International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Efficient Clustering Technique for Information Retrieval in Data Mining

Anjali D. Raut[1], Prof. Prajakta. P. Chapke[2]
[1]*B.E. Final Year,* [1]*Computer Science and Engineering*
[2]*Computer Science and Engineering, H.V.P.M'S, C.O.E.T., Amravati,*

Abstract - In the emerging new wave of applications where people are the ultimate target of text clustering methods, cluster labels are intended to be read and comprehended by humans. The primary objective of a clustering method should be to focus on providing good, descriptive cluster labels in addition to optimizing traditional clustering quality indicators such as document-to-group assignment. In yet other words: in document browsing, text clustering serves the main purpose of describing and summarizing a larger set of documents; the particular document assignment is of lesser importance. Descriptive clustering is a problem of discovering diverse groups of semantically related documents described with meaningful, comprehensible and compact text labels. In this paper a new type of clustering methods for information retrieval which focuses on revealing the structure of document collections is discussed, summarizing their content and presenting this content to a human user in a compact way.

Keywords – data mining technique, clustering methods, clustering, Proposed techniques

I. INTRODUCTION

Just a few years ago size seemed to be everything the search engines weekly published numbers of Web pages indexed and available for immediate access to the public. But as the numbers grew into billions, size became just another incomprehensible factor. Nowadays electronic information sources on the Web include a great variety of different content. Alongside traditional Web pages in html, we have newswire stories, books, e-mails, blogs, source code repositories, video streams, music and even telephone conversations. Even narrowing the scope to textual content, the range of different possibilities is overwhelming.

A piece of text downloaded from the Internet is usually unstructured, multilingual, touching upon all kinds of subject (from encyclopedia entries to personal opinions) and in general unpredictable (think of all the typographical conventions, abbreviations, new words that come with electronic publications). We search for concrete pieces of information when we need to find a particular Web page, document, historical fact or a person.

This kind of information need has an interesting property: we can express it to a computer system using a query. The computer system may try to find a direct answer to our query (as in question-answering systems), but more often just locate a resource (document) that possibly contains the answer our query. The latter systems are called document-retrieval systems or in short search engines and in this paper our discussion concerns mostly programs of this type [2] and [3] and [4].

II. BACKGROUND

A single query typically matches a number of documents and a search engine must arrange the result into an ordered list, sorting documents according to their relevance to the query. This final list is often called a *hit list* and its topmost entries are shown to the user. Obviously, users rarely have the time to browse through all the returned documents and limit their effort to the hit list's topmost few entries, so calculating relevance is a key factor assuring documents most likely containing answers to the query are pushed up in the final ranking. These are the basic foundations of all search engines available at the time of writing this paper: use a very simple to find matching documents and rely on two things to provide valuable service to the user: qualitative ranking algorithms and, most of all, the user's ability to rephrase the query until his or her information need is satisfied.

The activity of exploring a collection of documents takes place when there is no information need or it is too vague to formulate a specific query. For example, imagine creating a query for the following question: given a set of previously unseen documents, what subjects are they about? An alternative task could be this: what subjects dominate the headlines of all major newspapers today? A human being could answer these questions simply by reading through all the available documents, but such solution is usually unacceptable as it requires too much time and effort. Exploration problems are also encountered in combination with search engines.

Queries issued to search engines are mostly short and ambiguous and match vast numbers of documents concerning variety of subjects. Creating a linear hit list out of such a broad set of results often requires trade-offs and hiding document that could prove useful to the user. If shown an explicit structure of topics present in a search result, users quickly narrow the focus to just a particular subset (slice) of all returned documents [2] and [6].

A. Document Clustering

Given a number of objects or individuals, each of which is described by a set of numerical measures, devise a classification scheme for grouping the objects into a number of classes such that objects within classes are similar in some respect and unlike those from other classes. The number of classes and the characteristics of each class are to be determined.

By analogy to the above definition *document clustering*, or *text clustering*, can be defined as a process of organizing pieces of textual information into groups whose members are similar in some way, and groups as a whole are dissimilar to each other. But before we delve into text clustering, let us take a look at clustering in general. There are many kinds of clustering algorithms, suitable for different types of input data and diverse applications. A great deal depends on how we define similarity between objects. We can measure similarity in terms of objects' proximity (distance), or as a relation between the features they exhibit.

1) *Feature Extraction:* This is used for extraction of features (important words and phrases in this case) from the documents. We have used Named-Entity tagger to extract the important words from the documents.

2) Feature Clustering: This is the most important phase in which the extracted features are clustered based on their co-occurrence.

3) Document Clustering: This is the final phase in which documents are clustered using the feature clusters. For this we have used a simple approach in which a document is assigned to the cluster of words of which it has the maximum word

B. *Overview of Selected Clustering*

Clustering analysis is a very broad field and the number of available methods and their variations can be overwhelming. A good introduction to clustering can be found in Cluster Analysis or in Cluster Classification. A more up-to-date view of clustering in the context of data mining is available in Data Mining: Concepts and Techniques.

C. *Partitioning Methods*

Suppose we are given a database of 'n' objects and the partitioning methods constructs 'k' partition of data .Each partition will represent a cluster and k<=n. it mean that it will classify the data into k groups ,which satisfy the following requirements- Each group contain at least one object. Each object must belong to exactly one group.

D. *Hierarchical Methods*

1) A family of hierarchical clustering methods can be divided into agglomerative and divisive variants

2) *Agglomerative* is a bottom up approach. Each observations starts in its own clusters, and pairs of clusters are merged as one moves up the hierarchy

3) *Agglomerative hierarchical clustering (ahc*) initially places each object in its own cluster and then iteratively combines the closes clusters merging their content.

4) *Divisive* is "top down" approach all observations start in one cluster, and splits are perform recursively as one moves down the hierarchy.

III. TECHNIQUE

The propose techniques are define in two parts, first is an clustering search results capable to discover diverse groups of documents and at the same time keep cluster labels sensible and second is Descriptive k-Means which is applicable to collections of several thousand short and medium documents.

In first phase, clustering algorithm processes the input in four phases: snippets preprocessing, frequent phrase extraction, cluster label induction and content allocation. The parallels to the generic scheme introduced in the DCF are illustrated in Figure.

Volume 4 Issue III, March 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

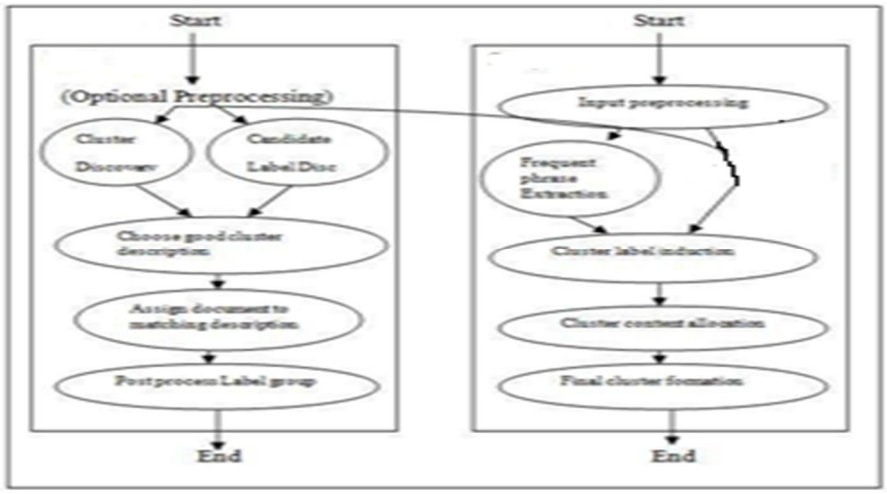

Figure1: Generic elements of DCF and their counterparts in Lingo.
The decomposition takes place inside cluster label induction phase; it is extracted here for clarity.

A. Input Preprocessing

Collection of Data includes the processes like indexing, filtering etc. which are used to collect the documents that needs to be clustered, index them to store and retrieve in a better way, and filter them to remove the extra data. Preprocessing is done to represent the data in a form that can be used for clustering. There are many ways of representing the documents like, graphical model etc. Many measures are also used for weighing the documents and their similarities.

IV. STEMMING

Stemming is the process of reducing words to their stem or root form. For example 'cook', 'cooking', 'cooked' are all forms of the same word used in different constraint but for measuring similarity these should be considered same.

A. Frequent Phrase Extraction

The aim of this step is to discover a set of cluster label candidates — phrases (but also single terms) that can potentially become cluster labels later. Lingo extracts frequent phrases using a modification of an algorithm presented. This allows to identify all frequent complete phrases in order of n time, n being the total length of all input snippets. The frequent phrase extraction algorithm ensures that the discovered labels fulfill the following conditions:

1) Appear in the input at least a given number of times.
2) Not cross sentence boundaries; sentence markers indicate a topical shift, therefore a phrase extending beyond one sentence is unlikely to be meaningful;
3) Be a complete frequent phrase (the longest possible phrase that is still frequent); compared to partial phrases, complete phrases should allow clearer description of clusters.
4) A Neither begins nor ends with a stop word; stop words that appear in the middle of phrase should not be discarded.

B. Cluster Label Induction

During the cluster label induction phase, Lingo identifies the abstract concepts (or dominant topics in the terminology used in DCF) that best describe the input collection of snippets. There are one steps to this: abstract concept discovery, phrase matching.

C. Cluster Content Allocation

The cluster content allocation explain throughout our segment on hard drives, as well as their structure and file system, the purpose of the file system itself is to organize data on the drive.

System uses specific technique for dividing storage on a disk volume into discrete areas so as to create a balance between efficient disk use and performance. This discrete area are referred as a cluster, and process by which files are assigned to cluster is called allocation. We discussed on what clusters are and how they are assigned to document. In order for FAT to manage file with some

form of efficiency is to group sectors into larger blocks referred to a cluster, or allocation unit.

D. Final Cluster Formation

Finally, clusters are sorted for display based on their score. The scoring function, although simple, prefers well-described and relatively large groups over smaller ones.

E. Descriptive k-Means algorithm

The second phase of propose technique is Descriptive k-Means closely follows the DCF approach. The cluster label discovery phase is implemented in two alternative variants: using frequent phrase extraction and with shallow linguistic processing for English texts (extraction of noun phrase chunks). Dominant topic discovery is performed by running a variant of k-Means algorithm on a sample of input documents.

F. Preprocessing

In the preprocessing step initialize two important data structures: an index of documents and an index of cluster candidate labels. An *index* is a fundamental structure in information retrieval. Each entry added to an index (document or candidate cluster label in our case) is accompanied by a vector of terms and their counts appearing in that entry. The index also maintains an associated list containing all unique terms and pointers to entries a given term occurred in (inverted index).

The index allows performing *queries* that is search for entries that contain a given set of terms and sort them according to weights associated with these terms.

By utilizing a document retrieval library that creates indices. Indices are essential in dkm to keep the processing efficient. Note that the index of documents is usually created anyway to allow searching in the collection and the index of cluster labels may be reused in the future, so the overhead of introducing these two auxiliary data structures should not be too big.

Each incoming document is segmented into tokens using the heuristic implemented. A unique identifier is assigned to the document and then it is added to an index *ID*.

If cluster candidate labels are to be extracted directly from the input documents, this process takes place concurrently to document indexing. Depending on the variant of dkm, extract frequent phrases or noun phrases (from English documents). The resulting set of candidate labels is added to a separate index *IP*. Each candidate cluster label is indexed as if it were a single document. To minimize the number of identical index entries, keep a buffer of unique labels in memory and flushing them to the index in batches.

G. Preparation of Document Vector

Let us recall that the index contains a vector of terms and their occurrences for each document. Depending on the input size, we either take all documents or select a uniform random subset and fetch their feature vectors from the index. To speed up computations we weight all the features and then limit the number of features for each selected document to a given number of most significant terms to make document vectors even more sparse.

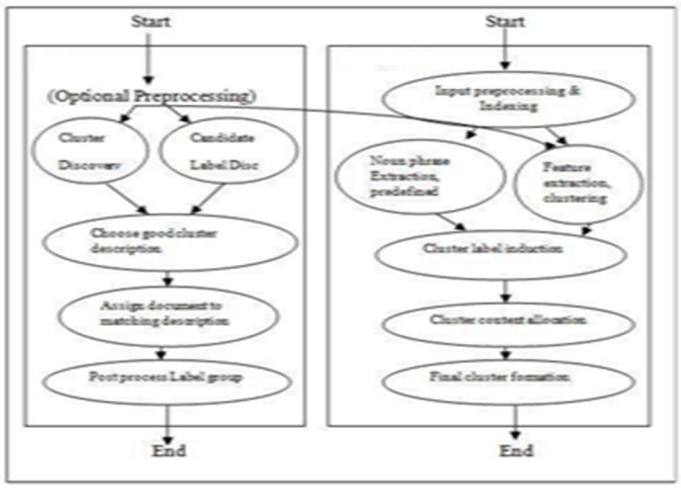

Figure 2: Generic elements of DCF and their counterparts in Descriptive k-Means

H. Selection of the Desired Number of Cluster

Any clustering methods, whatever the algorithm ,allows you to choose the number of cluster you want. Either by specifying The K-means method or selecting this number using k-means method .clustering methods is not driven by wish to get data known number of cluster. It is driven by nature of your data such as numeric, binary? assumption about the distribution or shape of cluster.

V. CONCLUSION

I started this paper from the observation that clustering methods in such applications are inevitably connected with finding concise, comprehensible and transparent cluster labels a goal missing in the traditional definition of clustering in information retrieval.

I collected the requirements and expectations to formulate a problem of *descriptive clustering* a document grouping task emphasizing comprehensibility of cluster labels and the relationship between cluster labels and their documents.

Next, i devised a general method called *Description Comes First*, which showed how the difficult step of describing a model of clusters can be replaced with extraction of candidate labels and selection of pattern phrases labels that can function as an approximation of a dominant topics present in the collection of documents. clustering results returned by search engines and clustering larger collections of longer documents such as news stories or mailing lists. The paper ends with a presentation of results collected from empirical experiments with the two presented algorithms.

The motivation for this paper arose as a consequence of observing new applications of clustering methods in information retrieval and the needs of real users using these applications. my initial goals were to create a method able to accurately describe existing clusters, but they soon changed when i realized that the problem itself needs to be rewritten to permit sensible solutions. The definition of descriptive clustering is, in our opinion, a better way of reflecting the needs of a user who needs to browse a collection of texts, whether they are snippets or other documents.

Moreover, I show that dcf combined with smart candidate label selection (noun phrases, for example), allows easier resolution to the problem of cluster labeling that are more likely to fulfill the requirements of descriptive clustering defined at the beginning of this paper, especially comprehensibility and transparency.

REFERENCES

[1] El Oirrak, and D. Aboutajdine, “Combining BOW representation and Appriori algorithm for Text mining”, IEEE 2010.[2] Jiabin Deng, JuanLi Hu, Hehua Chi, and Juebo Wu, “An Improved Fuzzy Clustering Method for Text Mining”, IEEE 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing, pp. 65-69.

[2] Bin XU and Yufeng Zhang, “A New Polarity Clustering AlgorithmBased on Semantic Criterion Function For Text of the Chinese Commentary”, IEEE 20I0 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), pp. V4-116-119.

[3] Yan Zhang, Mingyan Jiang, “Chinese Text Mining Based on Subspace Clustering”, IEEE 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010), pp. 1617-1620.

[4] Mohamed Yassine, and Hazem Hajj, “A Framework for Emotion Mining from Text in Online Social Networks”, IEEE 2010 International Conference on Data Mining Workshops, pp. 1636-1642.

[5] Hui Yang , Bin Yang, Xu Zhou, Chunguang Zhou, and Zhou Chai, “Community Discovery and Sentiment Mining for Chinese BLOG”, IEEE 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010), pp. 1740-1745.

[6] Lijun Wang, Manjeet Rege, Ming Dong, and Yongsheng Ding, “Low-rank Kernel Matrix Factorization for Large Scale Evolutionary Clustering”, IEEE 2010.

[7] Rekha Baghel and Renu Dhir, “Text Document Clustering Based on Frequent Concepts”, IEEE 2010 1st International Conference on Parallel, Distributed and Grid Computing (PDGC - 2010), pp. 366-371.

[8] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka, “A Web Search Engine-based Approach to Measure Semantic Similarity between Words”, IEEE 2010.

[9] Farid Bourennani, Mouhcine Guennoun, Ying Zhu, “Clustering Relational Database Entities using K-means”, IEEE 2010 Second International Conference on Advances in Databases, Knowledge, and Data Applications, pp. 143-148.