www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 7 July 2015, Page No. 13086-13090

Text Mining using Ontology based Similarity Measure

Atiya Kazi[1], D.T.Kurian[2]

[1]RMDSSOE, Savitribai Phule Pune University
Pune, India
atiyakazi@gmail.com

[2]RMDSSOE, Savitribai Phule Pune University
Pune, India
dtkurian@sinhgad.edu

Abstract: ***In many text mining applications, the side-information contained within the text document will contribute to enhance the overall clustering process. The proposed algorithm performs clustering of data along with the side information, by combining classical partitioning algorithms with probabilistic models to boost the efficacy of the clustering approach. The clusters generated will be used as a training model to solve the classification problem. The proposed work will also make use of a similarity based ontology algorithm, by incorporating two shared word spaces, to perk up the clustering approach. This will boost the amount of knowledge gained from text documents by including ontology with side-information.***

Keywords: Clustering, Data Mining, Ontology, Side-Information

1. Introduction

There are several attributes in a text document that carry side-information for clustering purposes. But, an optimized way is necessary enable the mining process, so that the side information is correctly utilized. The probabilistic approach of mining can be also extended to the classification problem. Along with it an existing ontological schema can be added to the clustering process at compile time and its effects on the generated output could be analyzed. The current work statement can be put down as, Developing a novel Clustering approach for mining raw text data along with it's side information, and comparing it with Ontology based clustering that provides semantically enhanced clusters. Data Mining is the process of scrutinizing data from different perspectives and summarizing it to gain valuable knowledge. It comprises of clustering and classification on text based data, numeric data and web based data. In many application domains, a remarkable amount of side information is usually available along with the documents which is not considered during pure text based clustering[8]. Clustering text collections has been scrutinized under Data mining in [13]. Some efficient streaming techniques use clustering algorithms, that are adaptive to data streams, by introducing a forgetting factor that applies exponential decay to historical data [9]. Normally, text documents typically contain a large amount of meta information which may be helpful to enhance the clustering process. While such side-information can improve the quality of the clustering process, it is essential to make sure that the side-information is not noisy in nature. In some cases, it can hamper the eminence of the mining process. Therefore, one needs an approach which, carefully perceives the consistency of the clustering distinctiveness of the side information, along with the text content. The core approach is to determine a clustering process where text attributes along with the additional side-information provide comparable hints regarding the temperament of the basic clusters, as well as, they ignore conflicting aspects. The goal is to show that the reward of using side-information broadens the data mining process beyond a pure clustering task. Recently, Ontologies have become an integral part of fabricating knowledge, so as to create knowledge-rich systems. An ontology is formally defined as an explicit formal hypothesis of some domain of interest which helps in the interpretation of concepts and their associations for that particular domain [2]. To create an ontology, one needs a data mining expert who understands all the domain concepts, domain hierarchies and the relationships between them for a specialized domain. A similar approach is proposed in [5], which uses domain based, schema based, constraint based and user preference based ontologies for enhancing the test clustering process. The current work focuses on techniques, which incorporate a user-preference ontology during the data mining process.

2. Related Work

The major work in the field of data mining looks upon scalable clustering of spatial data, data with boolean attributes, identifying clusters with non spherical shapes and clustering for large databases[7]. Several general clustering algorithms are discussed in [3]. An efficient clustering algorithm for large databases, known as CURE, has been covered in [14]. The scatter-gather technique, which uses clustering as its primitive operation by including liner time clustering is explained in [16]. Two techniques which develop the cost of distance calculations, and speed up clustering automatically affecting the quality of the resulting clusters are studied in [10]. An Expectation Maximization (EM) method, which has been around ages for, text clustering has been studied in [12]. It selects relevant words from the document, which can be a part of the clustering process in future. An iterative EM method helps in refining the clusters thus generated. In topic-modeling, and text-categorization, a method has been proposed in [11] which makes use, of a mathematical model for defining each category. Keyword extraction methods for text clustering are discussed in [10]. The data stream clustering problem for text and categorical data domains is discussed in [8]. Speeding up the clustering process can be achieved by, speeding up the distance calculations for document clustering routines as

discussed in [15]. They also improve the quality of the resulting clusters. However, none of the above mentioned works with the combination of text-data with other auxiliary attributes. The previous work dealing with network-based linkage information is depicted in [6], [7], but it is not applicable to the general side information attributes. The current approach uses additional attributes from side information in conjunction with text clustering. This is especially useful, when the Side-information can regulate the creation of more consistent clusters. There are three forms of extending the process of knowledge discovery, with respect to their related ontologies, which are categorized as follows [4],

- Using on hand ontologies for knowledge discovery, during data mining.
- Construction of ontologies through knowledge discovery from mined results.
- Constructing and extending ontologies through knowledge discovery via existing ontologies.

The combination of the first two plays a major role in the methodology of the current paper.

3. SYSTEM ARCHITECTURE

There are three major modules in the system as depicted using figure 3.1, the first is the Preprocessing module, the second one is Clustering module and the last is the Classification module. They are described as below.

3.1 Preprocessing Module

Documents from the datasets are stored within the corpus. In the preprocessing module, extracted documents from the repository are preprocessed. Preprocessing technique includes tokenizing the word, removing stop words, stemming the word and other preprocessing tasks such as calculating the Term Frequency for each word.

3.2 Clustering Module

The role of this module is the creation of clusters which are according to the content of the document. The system uses either COATES algorithm or an Ontology based method to generate the clusters. In the ontology based module, document similarity is usually measured by a pair-wise similarity function. A simple similarity measure, like cosine function, is often used to reflect the similarity between two documents.

3.3 Classification Module

The classification engine is powered by an ontology of similarity indices that categorizes the input document with respect to the clusters generated using DISCO ontology. This ontology can be extended dynamically to allow classification without recompiling the system.

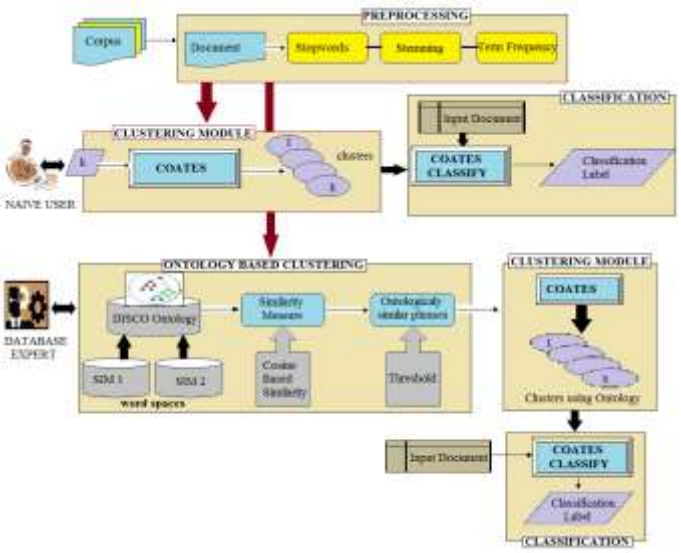

Figure 1: System Architecture for Ontology based Clustering

3.4 DISCO API

DISCO stands for extracting distributional related words using co-occurrences. It is a Java application which helps in regaining the semantic parallel between capricious words and phrases. The similarities are based on the numerical analysis of very large text collections. The DISCO Java API provides methods for extracting the semantically most similar words for an input word, e.g. shy = (timid, quiet ,soft-spoken, gentle). It also works in the assessment of the semantic similarity between two input key words or phrases. The fundamental principles on which the method for knowledge discovery is based on says that the knowledge discovery process is dominated by pre-existing data and the ontologies relevant to the considered domain. Both data and ontologies evolve over a period of time by interacting with each other. The ontologies are enriched with knowledge from the patterns extracted with the help of the data mining tools, while the data is enriched through new inferences which are derived from the ontologies.

Table 1: DISCO Word Space Information

WordSpace Name	Corpus Size	Packet Size	Word Space Type
enwiki-20130403-sim- lemma mwl-lc	1.9 Billion Token	2.3 GB	SIM
enwiki-20130403-word2vec-lm-mwl-lc-sim	1.9 Billion Token	1.4 GB	SIM

Data mining techniques are used to produce suitable patterns that can be filtered out and selected on the basis of their integration with the ontologies. Ontologies are used to select the input of the data mining techniques, based on their common

relevance. New ontological models help in abstracting and validating the existing ones on their consistency. They help in consolidating the available data leading to multiple versions of ontologies and data. They can branch over multiple iterations. The proposed data mining system framework helps in supporting the system's intelligence by incorporating ontologies in the data mining framework. It includes the characteristics of a data warehouse schema, along with the user preference based ontologies.

4. Algorithm Working

The algorithm is referred to as COATES, which corresponds to a COntent and Auxiliary attribute based TExt cluStering algorithm[1]. The input to the algorithm is the number of clusters k. As in the case of all text-clustering algorithms, it is assumed that stop-words have been removed, and stemming has been performed in order to improve the discriminatory power of the attributes.

Figure 2: COATES Clustering

In each content-based phase, a document will be clustered according to its closest seed centroid based on a cosine similarity function. Figure 2 shows the steps involved during the running of COATES algorithm, while the Clustering process steps are depicted in figure 3. Each auxiliary phase, generates a probabilistic model, which combines the attribute probabilities with the cluster-membership probabilities, based on the clusters which have already been created in the most recent text-based phase. This determines the coherence of the text clustering by including side-information.

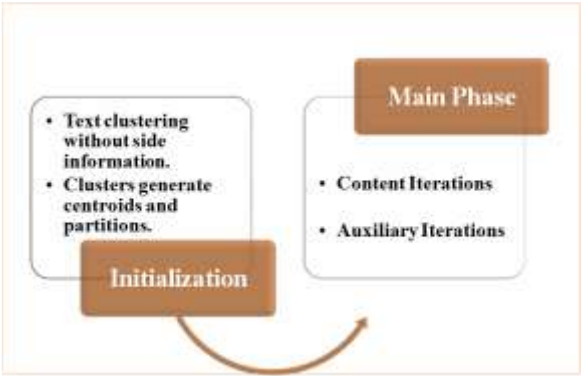

Figure 3: Clustering Phases

4.1 Ontology based Similarity distance Measurement

The algorithm which will use ontology based distance measurement before the clustering process begins is described in the figure 4. The process will generate clusters using the Sem_Dis(C1,C2) for two concepts C1 and C2, with threshold above 0.4. This threshold is decided by calculating the average value of 300 random words from the text documents and their synonyms from the word spaces.

Input: two concepts C_1, C_2
Output: semantic distance: Sem_Dis(C_1,C_2)
For two ontological concepts: C_1, C_2
If C_1, C_2 are the same concept
Sem_Dis(C_1, C_2)=0
Else if there exists the direct path relation between C_1 and C_2
Sem_Dis(C_1, C_2) = w[sub(C_1, C_2)]
Else if there exists the indirect path relations between C_1 and C_2

$$Sem\_Dis(C_1,C_2) = \sum_{C \in SPath(C_1,C_2)} w_c[sub(C_1,C_2)]$$

Where, $SPath$ denotes the shortest path between C_1 and C_2
Else
Sem_Dis(C_1, C_2) = min{Sem_Dis(C_1,C_0)} + min{Sem_Dis(C_2,C_0)}

Figure 4: Ontology Semantic distance solving algorithm

5. Result Analysis

This section reports experimental results when applying the basic ontology algorithm to cluster documents. During experimentation, 9 datasets was used. To ascertain the performance of the models, several experiments were conducted. All the experiments were conducted using an Intel Core 2 duo machine with 2GB RAM. Six performance metrics, namely, Accuracy of a cluster, Sensitivity, Specificity, Precision, F-measure and CPU execution time were used. The results were compared with the existing COATES CLASSIFY algorithm and K-means clustering algorithm. The overall results obtained for the three algorithms for different number of clusters are depicted using tables and graphs. As observed from the results, the Precision and Accuracy are highest for the proposed work as compared to existing kmeans and COAT CLASSIFY method. These values depend upon the True Positive and False Negative values gained after classification.

5.1 Precision and Recall Value

To evaluate the accuracy of our clustering algorithm, one can use Recall and Precision performance metrics. The value of precision and recall can be calculated as:

$$Precision = x/x+y \quad (1)$$

$$Recall = x/x+z \quad (2)$$

where x is the number of total true positives, i.e. the total number of items clustered together in predefined class and that are indeed found together by the clustering algorithm. y is the total number of false positives, i.e. the number of items not supposed to be found together but are clustered together and z is the number of total false negatives, i.e. the number of items which are expected to be found together but not clustered together by the clustering algorithm. The result of the experiment based on these values for the 9 datasets is shown in

the graphs figure 2 and figure 3 where it is matched with the highest similarity.

TABLE 2: Precision Value Comparison

Dataset Name	Precision-COATES	Precision-ONTOLOGY
boptradehoglivestock-crude.txt	0.286052009	0.931372549
earn-acq-money-fx.txt	0.291249165	0.712195122
earnacq-money-fxinterest-money-fxsaudriyal.txt	0.27443609	0.743727599
goldplatinum-wpi-nat-gas.txt	0.250626566	0.817982456
grainrice-grainwheat-graincornbarley.txt	0.286089239	0.790740741
grainwheatrice-ship-crudegas.txt	0.225255973	0.871212121
iron-steel-pet-chem-graincornoilseed soybean.txt	0.220883534	0.718518519
sugar-coffee-cocoa-docs.txt	0.205426357	0.805084746

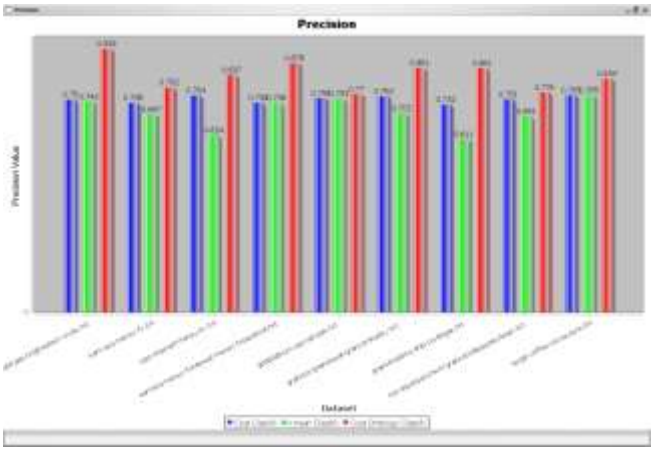

Figure 5: PRECISION COMPARISON GRAPH

TABLE 3: Recall Value Comparison

Dataset Name	Recall-COATES	Recall-ONTOLOGY
boptradehoglivestock-crude.txt	0.286052009	0.931372549
earn-acq-money-fx.txt	0.291249165	0.712195122
earnacq-money-fxinterest-money-fxsaudriyal.txt	0.27443609	0.743727599
goldplatinum-wpi-nat-gas.txt	0.250626566	0.896174863
grainrice-grainwheat-graincornbarley.txt	0.286089239	0.790740741
grainwheatrice-ship-crudegas.txt	0.225255973	0.871212121
iron-steel-pet-chem-graincornoilseed soybean.txt	0.220883534	0.718518519
sugar-coffee-cocoa-docs.txt	0.205426357	0.805084746

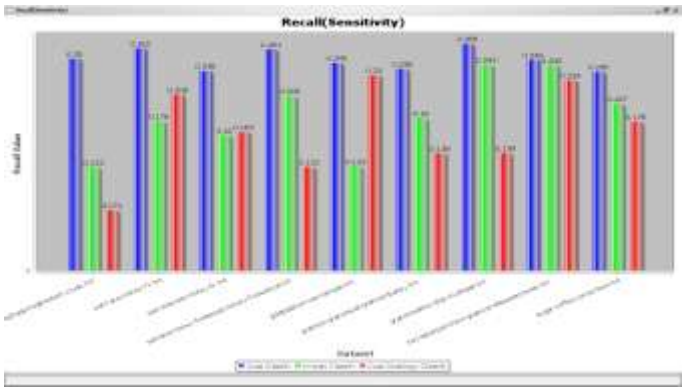

Figure 6: RECALL COMPARISON GRAPH

5.2 Time Difference

The time requirements show that the proposed algorithm takes more time than existing k means as well as COATES. This is because of the gigantic size of ontology repository, due to which the matching process may take more time. To avoid this, one needs a hierarchical structure with semantic interpretation of data.

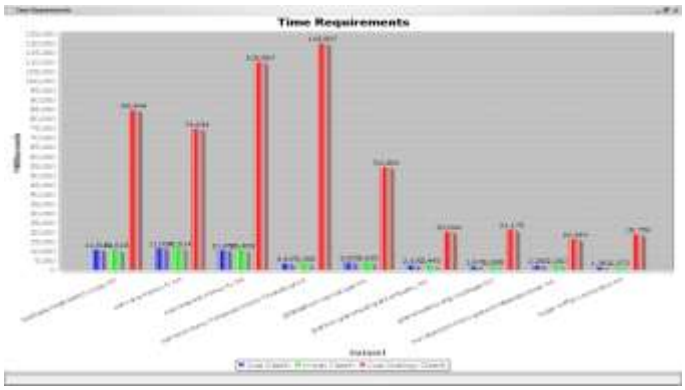

Figure 7: TIME DIFFERENCE COMPARISION GRAPH

TABLE 4: Time Difference Comparison

Dataset Name	TimeRequirement-COATES(Ms)	TimeRequirement-ONTOLOGY(Ms)
boptradehoglivestock-crude.txt	24482	32721

earn-acq-money-fx.txt	19144	33769
earnacq-money-fxinterest-money-fxsaudriyal.txt	23322	34983
goldplatinum-wpi-nat-gas.txt	10935	18331
grainrice-grainwheat-graincornbarley.txt	21533	17771
grainwheatrice-ship-crudegas.txt	8502	35103
iron-steel-pet-chem-graincornoilseed soybean.txt	11944	20626
sugar-coffee-cocoa-docs.txt	7009	15694

6. CONCLUSION

The primary goal was to study the clustering problem and where auxiliary information is available with text and compare it with ontology based clustering. There is also an extension to include problem classification, which provides superior results because of the incorporation of side information and ontology. The generated results have proved how the use of ontology elevates the quality of text clustering and classification, while maintaining a high level of efficiency. It was also observed that applying the ontologies before the phase of clustering minimally partitions the documents into coherent, clustered branches. The simple process of clustering and indexing documents by their ontological relationships puts ordered implication to the meaning of documents. While classification hierarchies only suggest, "what a document is about," ontological knowledge assigns richer significance to documents. Clustering algorithms that rely exclusively on probabilistic techniques may not help in uncovering the more complex semantic significance, endorsed to text document collections, by more affluent ontologies. For future work one can propose the idea of helping the naive user to acquire knowledge from the domain expert. The user will use the extracted knowledge as a guide in acquiring knowledge from the domain expert. The domain expert will corroborate the extracted knowledge, and retain information about the missed knowledge. One can explore the tactic for building ontology from amorphous data such web pages and documents. One can make use of a representation based, control based or domain specific ontology to tune the mining engine with the help of a Database expert.

References

[1] C. C. Aggarwal et al, "On the use of side-information for mining text data", IEEE Trans. Knowl. Data Eng, vol 26, pp. 1415-1429, June 2014.

[2] Henrihs Gorskis, Yuri Chizhov, "Ontology Building Using Data Mining Techniques", Information technology and management science, vol 15, pp 183-188, 2013.

[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] Mathieu d'Aquina, Gabriel Kronbergerb, and Mari Carmen Suárez-Figueroa, "Combining Data Mining and Ontology Engineering to enrich Ontologies and Linked Data", Proc. first International workshop on knowledge discovery and Data Mining , pp 19-24, 2012.

[5] Chin-Ang Wu et al., "Toward Intelligent Data Warehouse Mining: An Ontology-Integrated Approach for Multi-Dimensional Association Mining", Information Technology and Management Science, Expert Systems with applications, volume 38, Issue 9, pp 11011-11023, sept-2011.

[6] J. Chang and D. Blei, "Relational topic models for document networks", in Proc. AISTASIS, Clearwater, FL, USA, 2009, pp. 81-88.

[7] R. Angelova and S. Siersdorfer, "A neighborhood-based approach for clustering of linked document collections", in Proc. CIKM Conf., New York, NY, USA, 2006, pp. 778–779.

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

[9] S. Zhong, "Efficient streaming text clustering", Neural Netw., vol. 18, no. 5–6, pp. 790–798, 2005.

[10] Y. Zhao and G. Karypis, "Topic-driven clustering for document datasets", in Proc. SIAM Conf. Data Mining, 2005, pp. 358-369.

[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] 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.

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

[14] 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.

[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.