IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 9, September 2017

Survey of Automatically Mining Facets for Queries from Their Search Results

S. Saranya, M.C.A.[1], M. Baskar, M.Sc., M. Phil.[2],

Research Scholar, Computer Science, Vivekanandha College for Women, Tiruchengode, India[1]

Assistant Professor, Computer Science, Vivekanandha College for Women, Tiruchengode, India[1]

Abstract: QDMiner aims to offer the opportunity of finding the main points of multiple documents and thus save users' time on reading whole documents. The difference is that most existing summarization systems dedicate themselves to generating summaries using sentences extracted from documents. In addition, return multiple groups of semantically related items, while they return a flat list of sentences. However, the relative importance of this side-information may be difficult to estimate, especially when some of the information is noisy. In such cases, it can be risky to integrate side-information into the mining process, because it can either improve the quality of the representation for the mining process, or can add noise to the process. Therefore, a principled way is required to perform the mining process, so as to maximize the advantages from using this side information. This project designs an algorithm which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach.

Keyword: DQMiner, partitioning algorithms, summarization.

I. INTRODUCTION

Clustering is an automatic learning technique aimed at grouping a set of objects into subsets or else clusters. The goal is to create clusters that are coherent internally, but substantially different from each other. In plain words, objects in the same cluster should be as similar as possible, whereas objects in one cluster should be as dissimilar as possible from objects in the other clusters. Automatic document clustering has played an important role in many fields like information recovery, data mining, etc. The aim of this thesis is to improve the efficiency and accuracy of document clustering. In this proposed system two clustering algorithms and the fields where these perform better than the known standard clustering algorithms.

A. AUTOMATIC LEARNING TECHNIQUE

Clustering is a division of data into groups of related objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. In other words, the goal of a good document clustering scheme is to minimize intra-cluster distances between documents, while maximizing inter-cluster distances (using an appropriate distance measure between documents). A distance measure (or, dually, similarity measure) thus deception at the heart of document clustering.

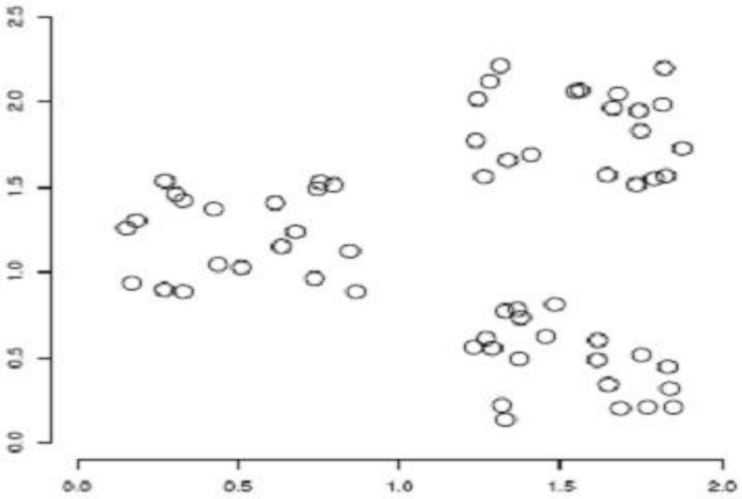

Figure 1.1 Cluster Structure

DOI10.17148/IJARCCE.2017.6909

Clustering is the most common form of unsupervised learning and this is the main difference between clustering and classification. No super-vision means that there is no human expert who has assign documents to classes. In clustering, it is the allocation and structure of the data that will determine cluster membership. Clustering is sometimes erroneously referred to as regular classification; however, this is inaccurate, since the clusters create are not known prior to processing whereas in case of classification the classes are pre-defined.

In clustering, it is the sharing and the nature of data that will conclude cluster membership, in opposition to the classification where the classifier learns the association among objects and classes from a so called training set, i.e. a set of data properly labeled by hand, and then replicates the learnt behavior on unlabeled data.

The aim of a document clustering method is to minimize intra-cluster distances between documents, even as maximizing inter-cluster distances (using an appropriate distance measure among documents). A distance measure (or, dually, similarity measure) thus lies at the heart of document clustering. The large variety of documents make it almost impossible to create a common algorithm which can work best in case of all kinds of datasets.

B. CHALLENGES IN DOCUMENT CLUSTERING

Document clustering is individual studied from many decades but at rest it is far from a trivial and solved problem. The challenges are:

- Selecting suitable features of the documents that should be used for clustering.
- Selecting an proper similarity measure between documents.
- Selecting an appropriate clustering method utilising the more than similarity measure.
- Implementing the clustering algorithm in an efficient way that makes it feasible in terms of required memory and CPU resources.
- Finding way of assessing the worth of the performed clustering.

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.Finally, the term distributions in the original database and the expanded database to identify the terms that can be used to construct browsing facets are compared. In order to support such exploratory interactions, the majority of the web sites mentioned above use a form of concept hierarchies to support browsing on top of large sets of items. Commonly, browsing is supported by a single hierarchy or a taxonomy that organizes thematically the contents of the database. Unfortunately, a single hierarchy can very rarely organize coherently the contents of a 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

1. 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 ... TN. It is assumed that the set of distinct words in the entire corpus S is denoted by W. Associated with each document Ti have a set of side attributes Xi. Every set of side attributes Xi has d dimensions, which are denote by (xi1 ... xid). We refer to such attributes as auxiliary attributes. For ease in notation and analysis, we assume that each side-attribute xid is binary, though both numerical and categorical attributes can easily be converted to this format in a fairly straightforward way. This is because the different ethics of the categorical element 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 study application, we think 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 amount of pages in a site may be huge, but the number of documents accessed by a particular user may be quite small.
- In a network application, we think that xir corresponds to the 0-1 variable related to whether or not the ith document Ti has a hyperlink to the rth page Tr. If desired, it can be absolutely assumed that each page links to itself in order to maximize linkage-based connectivity effects during the clustering process. Since hyperlink graphs are huge and sparse, it follows that the amount of such auxiliary variables are high, but only a small fraction of them take on the value of 1.
- In a document application with related GPS or provenance information, the likely attribute may be drawn on a large number of possibilities. Such attributes will obviously satisfy the sparsity property.

2. CONTENT AND AUXILIARY ATTRIBUTE [COATES Algorithm]

Content and secondary 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 purpose of classification. The steps used in the training algorithm are as follows:

- Element Selection: In the first step, we use feature selection to take out those attributes, which are not related to the class label. This is performed equally for the text attributes and the auxiliary attributes.

DOI10.17148/IJARCCE.2017.6909

- Initialization: In this step, we use a supervised k-means approach in order to execute the initialization, with the use of purely text content. The main difference among a supervised k-means initialization, and an unsupervised initialization is that the class memberships of the report in each cluster are pure for the case of supervised initialization. Thus, the k-means clustering algorithm is modified, so that each cluster only contain records of a particular class.
- Cluster-Training Model Construction: In this phase, a grouping 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 through this phase.

Once the features have been selected, the initialization of the training process is performed only by the content attributes. This is achieved by applying a k-means type algorithm as discussed to the approach, excluding that class label constraints are used in the process of assigning data points to clusters. Each cluster is associated with a particular class and all the records in the cluster go to that class. This goal is achieved by first create unsupervised cluster centroids, and then adding supervision to the process. In order to achieve this goal, the first two iterations of the k-means type algorithm are run in exactly the clusters are permitted to have different class labels. After the second iteration, each cluster centroid is strictly associated with a class label, which is identified as the majority class in that cluster at that point. In subsequent iterations, the records are controlled to only be assigned to the cluster with the associated class label. Each iteration for a given document, its distance is computed only to clusters which have the same label as the document. The document is then assigned to that cluster. This approach is continued to convergence.
The algorithm requires two phases:

- **Initialization:** We use a lightweight initialization phase in which a regular 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 supply an initial starting point for the second phase. We note that the first phase is based on text only, and does not apply the auxiliary information.
- **Main Phase:** The main phase of the algorithm is executed behind the first phase. This phase starts off among these initial groups, and iteratively reconstructs these clusters with the use of both the text content and the auxiliary information. This phase perform alternating iterations which use the text content and auxiliary attribute information in order to develop the quality of the clustering. We call these iterations as content iterations and auxiliary iterations correspondingly. The combination of the two iterations is referred to as a major iteration and each major iteration thus contains two minor iterations, corresponding to the auxiliary and text-based methods.

IV. CONCLUSION

This survey methods for mining text data with the use of side-information. Many forms of text-databases contain a large amount of side-information or meta-information, which may be used in order to improve the clustering process. Since the relative importance of this side-information may be difficult to estimate, especially when some of the information is noisy. In such cases, it can be risky to incorporate side-information into the mining process, because it can either improve the quality of the representation for the mining process, or can add noise to the process. Therefore, we need a principled way to perform the mining process, so as to maximize the advantages from using this side information.In order to design the clustering method, this project combined an iterative partitioning technique with a probability estimation process which computes the importance of different kinds of side-information. This general approach is used in order to design clustering algorithms. The proposed system also considers the cluster splitting into coarse and fine grained cluster so that most relevant documents are felt into fine cluster and other in coarse cluster in single given cluster. The results show that the use of side-information can greatly enhance the quality of text clustering, while maintaining a high level of efficiency.

ACKNOWLEDGMENT

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.

DOI10.17148/IJARCCE.2017.6909

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.

BIOGRAPHIES

Ms. Saranya working as M.Phil Scholar in the Department of Computer Science at Vivekanandha College for Women, Tiruchengode, India. She has obtained her Under Graduate Degree in Mathematics (CA) from Vivekanandha College for Women, Tiruchengode, India and Master Degree in Computer Application from K.s. Rangasamy college of Technology, Tiruchengode, India.

Mr. Baskar is currently working as Assistant Professor in the Department of Computer Science at Vivekanandha College for Women, Tiruchengode, India. He did his under graduate degree at Bharathidasan University, Trichy, India. He has obtained her Master degree in Bharathidasan University, Trichy, India. He did his M. Phil degree at Madurai Kamaraj University, Madurai, India. And he worked as the Lecturer in Thanthai Hans Roever College, Elambalur, from 2001 August to 2007 April and Assistant Professor in Vivekanandha college for women, Tiruchengode, India from 2007 June to 2009 july and University Of Goundar, Ethiopia from 2009 September to 2010 July.

DOI10.17148/IJARCCE.2017.6909