Efficient Use Of Side Information For Mining Text Data With TF-IDF

[1]M Mintu, [2]Varsha Philip
[1]P.G. Scholar, Department of CSE , MIT Anjarakandy, Kannur, Kerala, India
[2]Guide, Department of CSE , MIT Anjarakandy, Kannur, Kerala, India
[1]mintupramod@gmail.com, [2]varshaphilip88@gmail.com

Abstract-- **Text mining is the process of deriving high quality information from text. In such applications side information is embedded with text documents. It contains a vast majority of information that enhanceclustering approach. The use of side information may become inefficient when some of the data are erroneous. In such cases, it can be risky to use the side-information into the mining process, because it may either destroy the quality of the collection of data for the mining process, or may add noise to the process. Therefore, we need some advanced efficient way to perform such mining process, so as to increase the capabilities and advantages from using this side information. So the mining process must be carried out in a proper way so as to make use of the side information. Besides to the existing side information like links in the document, user-access behavior from web logs, metadata, this paper proposes a method to mine text data using TF-IDF. It is a numerical static that is intended to reflect how important a word is to a document in a corpus. In this paper we design a distance based clustering algorithm with vector space inorder to create an efficient clustering approach using TF-IDF. We then present how this can add-on to classification problem.**

***Index Terms*— Text mining, Side Information, Clustering, Classification.**

I. INTRODUCTION

A vast majority of data are stored in document and they are increasing day by day.So text mining is a technique used to extract information from text documents. The text clustering becomes an issue in many type of application domains such as web, social network etc. However in many applications a vast amount of side information is embedded with the documents. Side information provide a huge amount of information that can enhance clustering process.The problem of text clustering has been studied in [7], [8], [9].

[1] Some examples of such side informations are:
[2]User acces web documents
[3] Keep a web log based on user acces behavior and based on that behavior datamining process can be implemented.
[4]Links in text documents
[5] Links can also be treated as attributes. It also contain information about datamining process.
[6]Meta data

Other information or meta data is associated with documents. They are also informative for mining process.

Such side-information are sometimes useful in improving the quality of the mining process, it may be a risky technique when the side-information contains noises. In such cases, it can actually decrease the quality of our process. Therefore, we will use an approach which carefully maintains the structure of the clustering characteristics of the side information with that of the text content. This helps us magnifying the clustering result of both kinds of data.

$$TF(t,d) = \frac{Nooftimestappearsinadocument}{Totalnooftermsinthedocument} \qquad (1)$$

The IDF is a measure of how much information the word provides, i.e whether the term is common or rare across all documents.It is calculated as:

$$IDF(t,D) = \log \frac{Total\ no\ of\ docs\ in\ a\ corpus}{No\ of\ docs\ matching\ term} \qquad (2)$$

Then the TF-IDF is calculated as:

$$TF - IDF(t,d,D) = TF(t,d) * IDF(t,D) \qquad (3)$$

Essentially, TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document.

The main goal of this paper is to mine text data using TF-IDF besides from the existing side information attributes. The TF-IDF is a numerical static that is intended to reflect how important a word is to a document in a corpus. It is often used as a weighting factor in information retrieval and text mining. The TF-IDF value increases proportionally to the number of times a word appears in the document. The TF-IDF is calculated based on pattern matching algorithm. Then a distance based clustering algorithm [3] with vector space is used inorder to enhance the quality of clustering process.

While this side information can be useful in enhancing the clustering process, this may become dangerous when the side information is erroneous. In such a case it can degrade the mining process. So the mining process must be carried out in proper way so as to maximize the benefits from using this side information.

II.RELATED WORK

A general survey of clustering algorithm is studied in [5]. The problem of clustering has also been found in the context of text data. A comparative different document clustering methods may be found in [6]. Detailed surveys on text classification has been found in [3].

Many forms of text database contain a large amount of side information which may be used inorder to improve the clustering process.

Some example of existing side attributes are:

User acces web documents

In an application, user access behaviour may be captured in the form of web logs. It is used to indicate whether or not the documents has been accessed by a user. It correspond to the mining behavior of different users.

Links in text documents

Links can also be treated as attributes. It also contain information about mining process.

Meta data

Other information or meta data is associated with documents. In a document application which are associated GPS or provenance information, the possible attributes may be drawn on a large number of possibilities. Such attributes will certainly satisfy the sparsity property.

The issues of text-clustering has been studied widely by the database community [7], [8], [9]. The major point of this work has been on scalable clustering of multi dimensional data of various types [7], [13], [8], [9]. A general survey of clustering algorithms may be specified in [5]. The issues of clustering has also been researched quite extensively in the place of text-data. A survey of text clustering methods may be located in [2]. One of the most well known methods for text-clustering is the scatter-gather technique [16], which uses a collection of agglomerative and partitional clustering. Other similar methods for text-clustering which use same methods are discussed in [15]. Co-clustering methods for text data are proposed by authors of [11], [12]. An Expectation Maximization (EM) technique for text clustering has been focused in [10].

The first approach for mining text data with the use of side information presented in [4]. But it is limited to clustering approach only. And in [1] the method is extended to the problem of text classification. An efficient clustering and classification approach can be enhanced by mining text data with TF-IDF besides in [1]. Such an approach is useful, when the meta information is highly informative.

III.PROPOSED SYSTEM

This paper proposed method for mining text data with side information using TF-IDF. Other than the existing side information like links in the documents, user access web logs, meta data etc. TF-IDF is also added in order to improve the quality and also the performance of clustering. The term frequency-inverse document frequency is calculated based on pattern matching algorithm. Then a distance based clustering algoritham with vector space is used inorder to provide efficient clustering process. After that we show, how this approach can add-on to classification problem.

A.Clustering with TF-IDF

In this section, we describe how we can cluster text data by using TF-IDF besides other side infromation like links in documents, user access behavior from web logs and meta data.

Charu C Agarwal et.al assumed [1] that there a corpus M of text documents. The total no of documents is N and they are denoted by $T_1 \ldots . T_n$.. The set of distinct words in the entire corpus M is assumed to be L. So there is a side attribute $\overline{Xi}$ associated with each document. Each $\overline{Xi}$ has dimension d.

So as noted in the above example, side information is available with all documents and it is sparse. Besides from the above side information, we also provide mining text data with TF-IDF.

- Text Clustering with TF-IDF

In a document application associated with specific term frequency, we count the no of times each term occur by using a paatern matching algorithm and cluster them based on distance based clustering algorithm using vector space.

B.Cwtf-Idf Algorithm

In this section, we will describe our algorithm for text clustering with TF-IDF. The algorithm is referred to as clustering with TF-IDF. Here the stopping words and erroneous datas have been removed.This algorithm has two phases:

Initialization: Without using side information initialize vector space as ϕ.

Clustering Phase: Assign values to vector space by TF-IDF. It is calculated based on pattern matching algorithm. And it iteratively constructs the clusters using distance based clustering algorithm.

The focus of the first phase is simply to construct an empty vector space.

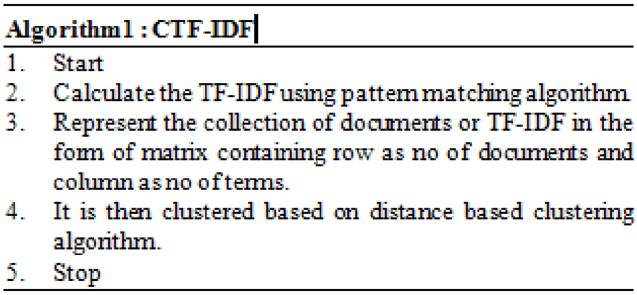
The TF-IDF is calculated based on pattern matching algorithm. It returns the percentage of matching terms. It is then represented in vector space. A vector space is a model used for representing collection of documents in the form of matrix containing row as no of documents and column as no of terms.

The final step is text clustering based on distance based clustering algorithm. It is designed by using a similarity function to measure the closeness between the text objects. So the similarity function which is used in text domain is the cosine similarity function.

Let $U = f(u_1) \ldots f(u_k) \& V = f(v_1) \ldots f(v_k)$ be the frequent term vector in two different documents$U \& V$. Also the values $u_1 \ldots u_k$ & $v_1 \ldots v_k$ be the term frequency.

Then cosine similarity between two document is calculated as:

$$cosine(U,V) = \frac{\sum_{i=1}^{k} f(u_i).f(v_i)}{\sqrt{\sum_{i=1}^{k} f(u_i)^2}.\sqrt{\sum_{i=1}^{k} f(v_i)^2}} \qquad (4)$$



```
Algorithm1 : CTF-IDF
1.  Start
2.  Calculate the TF-IDF using pattern matching algorithm
3.  Represent the collection of documents or TF-IDF in the
    form of matrix containing row as no of documents and
    column as no of terms.
4.  It is then clustered based on distance based clustering
    algorithm.
5.  Stop
```

A.Classification Approach

In this section we will focus how to add-on the approach to classification. As before, Charu C Agarwal et.al assumed [1] that there a corpus M of text documents. The total no of documents is N and they are denoted by $T_1 \ldots .T_n$. A training label $\overline{l_i}$ is associated with each document T_i. So there is a side attribute $\overline{Xi}$ associated with each training document

B. Cltf-Idf Algorithm

We refer to the algorithm as CLassification with TF-IDF. The algorithm has three phases:

- Feature Selection:

In this phase,feature selection is performed inorder to remove stopping words and those terms which are not related to vector space from documents.

- Initialization:

In this step, initialize a vector space with previous values of vector space.

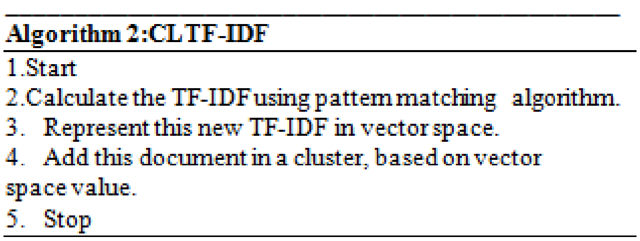
- Classifying Phase:

In this phase, a combination of text and side information as in [1] and also TF-IDF is used for the classification purposes.

The most common feature selection which is used in both supervised and unsupervised applications is that of stop-word removal and stemming. In stop-word removal, we determine the common words in the documents which are not specific or discriminatory to the different classes. In stemming, different forms of the same word are consolidated into a single word. While feature selection is important to remove irrelevant features in the document.

So once the supervised clusters have been created, they are used for the process of classification. If new document arrives, it is compared with existing vector space. The unrelated terms and stopping words are removed from the documents.

If already classification is performed, then initialize vector space with final value of vector space in classification. If not, then initialize vector space with final vector space of clustering.



```
Algorithm 2:CLTF-IDF
1.Start
2.Calculate the TF-IDF using pattern matching  algorithm.
3.  Represent this new TF-IDF in vector space.
4.  Add this document in a cluster, based on vector
space value.
5.  Stop
```

IV.EFFICIENCY IMPROVEMENT

For mining a data in existing system $n+k$ comparison is required.

Consider the following variables,

N : Total number of documents

k : Number of clusters in existing method

n : Average number of documents in a cluster for existing method

k' : Number of clusters in proposed method

n' : Average number of documents in a cluster for proposed method

Now,

$$n = \frac{N}{k} \text{ and } n' = \frac{N}{k'}$$

$$k \le k' \text{ and } n \ge n'$$

For mining a data in proposed system only $n'+k'$ comparisons required. For higher values of N, $n'+k' \leq n+k$. ie. the performance of existing can be increased by applying proposed concept with it.

V.CONCLUSION

In this paper, we presented a method to mine text data using side information with TF-IDF. Many forms of side informations are available with text documents. Inorder to design a clustering process, we combined vector space with distance based clustering algorithms. TF-IDF is an efficient and simple algorithm for matching words in a documents. The use of TF-IDF can greatly enhance the the quality of text clustering and classification, while maintaining a high level of efficiency than existing.

REFERENCES

[1] Charu C. Aggarwal, Yuchen Zhao, and Philip S. Yu, Fello On the Use of Side Information for Mining Text Data, IEEE Transactions on knowledge and data engineering vol 26,no.6 pp 1415-1429,2014

2] C. C. Aggarwal and C.-X. Zhai, Mining Text Data. New York, NY, USA: Springer, 2012.

[3] C. C. Aggarwal and C.-X. Zhai, A survey of text classification algorithms, in Mining Text Data. New York, NY, USA: Springer, 2012.

[4] C. C. Aggarwal and P. S. Yu, On text clustering with side information, in Proc. IEEE ICDE Conf., Washington, DC, USA, 2012.

5] A. Jain and R. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1988.

[6] M. Steinbach, G. Karypis, and V. Kumar, A comparison of document clustering techniques, in Proc. Text Mining Workshop KDD, 2000, pp. 109-110.

[7] S. Guha, R. Rastogi, and K. Shim, CURE: An efficient clustering algorithm for large databases,• in Proc. ACM SIGMOD Conf., New York, NY, USA, 1998, pp. 73-84.

[8] R. Ng and J. Han, Efficient and effective clustering methods for spatial data mining, in Proc. VLDB Conf., San Francisco, CA, USA, 1994, pp. 144-155

[9] T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH: An efficient data clustering method for very large databases, in Proc. ACM SIGMOD Conf., New York, NY, USA, 1996, pp. 103-114.

[10] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma, An evaluation of feature selection for text clustering, in Proc. ICML Conf., Washington, DC, USA, 2003, pp. 488-495.

[11] I. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in Proc. ACM KDD Conf., New York, NY, USA, 2001, pp. 269-274.

[12] I. Dhillon, S. Mallela, and D. Modha, Information-theoretic co-clustering, in Proc. ACM KDD Conf., New York, NY, USA, 2003, pp. 89-98.

[13] S. Guha, R. Rastogi, and K. Shim, ROCK: A robust clustering algorithm for categorical attributes, Inf. Syst., vol. 25, no. 5,pp. 345-366, 2000.

[14] T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH: An efficient data clustering method for very large databases,• in Proc. ACM SIGMOD Conf., New York, NY, USA, 1996, pp. 103-114.

[15] H. Schutze and C. Silverstein, Projections for efficient document clustering, in Proc. ACM SIGIR Conf., New York, NY, USA, 1997, pp. 74-81.

[16] D. Cutting, D. Karger, J. Pedersen, and J. Tukey, Scatter/Gather: A cluster-based approach to browsing large document collections,• in Proc. ACM SIGIR Conf., New York, NY, USA, 1992, pp. 318-329.