IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 11, November 2015

Enhancing Text Mining Using Side Information

Prof. P. S. Toke[1], Rajat Mutha[2], Omkar Naidu[3], Juilee Kulkarni[4]

Professor, Computer Engg, Department, PVPIT, Pune, India [1]

Student, Computer Engg, Department, PVPIT, Pune, India [2, 3, 4]

Abstract: Clustering is a widely studied data mining problem in the text domains. This problem finds numerous applications in classification, visualization, document organization, collaborative filtering and indexing. Large quantity of information from document is present in the form of text. Data is not purely available in text form. It also contains a lot of Side Information, can be different kinds of link in the document, user-access behaviour, document provenance information from web - logs or other non-textual attributes. These attributes may contain large amount of information in the clustering purposes. However, it is difficult to estimate the relative information, when some of information is noisy data. In such situation, it will be risky to integrate this side-information into the mining process, because it can add noise to the process or improve the quality of the illustration for the mining process. An ethical way is needed to perform the mining process, and to maximize the advantages of using this available side information. In this paper, we propose the use of K-means algorithm for better and efficient clustering of the information.

Keywords: Clustering, Information-Retrieval, K-means, Side Information, Text mining.

I. INTRODUCTION

The problem of text clustering arises in the context of many application domains such as the web, social networks, and other digital collections. Auxiliary information is known as Side Information or meta-data which is available with text document in several text mining application. Links in the document, Document provenance information, and other non-textual attributes which are contained in the document and web logs are the different kind of side information.

1. Information Retrieval –

Information retrieval (IR) is the activity of obtaining information resources relevant to an information need from a collection of information resources. Automated information retrieval systems are used to reduce what has been called "information overload". Web search engines are the most visible IR applications. An information retrieval process begins when a user enters a query into the system. Queries are formal statements of information needs, for example search strings in web search engines.

2. Text Mining-

Text mining refers to the process of deriving high-quality information from text. Text mining usually involves the process of structuring the input text, deriving patterns within the structured data, and finally evaluation and interpretation of the output. Typical text mining tasks include text categorization, text clustering, concept extraction, document summarization, and entity relation modelling. A typical application is to scan a set of documents written in a natural language and either model the document set for predictive classification purposes or populate a database or search index with the information extracted.

3. Side Information

In many application domains, a tremendous amount of side information is also associated along with the documents. This is because text documents typically occur in the context of a variety of applications in which there may be a large amount of other kinds of database attributes or meta-information which may be useful to the clustering process. Some examples of such side-information are as follows:

- In an application in which we track user access behaviour of web documents, the user-access behaviour may be captured in the form of web logs. Such logs can be used to enhance the quality of the mining process in a way because the logs can often pick up subtle correlations in content, which cannot be picked up by the raw text alone.
- Many text documents contain links among them, which can also be treated as attributes. Such attributes may often provide insights about the correlations among documents in a way which may not be easily accessible from raw content.
- Many web documents have meta-data associated with them such as ownership, location, or even temporal information may be informative for mining purposes.

4. Clustering

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group called a cluster, are more similar to each other than to those in other clusters. It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Clustering can be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure.

Copyright to IJARCCE
DOI 10.17148/IJARCCE.2015.411116

II. LITERATURE REVIEW

Text clustering has been studied widely by the database community the major focus of this work has been on scalable clustering of multi-dimensional data of different a general survey of clustering algorithms may be found. The problem of clustering has also been studied quite extensively in the context of text data. A survey of text clustering methods may be found in. One of the most well-known techniques for text-clustering is the scatter gather technique [2] which uses a combination of agglomerative and partitioned clustering. Other related methods for text clustering which use similar methods are discussed. Co-clustering methods for text data are proposed in [3]. An Expectation Maximization (EM) method for text clustering has been proposed. Matrix-factorization techniques for text clustering are pro-posed in [4]. Clustering is applied on text as well as auxiliary attributes by using COATES algorithm and classification methods across many baseline techniques on real and scientific datasets.

However, such algorithms significantly increase the time and space complexity without any justified increase in the output. For proper clustering of text data, Natural Language Processing techniques are required which increase the complexity further. On the internet, the amount of data to be processed can be near infinite. In order to be able to present results in a reasonable time we propose a simple approach to obtaining relevant results using k-means clustering and TF-IDF. This algorithm works better than existing ones because it builds upon already existing text retrieval apps such as Google search.

III.OBJECTIVE AND STUDY

Our objective will be to remove noisy information by filtering links and to organize the information in Partition Clusters. Our working application should suggest recommendations to users.

IV.DATA AND METHODOLOGY

Our approach involves building upon existing search engines like Google, Bing, Yahoo and many others which return web documents using various approaches of identifying relevant information from the observable internet. After obtaining these links (which will be referred to as original links henceforth), we attempt to remove noisy information which may not be relevant and identify additional links and web documents which may be appropriate. Data mining algorithms for clustering are one of the best ways to obtain related information, i.e., grouping similar objects together. For text clustering, the actual text needs to be converted in a form suitable for clustering. The general scheme is as follows:

A. Initialization Phase
B. Main Phase
C. Result Generation Phase

A. Initialization Phase

Step 1: The documents refer to the text that is obtained from web documents. These are obtained by doing a simple keyword search on a search engine. These documents are our original seeds. The additional documents which may be relevant are obtained by recursively obtaining web links from each of the original document contents.

Step 2: The documents that have been obtained cannot be use directly for clustering. This is because natural language contains a lot of redundant and unnecessary words which can mislead algorithms which prioritize word count and word weight. Stopping is applied to filter unwanted information, redundant words such as “is”, “am”, “are”, “there”, “when”, “then” etc. Stemming is the process of converting words into their root form. For example, words such are “driver”, “driving” are converted into their root, “drive”.

Step 3: Such filtered documents are saved separately for processing in the main phase. Each filtered document is collected and passed to the main phase.

B. Main Phase

Step 1: Filtered documents are treated one by one for applying text mining techniques i.e. Shared Word Count, Word Count Bonus (to calculate weight of words), similarity measures. Similarity measures which are considered are cosine similarity and Pearson measure. Other distance based measures like Euclidean distance or Minkowski distance are used to find out the distance in terms of similarity between different documents.

Step 2: Step 1 gives the list of attributes and base for clustering and classification of documents. All the filtered documents are reiterated to decide their clusters using weighted characteristics and similarity measures. K-means clustering is the primary way to group objects into different clusters. Nearest neighbour technique is also applied before assigning any document to a particular cluster. Each technique has some weight and sum of the weights is used for finalizing the documents. This step forms the core process of the working of the system. A detailed explanation of how k-means algorithm works is given later.

Step 3: Clustering is applied incrementally and hence performance of the method shall be high.

C. Result Generation Phase

Evaluation of the implemented system shall be done using various data mining measuring features such as accuracy, sensitivity, f-measure etc. Precision and Recall are two important measures of accuracy in any information retrieval algorithms. Precision gives us the number of retrieved documents that are actually relevant. Recall gives us the total number of relevant documents which were retrieved. While calculating the precision, we need to identify which documents might be considered relevant by the user. Recall may consider the original links and their sub links as the total number of documents to be considered. Results shall be compared with the results of original COATES algorithm in base paper.

V. MODELLING

K-means-

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well-

DOI 10.17148/IJARCCE.2015.411116

known clustering problem. The procedure follows a simple and easy way to classify a given data set. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. The better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycentres of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

where |xi-cj| is a chosen distance measure between a data point xi and the cluster centre cj, is an indicator of the distance of the n data points from their respective cluster centres.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move.

This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

TF-IDF-

TF-IDF stands for term frequency-inverse document frequency, and the TF-IDF weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the TF- IDF weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

How to Compute:

Typically, the TF-IDF weight is composed by two terms: the first computes the normalized Term Frequency (TF), a.k.a.the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

TF: Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

IDF: Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

VI. CONCLUSION

The process of text mining and retrieving information can be greatly enhanced by using an approach that can be effective as well as fast. Computer science is a field which builds and iterates over previously done work to make it more effective. We suggest a similar approach where processing the results obtained by using a search engine can help us get more appropriate results, faster. Out of the many different methods and algorithms available for reducing the dimensionality of text and clustering, we chose k-means and a bag-of-words approach over more advanced approaches involving Natural Language Processing or Artificial Intelligence. The reason for this is that the original documents that we receive already serve as a good base for relevant information. Using our approach simply acts as an additional layer to ensure quality.

ACKNOWLEDGMENT

We are glad to express our gratitude to all who rendered their valuable guidance to us. We would like to express our appreciation and thanks to our guide **Prof. P. S. Toke**. We are also thankful to **Dr. K. N. Barbole,** Principal, PVPIT, Pune and **Prof. N. D. Kale,** Head of Department, Computer Engg., PVPIT, Pune. We thank the anonymous reviewers for their comments.

REFERENCES

[1] Charu C. Aggarwal, Yuchen Zhao, Philip S. Yu., On the use of Side Information for Mining Text Data, IEEE Transactions, Vol.26, No.6, JUNE 2014.

[2] H. Frank, "Shortest paths in probabilistic graphs," Operations Research, vol. 17, no. 4, pp. 583-599, 1969.

[3] L. G. Valiant, "The complexity of enumeration and reliability problems," SIAM J. Comput., vol. 8, no. 3, pp. 410-421, 1979.

[4] N. J. Krogan, G. Cagney, and al., "Global landscape of protein complexes in the yeast saccharomyces cerevisiae," Nature, vol. 440, no. 7084, pp. 637-643, March

[5] S. Guha, R. Rastogi, and K. Shim, CURE: An efficient clustering algorithm for large databases, in Proc. ACM SIGMOD Conf.,

DOI 10.17148/IJARCCE.2015.411116

[6] R. Ng and J. Han, Efficient and effective clustering methods for spatial data mining, in Proc. VLDB Conf., San Francisco, CA, USA, 1994, pp. |144155.

[7] 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. 103114.

[8] Lei Meng, Ah-Hwee Tan, Dong Xu Semi-Supervised Heterogeneous Fusion for Multimedia Data Co-Clustering IEEE Transactions On Knowledge And Data Engineering, vol. 26, no. 9, pp.2293-2306, 2014.

[9] Vishal Gupta, Gurpreet S. Lehal, A Survey of Text Mining Techniques and Applications , in Journal of Emerging Technologies in Web Intelligence, Vol. 1, No. 1, August 2009, pp.60-76

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

[11] I. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in Proc. ACM KDD Conf., New York, NY, USA, 2001, pp. 269274.

[12] 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. 488495.

[13] R. Shamir, R. Sharan, and D. Tsur, "Cluster graph modification problems," Discrete Applied Mathematics, vol. 144, no. 1-2, pp. 173-182, 2004.

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

DOI 10.17148/IJARCCE.2015.411116