IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 31-37
www.iosrjournals.org

Enhanced Automatically Mining Facets for Queries and Clustering With Side Information Model

S. Saranya[1], M.C.A., M. Baskar[2], M.Sc.,M.Phil.,
Research Scholar, Computer Science, Vivekanandha college for women, Tiruchengode, India[1].
Assistant Professor, Computer Science, Vivekanandha college for women, Tiruchengode, India[2].

Abstract: *In this paper describe a specific type of summaries that Query facet the main topic of given text. Existing summarization algorithms are classified into different categories in terms of their summary construction methods (abstractive or extractive), the number of sources for the summary (single document or multiple documents), types of information in the summary (indicative or informative), and the relationship between review and query (generic or query-based. In this project, adding these lists may develop both accuracy and recall of query facets. Part-of-speech information can be used to check the homogeneity of lists and improve the quality of query facets. In this paper related topics to finding query facets. Good descriptions of query facets may be helpful for users to better understand the facets. Automatically generate meaningful descriptions is an interesting research topic. The problem of text clustering arises between many application domain like web, social networks, and other digital collections. The rapidly increasing amount of text data in the context of these large online collections has lead to an interest in create scalable and valuable mining algorithms.*

Keywords: *QD Mininer, Facet Analysis, COADES Algorithm, Side information, Clustering.*

I. Introduction

Data mining is the process of extracting patterns from database. Data mining is seen as increasingly important tool by modern business to transform data into an informational advantage. It is used in a profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery. These techniques can however, be used in the creation of new hypothesis to test against the larger data populations

Data mining derives its name by finding the similarities between searching for valuable information in a large database.. Document clustering is being studied from many decades but still it is far from a trivial and solved problem.

- Identifying sentences that provide the information regarding success factors and their relationships by utilizing data mining technique
- The optimal workflow derived from the results by similar classification task
- Selecting appropriate features of the documents to assist must be used for clustering.
- Selecting an appropriate similarity measure among documents.
- Implementing the clustering algorithm that makes it sufficient in terms of required memory and CPU resources.
- Finding ways of assessing the quality of the clustering technique.

II. Literature Survey

Weize Kong and James Allan [1] describe a faceted search helps users by offering drill-down options as a complement to the keyword input box, and it has been used successfully for many vertical applications, including ecommerce and digital libraries. However, this idea is not well explored for general web search, even though it holds great potential for assisting multi-faceted queries and exploratory search. In this paper, explore this potential by extending faceted search into the open-domain web setting, which is call Faceted Web Search. To tackle the heterogeneous nature of the web, propose to use query-dependent automatic facet generation, which generates facets for a query instead of the entire corpus. To incorporate user feedback on these query facets into document ranking, we investigate both Boolean filtering and soft ranking models. The authors evaluated Faceted Web Search systems by their utility in assisting users to clarify search intent and subtopic information. The authors described how to build reusable test collections for such tasks, and propose an evaluation method that considers both gain and cost for users. Faceted search enables users to navigate a multi-faceted information space by combining text search with drill-down options in each facet. For example, when searching \computer monitor" in an e-commerce site, users can select brands and monitor types from the the provided facets: fSamsung, Dell, Acer, ...g and f LET-Lit, LCD, OLEDg

Krisztian Balog, Edgar Meij and Maarten de Rijke [2] describe the task of entity search and examine to which extent state-of-art information retrieval (IR) and semantic web (SW) technologies are capable of answering information needs that focus on entities. We also explore the potential of combining IR with SW technologies to improve the end-to-end performance on a specific entity search task. We arrive at and motivate a proposal to combine text-based entity models with semantic information from the Linked Open Data cloud. The problem of entity search has been and is being looked at by both the Information Retrieval (IR) and Semantic Web (SW) communities and is, in fact, ranked high on the research agendas of the two communities. The entity search task comes in several flavors. One is known as entity ranking (given a query and target category, return a ranked list of relevant entities), another is list completion (given a query and example entities, return similar entities), and a third is related entity finding (given a source entity, a relation and a target type, identify target entities that enjoy the specified relation with the source entity and that satisfy the target type constraint.From a SW point of view, entity retrieval should be as simple as running SPARQL queries over structured data. However, since a true semantic web still has not been fully realized, the results of such queries are currently not sufficient to answer common information needs.

Chengkai Li, Ning Yan et al. [3] describe a faceted retrieval system for information discovery and exploration in Wikipedia. Given the set of Wikipedia articles resulting from a keyword query, Facetedpedia generates a faceted interface for navigating the result articles. Compared with other faceted retrieval systems, Facetedpedia is fully automatic and dynamic in both facet generation and hierarchy construction, and the facets are based on the rich semantic information from Wikipedia. The essence of our approach is to build upon the collaborative vocabulary in Wikipedia, more specifically the intensive internal structures and folksonomy. Given the sheer size and complexity of this corpus, the space of possible choices of faceted interfaces is prohibitively large. Authors propose metrics for ranking individual facet hierarchies by user's navigational cost, and metrics for ranking interfaces (each with facets) by both their average pairwise similarities and average navigational costs. Thus, develop faceted interface discovery algorithms that optimize the ranking metrics.

Wisam Dakka, Panagiotis G. Ipeirotis [4] describes an unsupervised technique for automatic extraction of facets useful for browsing text databases. In particular, observed through a pilot study, that facet terms rarely appear in text documents, showing that we need external resources to identify useful facet terms. For this, first identify important phrases in each document. Then, expand each phrase with "context" phrases using external resources, such as WordNet and Wikipedia, causing facet terms to appear in the expanded database.

Amaç Herdagdelen et al [5] describe a novel approach to query reformulation which combines syntactic and semantic information by means of generalized Levenshtein distance algorithms where the substitution operation costs are based on probabilistic term rewrite functions. We investigate unsupervised, compact and efficient models, and provide empirical evidence of their effectiveness. Further it explores a generative model of query reformulation and supervised combination methods providing improved performance at variable computational costs. Among other desirable properties, our similarity measures incorporate information-theoretic interpretations of taxonomic relations such as specification and generalization. Query reformulation is the process of iteratively modifying a query to improve the quality of search engine results, in order to satisfy one's information need. Search engines support users in this task explicitly; e.g., by suggesting related queries or query completions, and implicitly; e.g., by expanding the query to improve quality and recall of organic and sponsored results. Successful refinements are closely related to the original query. This is not surprising as reformulations involve spelling corrections, morphological variants, and tend to reuse parts of the previous query. More precisely, reformulations are close to the previous query both syntactically, as sequences of characters or terms,1 and semantically, often involving transparent taxonomic relations.

III. Methodology

3.1. Problem Definition

The information is contained in papers where knowledge is represented in text. Performing manual data extraction from text information is time-consuming and unreliable. This is because there is a large number of relevant articles. Moreover, consistency and reliability depend highly on the understanding of data extractors. For example, it depends on how much each extractor understands the extraction guideline, language, domain context, etc. These skills are uncontrollable, and hence it cannot guarantee whether the information is extracted with the same standards.

Web Document clustering has been analysised for use in a number of different areas of text mining and information retrieval. Initially, document clustering was analysis for improving the precision value or recall value in information retrieval systems and as an efficient way of learning the nearest neighbors of a document.

3.2. Classification

The proposed paper is focusing on the data extraction part, particularly, on the very first step of selecting the optimal data mining workflow for automatic classification of sentences. The classification divides sentences into a positive class which is a sentence that contains success factors and describes their relationships, and the negative class which is a sentence that does not contain such an information. The proposed system developing an application for recommendations of news articles to the readers of a news portal. The following challenges gave us the motivation to use clustering:

- The number of available articles was large.
- A large number of articles were added each day.
- Articles corresponding to same news be added from different sources.
- The recommendations had to be generated and updated in real time.

Clustering is a technique for automatically organizing or summarizing a large collection of text.

IV. Term Frequency – Inverse Document Frequency (TF-IDF)

The TF measures how frequently a particular term occurs in a document. It is calculated by the number of times a word appears in a document divided by the total number of words in that document. It is computed as **TF(the) = (Number of times term the 'the' appears in a document) / (Total number of terms in the document)**. The IDF measures the importance of a term. It is calculated by the number of documents in the text database divided by the number of documents where a specific term appears. While computing TF, all the terms are measured equally important. That means, TF counts the term frequency for normal words like "is", "a", "what", etc. Thus we need to know the frequent terms while scaling up the rare ones, by computing the following: **IDF(the) = log_e(Total number of documents / Number of documents with term 'the' in it)**.

For example, Consider a document containing 1000 words, in which the word give appears 50 times. The TF for give is then (50 / 1000) = 0.05. Now, assume that, 10 million documents and the word give appears in 1000 of these. Then, the IDF is calculated as log(10,000,000 / 1,000) = 4. The TF-IDF weight is the product of these quantities − 0.05 × 4 = 0.20.

4.1. Text Document Clustering

In this module, two documents are selected. Then the vector values for two documents are find out. Then the cosine similarity measure is applied. Then the correlation between two documents is found out using the following formula,

$$\mathbf{Corr(u,v) =[\ u^Tv\ /\ \sqrt{u^Tu}\ \sqrt{v^Tv}] = <\ u\ /\ ||u||,\ v/||v||\ >}$$

For example, the string "I have to go to school" is present in one document. the string "I have to go to temple" is present in other document. Then the data is prepared such that

4.2. Text Document Co-Clustering

In this module s is a non-symmetric measure of the difference between two probability distributions of two document *P* and *Q*. Specifically, the Kullback–Leibler divergence (KL Divergence) of *Q* from *P*, denoted $D_{KL}(P||Q)$, is a measure of the information lost when *Q* is used to approximate *P*.

The KL divergence measures the estimated number of extra bits required to code samples from *P* when using a code based on *Q*, rather than using a code based on *P*. Typically *P* represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution. The measure *Q* typically represents a theory, model, description, or approximation of *P*. Although it is often intuited as a metric or distance, the KL divergence is not a true metric, for example, it is not symmetric: the KL divergence from *P* to *Q* is frequently not the similar as that from *Q* to *P*. However, its infinitesimal form, specifically its Hessian, is a metric tensor: it is the Fisher information metric.

4.3 Multi Document Clustering

K-means clustering is a data mining the machine learning algorithm used to cluster observations into groups of related observations without any prior knowledge of those relationships. The cluster's mean is then recomputed and the process begins again. Here's how the algorithm works:

- The algorithm arbitrarily selects k points as the initial cluster centers ("means").
- Each point in the dataset is assigned to the closed cluster, based upon the Euclidean distance between each point and each cluster center.
- Each cluster center is recomputed as the average of the points in that cluster.

- Steps 2 and 3 repeat until the clusters converge. Convergence may be defined differently depending upon the implementation, but it normally means that either no observations change clusters when steps 2 and 3 are repeated or that the changes do not make a material difference in the definition of the clusters.

4.4. Classical k-Means Algorithm (Both Text, HTML and XML Documents)

```
procedure  KMEANS(X,K)
    {s1, s2, · · · , sk} Select Random Seeds(K,X)
    for  i  ← 1,K  do
        μ(Ci) ← si
    end for
    repeat
        min k~x n −~μ(C k )k  C k  = C k  [ {~x n }
        for all C k  do
            μ(C k ) = 1
        end for
    until  stopping criterion is met
end procedure
```

The proposed algorithm fall within a subcategory of the flat clustering algorithms, called Model-based clustering. The model-based clustering assumes that data were generated by a model and then tries to recover the original model from the data. This model then defines clusters and the cluster membership of data.

The proposed algorithm is a generalization of K-Means algorithm in which the set of K centroids as the model that generate the data.

4.5. Clustering With Side Information

The clustering text data with side information is a corpus S of text documents. The total number of documents is N, and they are denoted by T1 ... This is because the different values of the categorical attribute can be assumed to be separate binary attributes, whereas numerical data can be discretized to binary values with the use of attribute ranges. Some examples of such side-attributes are as follows:

- In a web log analysis application, we assume that xir corresponds to the 0-1 variable, which indicates whether or not the ith document has been accessed by the rth user.
- This information can be used in order to cluster the web pages in a site in a more informative way than a techniques which is based purely on the content of the documents. As in the previous case, the number of pages in a site may be large, but the number of documents accessed by a particular user may be relatively small.
- In a network application, we assume that xir corresponds to the 0-1 variable corresponding to whether or not the ith document Ti has a hyperlink to the rth page Tr.

4.6. Content and Auxiliary Attribute
[COATES Algorithm]

Content and auxiliary attribute-based text classification algorithm. The algorithm uses a supervised clustering approach in order to partition the data into k different clusters. This partitioning is then used for the purposes of classification. The steps used in the training algorithm are as follows:

- Feature Selection: In the first step, we use feature selection to remove those attributes, which are not related to the class label. This is performed both for the text attributes and the auxiliary attributes.
- Initialization: In this step, we use a supervised k-means approach in order to perform the initialization, with the use of purely text content. The main difference between a supervised k-means initialization, and an unsupervised initialization is that the class memberships of the records in each cluster are pure for the case of supervised initialization. Thus, the k-means clustering algorithm is modified, so that each cluster only contains records of a particular class.
- Cluster-Training Model Construction: In this phase, a combination of the text and side-information is used for the purposes of creating a cluster-based model. As in the case of initialization, the purity of the clusters in maintained during this phase.

Once the features have been selected, the initialization of the training procedure is performed only with the content attributes This approach is continued to convergence. The algorithm requires two phases:

- **Initialization:** We use a lightweight initialization phase in which a standard text clustering approach is used without any side-information. For this purpose the algorithm described. The centroids and the partitioning created by the clusters formed in the first phase provide an initial starting point for the second phase. We note that the first phase is based on text only, and does not use the auxiliary information.

- **Main Phase:** The main phase of the algorithm is executed after the first phase. This phase starts off with these initial groups, and iteratively reconstructs these clusters with the use of both the text content and the auxiliary information.

4.7. PSEUDO CODE FOR ATTRIBUTE CLUSTERING ALGORITHMS

```
Step 1:   Start the process
Step 2:   Dataset collection
Step 3:   Data cleaning term addition as stem word, stop word and synonym word addition
Step 4:   Add Attribute along with observation data.
Step 5:   Parse dataset or select gene dataset in text file.
Step 6:   Entropy calculation between attributes and observation data for probability analysis
Step 7:   To find conditional probability for attributes and observation data with the computation value of
          conditional entropy value, Mutual information value, significance of attribute with another
          attribute and supervised similarity values.
Step 8:   To calculate the relevance value for attributes and observation data with the threshold specification
          value.
Step 9:   Process of Supervised clustering algorithm with classification of finer cluster result from coarse
          cluster result.
```

V. Experimental Results Analysis

The following **Table 5.1** describes experimental result for COATES Algorithm in existing system analysis. The table contains weight of text document, weight of clustering text document and average of text document clustering details are shown.

S.NO	Weight of Document	Weight of Clustering Document	Average of Clustering Document [%]
1	200	155	77.5
2	250	220	88.00
3	300	272	90.66
4	350	322	92.00
5	400	383	95.75
6	450	429	95.33
7	500	468	93.60
8	550	523	95.05
9	600	578	96.33
10	650	633	97.74

Table 5.1 COATES Algorithm-Average Clustering Documents

The following **Fig 5.1** describes experimental result for existing system analysis. The table contains weight of text document, weight of clustering Text document and average of text document clustering details are shown.

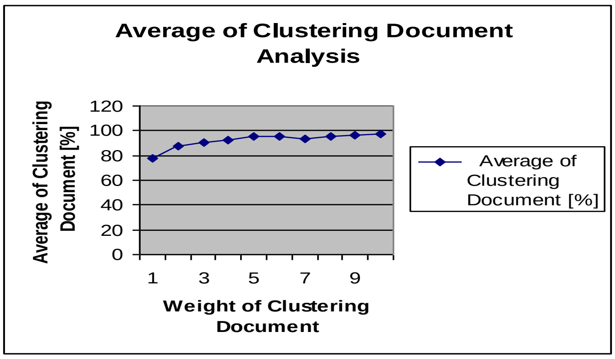

Fig 5.1 COATES Algorithm-Average Clustering Documents

The following **Table 5.2** describes experimental result for attribute selection clustering algorithm analysis. The table contains attribute, feature word and target concept word clustering details are shown

Attribute	Feature word	Target concept word
Adenoma1	204	614
BareNuclei	7	32
BlandChromatin	6	32
ClumpThickness	11	39
MarginalAdhesion	9	40
Mitoses	9	40
NormalNucleoli	11	32
SingleEpithelialCellSize	9	33
UniformityofCellShape	10	37
UniformityofCellSize	9	37

Fig 5.1 COATES Algorithm-Average Clustering Documents

The following **Figure 5.2** describes experimental result for attribute selection clustering algorithm analysis. The figure 5.2 contains attribute, feature word and target concept word clustering details are shown

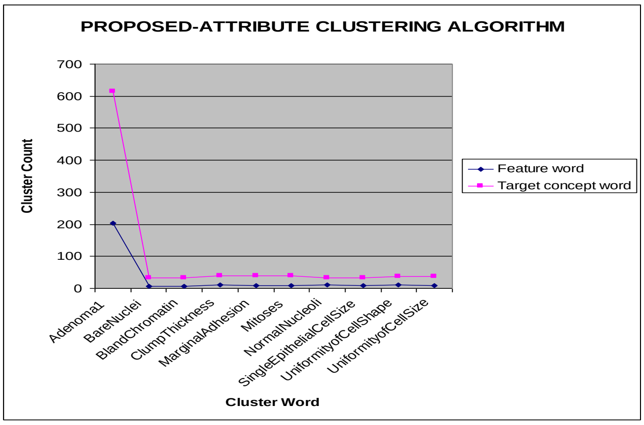

Fig 5.2 Proposed Attribute based Clustering Algorithms

VI. Conclusion

This proposed framework demonstrated how to construct various web document and word constraints and apply them to the constrained co-clustering process. A novel constrained co-clustering approach is proposed that automatically incorporates various word and web document constraints into information theoretic co-clustering. It describe the effectiveness of the proposed method for clustering textual documents. There are several directions for future works. The current anlaysis of unsupervised constraints is still preliminary. Furthermore, the proposed algorithm consistently output performed all the analysis constrained clustering and co-clustering methods under different conditions. The enhanced cosine similarity approach results in better clustering process. The future enhancements can be made for web documents of different languages. Investigation for better text features that can be automatically derived by using natural language processing or information extraction tools can be made.

Acknowledgements

My heartfelt gratitude goes to my beloved guide Mr. M. Baskar, Assistant Professor, Department of Computer Science, Vivekanandha College for Women, Tiruchengode, India for dedication and patience in assigning me her valuable advice and efforts during the course of my studies

References

[1] W. Kong and J. Allan, “Extending faceted search to the general web,” in Proc. ACM Int. Conf. Inf. Knowl. Manage., 2014, pp. 839–848.

[2] K. Balog, E. Meij, and M. de Rijke, “Entity search: Building bridges between two worlds,” in Proc. 3rd Int. Semantic Search Workshop, 2010, pp. 9:1–9:5.

[3] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, “Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia,” in Proc. 19th Int. Conf. World Wide Web, 2010, pp. 651–660.

[4] W. Dakka and P. G. Ipeirotis, “Automatic extraction of useful facet hierarchies from text databases,” in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 466–475.

[5] A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall, S. Riezler, and E. Alfonseca, “Generalized syntactic and semantic models of query reformulation,” in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. retrieval, 2010, pp. 283–290.

[6] [X. Xue and W. B. Croft, “Modeling reformulation using query distributions,” ACM Trans. Inf. Syst., vol. 31, no. 2, pp. 6:1–6:34, May 2013.

[7] L. Bing, W. Lam, T.-L. Wong, and S. Jameel, “Web query reformulation via joint modeling of latent topic dependency and term context,” ACM Trans. Inf. Syst., vol. 33, no. 2, pp. 6:1–6:38, eb. 2015.

[8] I. Szpektor, A. Gionis, and Y. Maarek, “Improving recommendation for long-tail queries via templates,” in Proc. 20th Int. Conf. World Wide Web, 2011, pp. 47–56.

[9] M. Damova and I. Koychev, “Query-based summarization: A survey,” in Proc. S3T, 2010, pp. 142–146.

[10] K. Latha, K. R. Veni, and R. Rajaram, “Afgf: An automatic facet generation framework for document retrieval,” in Proc.Int. Conf. Adv. Comput. Eng., 2010, pp. 110–114.

[11] J. Pound, S. Paparizos, and P. Tsaparas, “Facet discovery for structured web search: A query-log mining approach,” in Proc. ACM SIGMOD Int. Conf. Manage.Data,2011, pp. 169–180.

[12] [W. Kong and J. Allan, “Extracting query facets from search results,” in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2013, pp. 93–102.

[13] Y. Liu, R. Song, M. Zhang, Z. Dou, T. Yamamoto, M. P. Kato, H. Ohshima, and K. Zhou, “Overview of the NTCIR-11 imine task,” in Proc. NTCIR-11, 2014, pp. 8–23.