INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 4 ISSUE 2 – APRIL 2015 - ISSN: 2349 - 9303

Effective Feature Selection for Mining Text Data with Side-Information

Divya P[1]
[1]Department of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore, Tamil Nadu, India.
divyap44@gmail.com

G.S. Nanda Kumar[2]
[2]Department of Computer Science and Engineering,
Kumaraguru College of Technology,
Coimbatore, Tamil Nadu, India.
nandakumar.gs.cse@kct.ac.in

Abstract— Many text documents contain side-information. Many web documents consist of meta-data with them which correspond to different kinds of attributes such as the source or other information related to the origin of the document. Data such as location, ownership or even temporal information may be considered as side-information. This huge amount of information may be used for performing text clustering. This information can either improve the quality of the representation for the mining process, or can add noise to the process. When the information is noisy it can be a risky approach for performing mining process along with the side-information. These noises can reduce the quality of clustering while if the side-information is informative then it can improve the quality of clustering. In existing system, Gini index is used as the feature selection method to filter the informative side-information from text documents. It is effective to a certain extent but the remaining number of features is still huge. It is important to use feature selection methods to handle the high dimensionality of data for effective text categorization. In the proposed system, In order to improve the document clustering and classification accuracy as well as reduce the number of selected features, a novel feature selection method was proposed. To improve the accuracy and purity of document clustering with less time complexity a new method called Effective Feature Selection (EFS) is introduced. This three-stage procedure includes feature subset selection, feature ranking and feature re-ranking.

Index Terms— Effective Feature Selection (EFS), feature subset selection, feature ranking and feature re-ranking, Side-information.

1 INTRODUCTION

THE use of digital information is increasing day-by-day. Since increasing the amount of information it needs to extract relevant information for text mining. Thus there are tremendous amount of mining algorithms. Till now these mining algorithms use only the pure data but not the additional information. In order to handle such huge amount of data, need to index the data according to the users need. Meta-data will be used for this and such meta-data is the side-information which is present on most of the text documents. Many web documents consist of meta-data with them. These meta-data correspond to different kinds of attributes such as the source or other information related to the origin of the document. Data such as location, ownership or even temporal information may be considered as side-information. This huge amount of information may be used for performing text clustering. This information can either improve the quality of the representation for the mining process, or can add noise to the process. When the information is noisy it can be a risky approach for performing mining process along with the side-information. These noises can reduce the quality of clustering while if the side-information is informative then it can improve the quality of clustering. The primary goal of this paper is to improve the accuracy of document clustering along with side-information in less time complexity.

A. Side-Information

Since increasing the amount of information it needs to extract relevant information for text mining. Thus there are tremendous amount of mining algorithms. These mining algorithms use only the pure data but not the additional information. Many text documents may consist of side-information. It is nothing but the data such as location, ownership or temporal information. This huge amount of information may be used for performing text clustering. This information can either improve the quality of the representation for the mining process, or can add noise to the process. When the information is noisy it can be a difficult approach for performing mining process along with the side-information. These noises can reduce the quality of clustering while if the side-information is informative then it can improve the quality of clustering. Some examples of such side-information are as follows

1. User Access Web Documents

In an application the user-access behavior of web documents are tracked, this user-access behavior may be captured in the form of web logs. For each document, the browsing behavior of different users is considered as the meta-information. Such logs can be used to improve the quality of the mining process in a way which is more meaningful to the user. This is because the logs can often pick up subtle correlations in content, which cannot be picked up by the raw text alone.

2. Text Document Contains Links

Text documents may contain other text documents, which can also be treated as attributes. Such links give more information about the text document which is useful information for text mining purposes. Such attributes provide correlations among documents in a way which may not be easily accessible from raw text content.

3. Metadata

Many web documents consist of metadata with them. These metadata correspond to different kinds of attributes such as the source or other information related to the origin of the document. Data such as location, ownership or even temporal information may be considered as side-information. This huge amount of information may be used for performing text clustering.

Goal of this paper is to show that the advantage of using side-information, thus improve the accuracy of text document clustering in less time complexity.

The above section discusses the introduction of the clustering, classification and some examples of side- information available with the documents. Section II describes the related work. Section III formalizes the problem of text clustering with side-information. Section IV discusses how to extend these clustering techniques to improve the accuracy. Section V contains the conclusion.

2 RELATED WORK

The database community has been studied the problem of text-clustering [6]. Scalable clustering of multidimensional data of different types [5], [6], [7] is the major focus of their work. A general survey of clustering algorithms may be found in [10], [11]. 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 [6], [7], [8], [12]. The scatter-gather technique is the one of the most well known techniques for text-clustering [8], which uses a combination of agglomerative and partitional clustering. Other related methods for text-clustering which use similar methods are discussed in [10], [13]. Co-clustering methods for text data are proposed in [5]. In this context, a method for topic-driven clustering for text data has been proposed in [12]. Text clustering methods in the context of keyword extraction are discussed in [9]. A number of practical tools for text clustering may be found in [5]. The problem of text clustering has also been studied in context of scalability in [2], [3], [4]. However, all of these methods are designed for the case of pure text data, and do not work for cases in which the text-data is combined with other forms of data. Some limited work has been done on clustering text in the context of network-based linkage information [1], [2], [9], [13], though this work is not applicable to the case of general side-information attributes. Gini index is used for feature selection in [2]. Different feature selection methods are discussed in [14], [15], [16]. In this paper, provides a first approach to using other kinds of attributes in conjunction with text clustering. The result will show the advantages of using such an approach over pure text-based clustering. Such an approach is especially useful, when the auxiliary information is highly informative, and provides effective assistance in creating more coherent clusters.

The side-information can sometimes be useful in improving the quality of the clustering process; it can be a risky approach when the side-information is noisy. In such cases, it can actually reduce the quality of the text clustering. Therefore, here use an approach which carefully ascertains the coherence of the clustering characteristics of the side-information with that of the text content. This helps in improving the clustering effects of both kinds of data. The major idea of this approach is to find out a clustering in which the side-information and text attributes provide similar hints about the nature of the underlying clusters, and at the same time the provided conflicting hints are avoided. While the primary goal is to study the clustering problem, note that such an approach can also be extended in principle to other data mining problems in which auxiliary information is available with text. In a wide variety of data domains, such scenarios are very common. Therefore, propose a method in this work in order to extend the approach to the problem classification. The extension of the approach to the classification problem provides superior results because of the incorporation of side-information. The goal is to show that the advantages of using side-information extend beyond a pure clustering task, and can provide competitive advantages for a wider variety of problem scenarios. The problem occurs in the existing work is performance of the mining process is need to be improved. In existing work such side-information of the documents is used which can sometimes be useful in improving the quality of the clustering process, it can be a difficult approach when the side-information is noisy. Number of features is higher in this system. So the time complexity of the system is increased for high dimensional data.

3 TEXT CLUSTERING USING SIDE-INFORMATION

A. Data Set

CORA data set is used for evaluation. The CORA data set consists of 19,396 scientific articles in the computer science domain. In order to compose author-pair graph streams from the scientific publications, each scientific article as a graph object with co-author relationships as edges is considered. The research topics of scientific papers are used as the ground truth to evaluate the clustering quality. In the CORA data set, all research papers are classified into a topic hierarchy, with 73 sub topics on the leaf level. The second level topics are used as the labels to evaluate. There are 10 topics in total, which are *Operating Systems, Information Retrieval, Data Structures Algorithms and Theory, Artificial Intelligence, Encryption and Compression, Databases, Networking, Hardware and Architecture, Programming* and *Human Computer Interaction*. Each paper has an average 3.3 authors. For the side attributes, two types of side-information are obtained to assist clustering: terms and citations. The terms are extracted from the paper titles, and citations include a list of papers that a given article cites. One paper cites 4.3 papers and has 6.1 distinct terms in average.

B. Text Preprocessing

Many documents consist of side-information. The accuracy of text clustering can be improved when it considers the side-information. Mining from a preprocessed text is easy as compare to natural languages documents. So, it is an important task before performing the clustering. As Text documents can be represented as bag of words on which different text mining methods are based. Let Ω be the set of documents & W= {w1, w2, ----wm} be the different words from the document set. In order to reduce the dimensionally of the documents words, special methods such as filtering and stemming are applied. Filtering methods remove those words from the set of all words, which do not provide relevant information; stop word filtering is a standard filtering method. Words like conjunctions, prepositions, articles, etc. are removed that contain no informatics as such stemming methods: are used to produce the root from the plural or the verbs. For e.g. Doing, Done, Did may be represented as Do. Thus every word is represented by its root (or stem) word. Preprocessing text is called tokenization or text normalization.

- Stemming
- Elimination of Stopwords
- Frequency count

1. Identify Stemming Module

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. Stemming is a technique used to find out the root/stem of a word. For example, consider the words user, users, used, and using. The stem of these words is use. Similarly the stem of words engineering, engineered, and engineer is engineer. Since it matches similar words the stemming technique improves effectiveness of text mining and it reduces the indexing (data) size as much as 40-50% by combing words with same roots.

Basic stemming methods are

- remove ending
  - if a word ends with a consonant other than *s*, followed by an *s*, then delete *s*.
  - if a word ends in *es*, drop the *s*.
  - if a word ends in *ing*, delete the *ing* unless the remaining word consists only of one letter or of *th*.
  - If a word ends with *ed*, preceded by a consonant, delete the *ed* unless this leaves only a single letter.
  - …...
- transform words
  - if a word ends with “ies” but not “eies” or “aies” then “ies --> y.”

2. Identify Stop words Module

Many of the most frequently used words in English are worthless in text mining – these words are called *stop words*. A document typically contains about 400 to 500 such words. For example the, of, and, to, etc,. Stop words account 20-30% of total word counts. When removing these words it reduces the indexing (or data) file size. Since 20-30% of total words are stop words in a document it always has a large number of hits and such words are not useful for searching or text mining. Thus the removal of stop words improves efficiency.

3. Words frequent list module

The number of words (or terms) occur in a document is the word (or term) frequency count.

C. Term Frequency-Inverse Document Frequency (TF-IDF)

- Term Frequency: Number of times a term occurs in the document is the term frequency. Suppose a document contains total 60000 words (or "terms") and a word occurs 60 times. Then, Term Frequency, TF = 60/60000 =0.001.
- Inverse Document Frequency: It is a way to score the importance of words in a document based on how frequently they appear across multiple documents. Suppose one bought *Harry-Potter series*, all series. Suppose there are 7 series and a word *"AbraKaDabra"* comes in 2 of the series. Then, Inverse-Document Frequency, IDF =1 + log(7/2). And Finally, Term Frequency-Inverse Document Frequency is the product of Term Frequency and Inverse Document Frequency. i.e., TF-IDF = TF * IDF.

If a word appears frequently in a document, it's important. Give the word a high score. But if a word appears in many documents, it's not a unique identifier. Give the word a low score. Therefore, common words like "the" and "for", which appear in many documents, will be scaled down. Words that appear frequently in a single document will be scaled up.

D. Cosine Similarity

It measures the similarity between sentences or documents in terms of the value within the range of [0,1].

Cosine similarity calculation:

1. Convert strings into vectors (say A and B).
2. Take the union of those vectors to create a shared dimensional space.
3. Find the dot product of vectors A and B.
4. Calculate the magnitude of vector A.
5. Calculate the magnitude of vector B.
6. Multiple the magnitudes of A and B.
7. Divide the dot product of A and B by the product of the magnitudes of A and B.

$$\text{Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \, x \, B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \, x \, \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

4 PROPOSED WORK

In existing system, Gini index is used for feature selection. It is effective to a certain extent but the remaining number of features is still huge. It is important to use feature selection methods to handle the high dimensionality of data for effective text categorization. Feature selection in text mining focuses on identifying relevant information without affecting the accuracy of the classifier. In proposed system, feature selection method is performed in text documents. At the end of the feature selection, only a less number of features are obtained and those features increase the accuracy, with their corresponding rank. The method employs an efficient strategy of ensemble feature correlation with ranking method. The experimental results show that the proposed Effective Feature Selection (EFS) embedded classifier model achieves remarkable dimensionality reduction in the data. In this feature selection method, reduce the features in three stages. In first stage, Gini index based feature selection. Second stage, Correlation based Feature Selection (CFS) based ranking the features after that, in third stage Symmetric Uncertainty (SU) used for re-ranking the features.

TABLE 1
FILTER FEATURE SELECTION METHODS

Feature selection methods	**Advantages**	**Disadvantages**
Gini index	• Select features efficiently. • Measures the features' ability to discriminate between classes. • Widely used in building Classification Trees and determining more important splits.	• Select large number of features.
Pearson's Correlation Coefficient	• Both Supervised and unsupervised. • Works in univariate setting. • Very simple to interpret and implement.	• Works only on numeric attributes. • Can detect linear relationship.
Correlation-based Feature Selection (CFS)	• Supervised, Multivariat • Works with all type of data. • Simplicity of the theory. • Select fewer features with higher accuracy. • Quickly identify irrelevant, redundant features and noise.	• To obtain the optimal feature set, have to perform a search in the feature subspace which may not be required.
Mutual Information	• Supervised • Works with all type of data.	• It is said to be biased towards features with more value.
Symmetric uncertainty (SU)	• Supervised • Works with all type of data. • Eliminate biased nature. • Select fewer features with higher accuracy.	• Cannot apply to large datasets.
Crammer's V	• Supervised • Works with all type of data.	• There is a criticism when this is applied on high dimensional datasets. • Works only in supervised setting.

The correlation between each feature and the class and between two features can be measured and best-first search can be exploited in searching for a feature subset of maximum overall correlation to the class and minimum correlation among selected features. This is determined in the Correlation-based Feature Selection (CFS) method. Correlation based Feature Selection is an algorithm that wraps this evaluation formula with an appropriate correlation measure and a heuristic search strategy. CFS quickly identifies and removes irrelevant, redundant, and noisy features, and determines relevant features as long as their relevance does not strongly depend on other features. This fully automatic algorithm does not require the user to specify any thresholds or the number of features to be selected, although both are simple to incorporate if desired. In spite of feature extraction and selection, a problem is presented namely the classifier may be inclined towards the attributes with more values. Hence this inclined nature has to be eliminated for which employ Symmetrical Uncertainty (SU). It overcomes the problem of bias towards attributes with more values, by dividing information gain by the sum of the entropies of feature subsets Si and Sj. Symmetry is a desired property for a measure of correlations between features. However, information gain is biased in favor of features with more values. Furthermore, the values have to be normalized to ensure they are comparable and have the same influence. Therefore, choose symmetrical uncertainty.

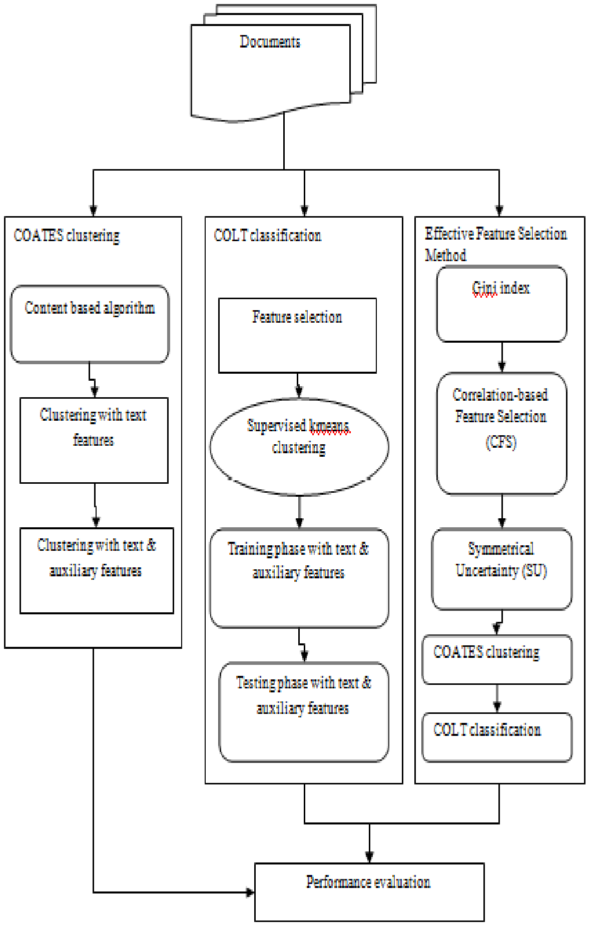
Documents
COATES clustering
Content based algorithm
Clustering with text features
Clustering with text & auxiliary features
COLT classification
Feature selection
Supervised kmeans clustering
Training phase with text & auxiliary features
Testing phase with text & auxiliary features
Effective Feature Selection Method
Gini index
Correlation-based Feature Selection (CFS)
Symmetrical Uncertainty (SU)
COATES clustering
COLT classification
Performance evaluation

Fig 1. Proposed System Architecture

5 IMPLEMENTATION

For experimentation, CORA dataset has been used. The screen shots show the document clustering without sing side-information (based on TF-IDF, cosine similarity), along with side-information and finally based on effective feature selection method.

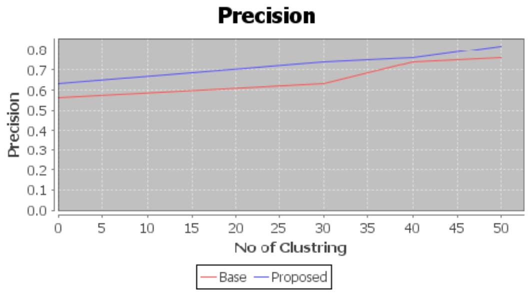

Fig 2. Variation of Precision in existing and proposed system

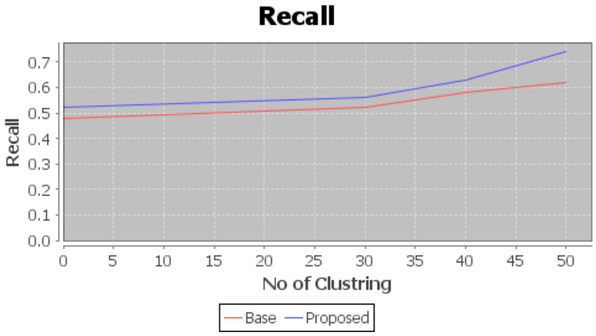

Fig 3. Variation of Recall in existing and proposed system

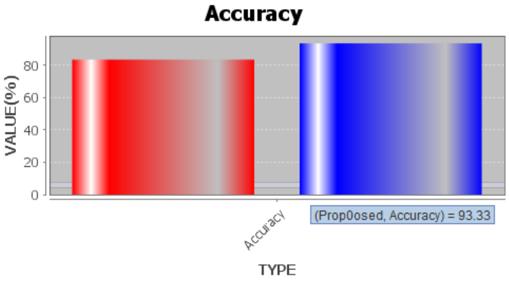

Fig4. Accuracy of text clustering with and without using side-information

6 CONCLUSION

In this paper, methods for mining text documents along with the use of side-information are presented. Side-information or meta-information is present in many forms of databases. It can be used to improve the clustering process. In order to design the advance clustering method, iterative partitioning technique and a probability estimation process are combined. It computes the importance of different kinds of side-information. For designing the clustering and classification algorithms a general approach is used**.** COATES Algorithm proves to be very effective. Effective feature selection method is used to extract the features in text documents. Besides using Gini index, correlation based feature selection and symmetric uncertainty are used. Thus it improves the accuracy of text clustering in less time complexity.

REFERENCES

[1] J. and Kamber, M., Data Mining: Concepts and Techniques, 2nd ed., Elsevier, Morgan Kaufmann, 2006.

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

[3] Charu C. Aggarwal, Yuchen Zhao, and Philip S. Yu, On the Use of Side Information for Mining Text Data, IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 6, June 2014.

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

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

[6] 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.

[7] S. Guha, R. Rastogi, and K. Shim, ROCK: A robust clustering algorithm for categorical attributes, Information Systems., vol. 25, no. 5, pp. 345–366, 2000.

[8] 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, pp. 318–329, 1992.

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

[10] C. C. Aggarwal and P. S. Yu, A framework for clustering massive text and categorical data streams, in Proc. SIAM Conf. Data Mining, pp. 477–481, 2006.

[11] C. C. Aggarwal, S. C. Gates, and P. S. Yu, On using partial supervision for text categorization, IEEE Trans. Knowl. Data Eng., vol. 16, no. 2, pp. 245–255, Feb. 2004.

[12] Y. Zhao and G. Karypis, "Topic-Driven Clustering for Document Datasets," Proc. SIAM Int'l Conf. Data Mining, pp. 358-369, 2005.

[13] Angelova, R., Siersdorfer, S., A neighborhood based approach for clustering of linked document collections, In Proc. of the 15th ACM CIKM, pp. 778–779, 2006.

[14] T Liu, S Liu, Z Chen, WY Ma., An Evaluation on Feature Selection for Text Clustering In ICML, <aaai.org, 2003>.

[15] H. H. Hsu, C. W. Hsieh, Feature Selection via Correlation Coefficient Clustering, Journal of Software, vol. 5, no. 12, pp. 1371-1377, 2010.

[16] H. Liu and L. Yu., Toward integrating feature selection algorithms for classification and clustering, Knowledge and Data Engineering, IEEE Transactions on, 17(4):49-502, April 2005.