller of the number of documents and the number of terms, r is the rank of X [7].

4. TF-IDF

As the name suggests, TF-IDF calculates values for each term (user query) in a document through an inverse proportion of the frequency of the term in a particular document to the number of words that appear in the document.

TF*IDF is evolved from IDF which is proposed by Sparck Jones [4, 5] with the thinking that if a query term which occurs in document many number of times may not provide relevant results and the documents containing less occurrences are not relevant .Equation given below is the classical formula of TF*IDF used for term weighting.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Where w $_{i,j}$, is the weight for i^{th} term in j^{th} document, N is the number of documents in the collection, $tf_{i,j}$ is the term frequency of i^{th} term in j^{th} document and df_i is the document frequency of i^{th} term in the collection.

The basis of TF*IDF is from the theory of language modeling that the terms present in a given document can be categorized into with and without the property of eliteness, i.e., the term is about the topic of the given document or not. The eliteness of a term for a given document can be calculated by TF and IDF is used for the measure of significance of this term in the collection.

4.1 Mathematical Framework

Here is a quick informal explanation of TF-IDF before we begin. Essentially, TF-IDF works by determining the relative frequency of words (user query) in a particular document and comparing it to the inverse proportion of that word (user query) over the entire document body. This calculation helps us in determining that the given word is how much relevance in that given document. Words that are common in a single or a small group of documents tend to have higher TF-IDF numbers than common words such as articles and prepositions.

The formal procedure for implementing TF-IDF has some negligible differences over all its applications, but the overall approach works as follows. Given a document *D*, a word *w*, and an individual document $d \in D$, We calculate

$$w_d = f_{w,d} * \log(|D|/f_{w,D}) \text{ (2),}$$

where, $f_{w,d}$ equals to the number of times *w* appears in *d*, |*D*| is the size of the body, and $f_{w,D}$ equals the number of documents in which *w* appears in *D*. There are a some different scenarios that can occur here for each word, depending on the values of $f_{w,d}$, |*D*|, and $f_{w,D}$, the most prominent of which we will examine.

Assume that $|D| \sim f_{w,D}$, i.e. the size of the body is approximately equal to the frequency of *w* over *D*. If $1 < \log(|D|/f_{w,D}) < c$ for some very small constant *c*, then w_d will be smaller than $f_{w,d}$ but still positive. This implies that *w* is relatively common over the entire corpus but still holds some importance throughout *D*. For example, if TF-IDF would examine the word ‘Krishna’ over the Vedas. More relevant to us, this result would be expected of the word ‘World’ in the body of World Health organization documents. This is also the problem with most of the common words such as articles, pronouns, and prepositions, which by themselves hold no significant meaning in a query (unless the user explicitly wants documents containing such common words). Such common words thus are allocated a very low TF-IDF score, not giving them much importance in the search.

Finally, suppose $f_{w,d}$ is large and $f_{w,D}$ is small. Then log ($|D|/f_{w,D}$) will be rather large, and so w_d will likewise be large. This is the scenario in which we are most interested, since words with high w_d

imply that w is an important word in *d* but not common in *D*. This *w* term is said to have a large discriminatory power. Therefore, when the user mentions w in the query, providing a document *d to the user* where w_d is large will very likely persuade the user

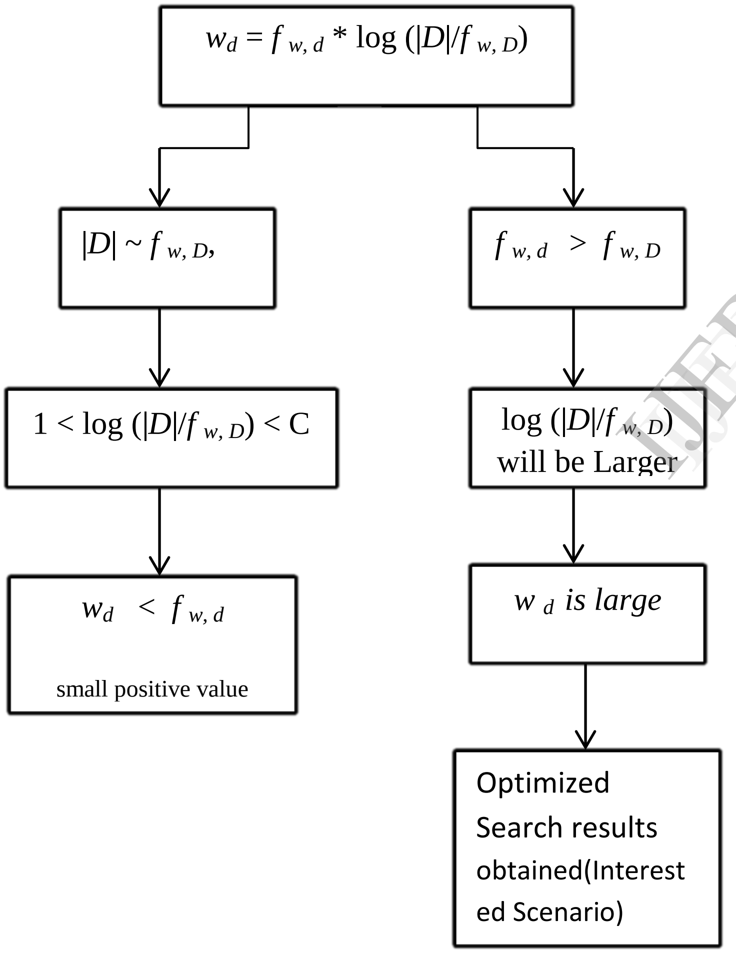
$w_d = f_{w,d} * \log(|D|/f_{w,D})$
$|D| \sim f_{w,D},$
$f_{w,d} > f_{w,D}$
$1 < \log(|D|/f_{w,D}) < C$
$\log(|D|/f_{w,D})$ will be Larger
$w_d < f_{w,d}$
small positive value
w_d *is large*
Optimized Search results obtained(Interested Scenario)

Figure 1. Mathematical Framework for TF-IDF

4.2 Encoding TF-IDF

The code for TF-IDF is well-designed in its straightforwardness. Given a query *q* composed of a set of words w_i, we calculate $w_{i,d}$ for each w_i for every document *d* є *D*. In the simplest way, this can be done by scanning the document collection and keeping a running sum of $f_{w,d}$ and $f_{w,D}$. Once done, we can easily calculate $w_{i,d}$ according to the mathematical framework presented before. Once all $w_{i,,d}$ are found, we return a set *D** containing documents *d* such that we maximize the following equation:

$$\Sigma_i \, w_{i,d} \ (3).$$

Either the user or the system can arbitrarily determine the size of *D** prior to initiating the query. Also, documents are returned in a decreasing order according to equation (3).This is the conventional method of implementing TF-IDF

5. Conclusion

The term vectors require the storage of roughly 400,000 additional values. Moreover, the values of LSI are real numbers while original term frequencies are integers which is addition to the storage costs. The fact that each term occurs in a limited number of documents can no longer be taken advantage of when using LSI vector, which accounts for the sparse nature of the term by document matrix. The storage requirements of LSI are not a critical problem, but the loss of sparseness has other, more serious inferences with recent advances in electronic media storage. Using an inverted index is one of the most significant speeds-up in vector space search. As a result, only documents that have some terms in common with the query must be studied during the search. However, the query must be correlated to every document in the collection with LSI. Thus, we see that TF-IDF is better than LSI .We have seen that TF-IDF is an efficient and simple algorithm for matching words in a query to documents that are relevant to that query. From the data collected, we see that TF-IDF returns documents that are highly relevant to a particular query. If a user were to input a query for a particular topic, TF-IDF can find documents that contain relevant information on the query. Furthermore, encoding TF-IDF is straight forward, making it ideal for forming the basis for more complicated algorithms and query retrieval systems .This algorithm is inclusive of hill-climbing and gradient descent to enhance performance. They also have put forward an algorithm for performing TF-IDF in a cross-language retrieval setting by application of statistical translation to the benchmark of TF-IDF. Future research might also incorporate the employment of TF-IDF to performing searches in documents written in a different language than the query. Enhancement of the already powerful TF-IDF algorithm would increase the success rate of query retrieval systems, which have already has quick graph risen upwards to become a key element of present global information exchange.

REFRENCES:

[1]Berger, A & Lafferty, J.(1999). Information Retrieval as Statistical Translation. In Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR’ 99), 222-229.

[2] G. Salton and C. S. Yang, “On the Specification of Term Values in Automatic Indexing,” Journal of Documentation, 29(4), 1973, pp. 351-372

[3] M. G. H. Jose, “Text representation for automatic text categorization,” Online:http://www.esi.uem.es/~jmgomez/tutorials/eacl03/slides.pdf.

[4] K.Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” Journal of Documentation, 28, 1972, pp. 11-21.

[5] K. Sparck Jones, “IDF term weighting and IR research lessons,” Journal of Documentation, 60(6), 2004, pp. 521-523.

[6]Brown, Peter F. et al. (1990). A Statistical Approach to Machine Translation. In Computational Linguistics 16(2): 79-85

[7]Littman, M., & Keim, G. (1997). Cross-Language Text Retrieval with Three Languages. In CS-1997-16, Duke University.

[8] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A.
Harshman, “Indexing by Latent Semantic Analysis,” Journal of American Society of Information Science, 41(6), 1990, pp. 391-407.