Approaches to Ontology Based Algorithms for Clustering Text Documents

V.Sureka
Research scholar
P.S.G.R. Krishnammal College for Women
Coimbatore, India.
vsureka80@yahoo.com

S.C.Punitha
Head of the Department
P.S.G.R Krishnammal College for Women
Coimbatore, India.
saipunith@yahoo.co.in

Abstract

The advancement in digital technology and World Wide Web has increased the usage of digital documents being used for various purposes like e-publishing, digital library. Increase in number of text documents requires efficient techniques that can help during searching and retrieval. Document clustering is one such technique which automatically organizes text documents into meaningful groups. This paper compares the performance of enhanced ontological algorithms based on K-Means and DBScan clustering. Ontology is introduced by using a concept weight which is calculated by considering the correlation coefficient of the word and probability of concept. Various experiments were conducted during performance evaluation and the results showed that the inclusion of ontology increased the efficiency of clustering and the performance of ontology-based DBScan algorithm is better than the ontology-based K-Means algorithm.

Keywords: Concept Weight, DBScan, Document Clustering, K-Means, Semantic Weight, Ontology

1. Introduction

Text or document clustering, a subfield of text data mining, is the process of automatically organizing text documents into meaningful groups in such a manner where all the documents in the same cluster have high similarity and have dissimilarity between clusters (Shawkat Ali, 2008)[1]. Text clustering techniques have wide usage in search engines (to present organized and understandable results to the user), digital libraries (clustering documents in a collection), automated (or semi-automated) creation of document taxonomies and in general, all information retrieval systems involving text. Perhaps the most popular application of document clustering is the Google News2 service, which uses document clustering techniques to group news articles from multiple news sources to provide a combined overview of news around the Web.

Several researches have been proposed to efficiently cluster text documents (Cao *et al.*, 2008 [2] ; Yoo and Hu, 2006) [3][4]. All these techniques can be grouped as flat and hierarchical algorithms. Irrespective of the technique, when applied to text clustering, four issues should to be considered. They are,

- Selection of features
- Dimensionality of Feature Space – Process
- Clustering process and
- Clustering algorithm.

Selection of important features is the process of identifying quality terms (word) that have positive impact on clustering performance by removing redundant or irrelevant data (Sebastiani, 2002)[5]. The features thus selected normally are high dimensional, which imposes a big challenge to the performance of clustering algorithms. Most of the clustering algorithms aim to reduce this high dimensionality while maintaining the document's semantic structure (Shahnaz *et al.*, 2006)[6]. Clustering process is the process of calculating a similarity measure that denotes the content similarity between two term vectors of two documents. The result is often used by the partitioning algorithm and is critical for obtaining quality clusters. Several similarity measures, like cosine similarity, are used. With the result of the similarity measure, the next step is the actual clustering process. A variety of clustering algorithms are available which includes k-means, EM (Expectation Maximization) algorithm, Self Organizing Maps (SOM), fuzzy clustering and choosing the one that best suit an application is a challenging task.

Another challenge faced by the existing document clustering algorithms is that they do not consider the semantics of features (terms) selected for clustering. To solve this problem, ontological-based clustering techniques are being popularly used in the past few decades. An ontology-based approach allows the analyst to represent the complex structure of objects, to implement the knowledge about hierarchical structure of categories as well as to show and use the information about relationships between categories and individual objects.

Inspite of different algorithms being proposed for efficient document clustering, in the domain of documents, clustering research is still at its peak. The reason behind this is that the text documents are constantly changing both in volume and dimension and new innovative techniques are need to search and extract useful knowledge from them. Because of the lack of satisfactory techniques for extracting information, researchers seek alternative schemes that either enhance the existing methods or propose new amalgamation of mining techniques. This research work concentrates on the first alternative and enhances an ontology-based algorithm for text document clustering.

Tar and Nyunt (2011) [7] proposed a technique that used ontology during term weighting and performed document clustering using k-means in accordance with the principles of ontology so that the important of words of a cluster can be identified by the weight values. It is well-known fact that the performance of the k-means algorithm degrades with the improper selection of 'k' values and initial parameters. In order to solve this problem, an enhanced DBScan algorithm is used in the proposed algorithm to improve the clustering process. The main objective of this paper is thus to develop a document clustering algorithm that uses ontology for term weighting and performs DBScan clustering to group similar documents and compare the result with ontology-based K-Means algorithm. The rest of the paper is organized as follows. A brief literature study is proposed in Section 2. The proposed methodology is presented in Section 3 and the experimental results are discussed in Section 4. The study is concluded with future research directions in Section 5.

2. Literature Study

Various research works have concentrated on comparing the performance of different clustering algorithms to analyze their merits and demerits and in order to suggest users the best technique for a particular situation. This section lists some of them. Steinbach *et al.* (2000) [8] compared traditional K-means algorithm along with a variant K-means, "bisecting" K-means with Hierarchical clustering algorithm. A comprehensive comparison study of various document clustering approaches such as three hierarchical methods (single-link, complete-link, and complete link), Bisecting K-means, K-means, and Suffix Tree Clustering in terms of the efficiency, the effectiveness, and the scalability was performed by Yoo and Hu (2006)[3][4]. Jain *et al.* (1999)[9] performed cluster analysis and used methodologies pertaining mainly into partitional and hierarchical clustering methods. Chakrabarti (2003)[10] also discusses various types of clustering methods and categorizes them into partitioning, geometric embedding and probabilistic approaches.

Saad et al. (2006)[11] evaluated various criterion functions in the context of the partitional approach, namely the repeated bisection clustering algorithm and compared the quality of the clusters produced by agglomerative and partitional algorithms from the perspective of different criterion functions to establish the right clustering algorithm to produce high quality clustering of real-world medical documents. A study that evaluated and compared concept-based and N-Grams Based Text Clustering Using SOM was reported by Amine et al. (2008)[12]. Recently, Chen et al. (2010)[13] compared the performance of SOM and K-means algorithms for clustering text documents.

From the literature study conducted, it is obvious that while studies that compare the performance and working of different clustering algorithms are available, studies comparing ontology-based clustering are sparse. This study, is one such work, where the existing method is enhanced and a comparison between existing and proposed methods are reported.

3. Methodology

The proposed document clustering using ontology combines concept weighting or semantic weighting and two clustering algorithms, namely, KMeans and DBScan algorithm. The system performs clustering in three steps, namely, document preprocessing, calculating concept weight based on the ontology and clustering documents with the concept weight.

Preprocessing consists of procedures that convert the textual data in a document to a structure ready for data mining. The main objective of preprocessing is to obtain the key features or key terms from online news text documents and to enhance the relevancy between word and document and the relevancy between word and category. In general, the preprocessing and document representation stage consists of the following steps.

a) Feature Generation
b) Index document using concept weighting
c) Dimensionality reduction using Document Frequency Thresholding method.

Initially, a document is parsed to remove unwanted terms and symbols, like HTML tags, non-alpha characters. Case folding (converting all characters to the same case, either lower or uppercase) is performed. Using Bag of Words (BoW) is used to extract terms, which are refined using stop word removal and stemming. This paper uses the SMART stop word list proposed by Salton (1971) [14] and Porter's Stemmer Algorithm (Porter, 1980) [15]for stemming.

The next step is document indexing which is sued to increase the efficiency by extracting from the resulting document a selected set of terms to be used for indexing the document. Document indexing consists of choosing the appropriate set of keywords based on the whole corpus of documents, and assigning weights to those keywords for each particular document, thus transforming each document into a vector of keyword weights.

In this research, instead of the traditional TF/IDF (Term Frequency/Inverse Document Frequency) uses a concept weighting scheme based on ontology. The reason behind this is that TF/IDF only considers the frequency of words appearance, while ignoring other factors which may impact the word weighs. In this regard, an ontology is defined as a set of concepts of interest domain organized as a hierarchical structure. The following assumptions are made during the calculation of weight.

1. More times the words appear in the document, more possibly it is the characteristic words
2. The length of the words will also affect the importance of words. Apparently, one concept in the ontology is related to other concept in that domain ontology. That also means that the association between two concepts can be determined using the length of these two concept's connecting path (topological distance) in the concept lattice.
3. If the probabilities of one word is high, then the word will get additional weight
4. One word may be the characteristic word even if it doesn't appear in the document.

A tighter combination of above depicted four assumptions leads to the concept weighting structure with the ontological aspects. The proposed clustering algorithm takes into account the frequency, length, specific area and score of the concept when calculating the weighs, using the function with weight values as follows (Equ 1)

W = Len× Frequency× Correlation Coefficient + Probability of concept (1)

where W is the weight of keywords, len is the length of keywords, Frequency is times which the words appear and if the concept is in the ontology , then correlation coefficient =1 , else correlation coefficient=0. Probability is based on the probability of the concept in the document. The probability is estimated using Equ (2)

$$P(\text{concept}) = \frac{\text{Number of Occurences of the Concept}}{\text{Number of Occurences of All Concepts in Document}} \qquad (2)$$

Finally, the system ranks the weights and selects the keywords that have with bigger weight for pre clustering process. Ontology can be represented by standard ontology language. The motivation behind this step is that the OWL is one of the most used standards in describing the knowledge base and is already used in many Semantic Web applications. Additional motivation for using OWL is the availability of the knowledge base development tools such as Protégé – OWL editor that supports OWL standard. The result of this step is then used to cluster the documents using KMeans (Figure 1) and DBScan (Figure 2) algorithms.

4.Experimental Results

The performance of the two algorithms, ontology based k-means algorithm and ontology-based DBScan algorithms, were compared using two text corpora, namely, ModApte (Apte *et al.*, 1994) [16] a popular variant of Reuters 21578 and 20 Newsgroup (Baeza-Yates and Ribeiro-Neto, 1999)[17]. The performance of the two algorithms was analyzed using the precision, recall, F measure and Accuracy. The F-measure is calculated from two measures, precision and recall, which are derived from four values, namely, true positive (TP), true negative (TN), false positive (FP) and false negative (FN) during analysis of performance (Figure 3).

The equation used to calculate precision (p) and recall (r) are given in Equ 4 and 5.

$$P(i,j) = \frac{N_{ij}}{N_j} \qquad (4)$$

$$R(i,j) = \frac{N_{ij}}{N_i} \qquad (5)$$

where N_{ij} is the number of objects of class 'i' in cluster 'j'. N_j is the number of objects in cluster 'j', N_i is the number of objects of class 'i'.

Steps

```
1. Begin with a decision on the value of k = number of clusters.
2. Put any initial partition that classifies the data into k clusters.
    • Take the first k data as single-element clusters
    • Assign each of the remaining (N-k) data to the cluster with the nearest centroid.
      After each assignment, recompute the centroid of the gaining cluster.
3. Take each data in sequence and compute its distance from the centroid of each of the clusters.
   If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster
   and update the centroid of the cluster gaining the new sample and the cluster losing the sample.The
   distance metric used is Euclidean distance (Equ3).
```

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \qquad (3)$$

```
4. Repeat step 3 until convergence is achieved, that is, until a pass through the training sample
   causes no new assignments.
```

Figure 1: Traditional K Means Algorithm.

```
DBSCAN(D, eps, MinPts)
  C = 0
  for each unvisited point P in dataset D
     mark P as visited
     N = getNeighbors (P, eps)
     if sizeof(N) < MinPts
        mark P as NOISE
     else
        C = next cluster
        expandCluster(P, N, C, eps, MinPts)

expandCluster(P, N, C, eps, MinPts)
  add P to cluster C
  for each point P' in N
     if P' is not visited
        mark P' as visited
        N' = getNeighbors(P', eps)
        if sizeof(N') >= MinPts
           N = N joined with N'
     if P' is not yet member of any cluster
        add P' to cluster C
```

Figure 2 : DBScan Algorithm

	Same category	Different categories
Same cluster	TP	FP
Different cluster	FN	TN

Figure 3 : Confusion Matrix

The F-measure is calculated using Equ 6.

$$F(i,j) = \frac{2P(i,j)R(i,j)}{P(i,j)+R(i,j)} \qquad (6)$$

It is always desired to obtain a large F-measure, which indicates better clustering performance. In general, a larger F-measure value indicates better clustering result (Steinbach *et al.*, 2000). The accuracy is calculated using Equ (7).

Accuracy=(TP+TN) / (TP+TN+FP+FN) (7)

In the results, OKM and ODB refer to the ontological-based K-Means and ontological-based DBScan algorithms. The precision and recall values obtained for the two algorithms are shown in Table 1.

Table 1: Precision and Recall

Data set	Precision		Recall	
	OKM	**ODB**	**OKM**	**ODB**
20-Newsgroup	0.613	0.785	0.726	0.798
Reuters-21578	0.515	0.591	0.832	0.864

From the values obtained during experimentation, it is obvious that though both the ontological-based algorithm produce good precision and recall, the DBScan based algorithm shows significant improvement.

Table 2 shows the clustering F-Measure and accuracy of the ontological K-Means and DBScan clustering algorithms.

. Table 2 : Accuracy

Dataset	F Measure		Accuracy	
	OKM	**ODB**	**OKM**	**ODB**
20-Newsgroup	0.665	0.791	92.43	93.96
Reuters-21578	0.636	0.702	90.16	91.66

From the tabulated results again the ontological DBScan proves to provide better clustering both in terms of F Measure and accuracy than ontological K-means with both the text datasets.

The results from the various experiments show that the clustering algorithm that uses semantics of the documents for term weighting and DBScan for clustering produces significant difference in results when compared with the ontological K-means algorithm and has improved the process of clustering.

5. Conclusion

This paper analyzed and compared two methods that used semantic weight and two clustering algorithms, namely, K Means and DBScan. In these clustering methods, after stop word removal and stemming a concept weight is calculated by taking into consideration the frequency of a word in a document, length of words, correlation coefficient of the word and probability of concept. The probability of concept is determined by using OWL ontology standard. The terms selected using semantic weights are then clustered using K-Means and DBScan algorithms. Experimental results that compared the performance of K-means and DBScan based on ontology showed that DBScan clustering algorithm using ontological weighting scheme produce better result than K-Means algorithm. In future, methods that combine KMeans and DBScan are to be probed.

References

[1] Shawkat Ali, A.B.W. (2008) K-means Clustering Adopting RBF-Kernel, Data Mining and Knowledge Discovery Technologies, David Taniar (Ed.), Pp. 118-142.

[2] Cao, T.H., Do, H.T., Hong, D.T. and Quan, T.T. (2008) Fuzzy named entity-based document clustering, Proceedings of IEEE International Conference on Fuzzy Systems, Hong Kong, Pp. 2028-2034.

[3] Yoo, I. and Hu, X. (2006) A comprehensive comparison study of document clustering for a biomedical digital library MEDLINE, Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries JCDL '06, ACM New York, Pp.220-229.

[4] Yoo, I. and Hu, X. (2006) A comprehensive comparison study of document clustering for a biomedical digital library MEDLINE, Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries JCDL '06, ACM New York, Pp.220-229.

[5] Sebastiani, F. (2002) Machine Learning in Automated Text Categorization, ACM Computing Surveys, Vol. 34, No. 1, Pp. 55-59.

[6] Shahnaz, F., Berry , M.W., Pauca, V.P. and Plemmons, R.J. (2006) Document clustering using nonnegative matrix factorization, Information Processing and Management, Vol. 42, Pp. 373–386.

[7] Tar, H.H. and Nyunt, T.T.S. (2011) Ontology-Based Concept Weighting for Text Documents, International Conference on Information Communication and Management, Vol.16, Pp. 165-169.

[8] Steinbach, M., Karypis, G. and Kumar, V. (2000) A comparison of document clustering techniques (M. Grobelnik, D. Mladenic and N. Milic-Frayling, Eds.) KDD workshop on text mining, 34(X), 35, IEEE Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4721382

[9] Jain, A.K., Murty, M.N. and Flynn, P.J. (1999) Data clustering: A review. ACM Comput. Surveys, Vol. 31, Pp. 264-323.

[10] Chakrabarti, S., (2003) Mining the Web: Discovering Knowledge from Hypertext Data, Morgan Kaufmann Publishers, California.

[11] Saad, F.H., de la Iglesia, B. and Bell, G.D. (2006) A Comparison of Two Document Clustering Approaches for Clustering Medical Documents, Conference on Data Mining, DMIN'06, Pp,425-431.

[12] Amine, A., Elberrichi, Z., Simonet, M. Malki, M. (2008) Evaluation and Comparison of Concept Based and N-Grams Based Text Clustering Using SOM, Pp.1-9.

[13] Chen, Y., Qin, B., Liu, T., Liu, Y. and Li, S. (2010) The Comparison of SOM and K-means for Text Clustering, Computer and Information Science, Vol. 3, No.2, Pp.268-274.

[14] Salton, G. (1971) The SMART Retrieval System-Experiment in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, New Jersey.

[15] Porter, M. (1980) An algorithm for suffix stripping, Program, Vol. 14, No. 3, Pp. 130–137.

[16] Apte, C., Damerau, F. and Weiss S. (1994) Automated learning of decision rules for text categorization. ACM Transactions on Information Systems, Vol. 12, No. 3, Pp. 233–251.

[17] Baeza-Yates, R. and Ribeiro-Neto, B. (1999) Modern Information Retrieval. Addison Wesley Longman.