International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 2, February- 2013

Analysis and Comparison of Web Document Clustering Algorithms with Lingo

Pragna Makwana
Prof. Neha Soni

Department of Computer Engineering, SVIT Vasad,Gujarat,India

Abstract

In Today's world, with the increased use of internet we have large amount of shared information on World Wide Web. To access small piece of relevant information from this largest repository is overwhelming. Even with the use of search engines, it is difficult to find the most relevant documents from the returned list of large number of documents in response to the user query. Sometimes, users with the absence of domain expertise gives the more abstract query terms, and it leads to the more irrelevant pages and the most relevant pages do not necessarily appear at the top of the query output sequence. It forces the need of documents clustering using the snippet returned by the query. In this paper we discussed various clustering methods, document clustering and web document clustering algorithm and their comparison with lingo algorithm.

1. Introduction

With the increased use of internet we have large amount of shared information on World Wide Web. To access small piece of relevant information from this largest repository is overwhelming. Even with the use of search engines, it is difficult to find the most relevant documents from the returned list of large number of documents in response to the user query. Sometimes, users with the absence of domain expertise gives the more abstract query terms, and it leads to the more irrelevant pages and the most relevant pages do not necessarily appear at the top of the query output sequence.

This has led to the need for the development of new techniques to assist users effectively navigate, trace and organize the available web documents, with the ultimate goal of finding those best matching their needs. Document clustering is the one of important technique to achieve this objective. Various document clustering algorithms are available nowadays.

The key points for web document clustering algorithms are as follows [7].

Relevance: The algorithm ought to produce clusters that group documents relevant to the user's query separately from irrelevant ones.

Browsable Summaries: The user needs to determine at a glance whether a cluster's contents are of interest. We do not want to replace sifting through ranked lists with sifting through clusters. Therefore the algorithm has to provide concise and accurate descriptions of the clusters.

Overlap: Since documents have multiple topics, it is important to avoid confining each document to only one cluster.

Snippet tolerance: The algorithm ought to produce high quality clusters even when it only has access to the snippets returned by the search engines, as most users are unwilling to wait while the system downloads the original documents off the Web.

Speed: As the algorithm will be used as part of an on-line system, it is crucial that it does not introduce noticeable delay to the query processing. Clustering aims at allowing the user to browse through at least an order of magnitude more documents compared to a ranked list.

Incrementality: To save time, the algorithm should start to process each snippet as soon as it is received over the Web.

Special requirements for web document clustering:

Dimensionality: The number of relevant terms in a document set is typically in the order of thousands. Each of these terms constitutes a dimension in a document vector. Natural clusters usually do not exist in the full dimensional space, but in the subspace formed by a set of correlated dimensions. Locating clusters in subspaces can be challenging.

Scalability: Real world data sets may contain hundreds of thousands of documents. Many clustering algorithms work fine on small data sets, but fail to handle large data sets efficiently.

Accuracy: A good clustering solution should have high intra-cluster similarity and low inter-cluster similarity, i.e., documents within the same cluster should be similar but are dissimilar to documents in other clusters. An external evaluation method, the F-measure is commonly used for examining the accuracy of a clustering algorithm.

Browse with Meaningful Cluster Description:
The resulting topic hierarchy should provide a sensible structure, together with meaningful cluster descriptions, to support interactive browsing.

Prior Domain Knowledge: Many clustering algorithms require the user to specify some input parameters, e.g., the number of clusters. However, the user often does not have such prior domain knowledge. Clustering accuracy may degrade drastically if an algorithm is too sensitive to these input parameters.

This paper includes the details regarding web document clustering and analysis of its algorithms and also includes the steps for lingo algorithm.

2. Basics of Clustering

Clustering can be considered the most important unsupervised learning problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data.
A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way".
A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. The basic requirements of clustering are scalability, dealing with different types of attributes, discovering objects with different shapes, ability to deal with noise and outliers, high dimensionality usability. Clustering is a form of unsupervised classification, which means that the categories into which the collection must be partitioned are not known, and so the clustering process involves the discovering of these categories.

3. Basics of Document Clustering

Document clustering is an automatic grouping of text documents into clusters so that documents within a cluster have high similarity in comparison to one another, but are dissimilar to documents in other clusters. Unlike document classification, no labeled documents are provided in clustering; hence, clustering is also known as unsupervised learning.

In order to cluster documents, one must first choose the type of the characteristics or attributes (e.g. words, phrases or links) of the documents on which the clustering algorithm will be based and their representation. The most commonly used model is the Vector Space Model. Vector Space Model is a mathematical model to represent Information Retrieval Systems which uses term sets to represent both documents and queries, employs basic linear algebra operations to calculate global similarities between them.

4. Web Document Clustering Algorithms

Clustering of web search results is an attempt to organize the results into a number of thematic groups in the manner a web directory does it. This approach, however, differs from the human-made directories in many aspects. First of all, only documents that match the query are considered while building the topical groups. Clustering is thus preformed after the documents matching the query are identified. Consequently, the set of thematic categories is not fixed – they are created dynamically depending on the actual documents found in the results. Secondly, as the clustering interface is part of a search engine, the assignment of documents to groups must be done efficiently and on-line. For this reason it is unacceptable to download the full text of each document from the Web – clustering ought to be performed based solely on the snippets returned by the search service [5].

4.1 Agglomerative Hierarchical Clustering (AHC) Algorithm

The basic process of hierarchical clustering:

1. If you have n items them make n clusters and assign each item to a cluster. Each cluster should have just one item.
2. Find the most similar pair of clusters and merge them into a single cluster, so that now you have one cluster less.
3. Compute similarities between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Step 3 can be done in different ways, which is what distinguishes single-linkage from complete-linkage and average-linkage clustering. In single-linkage clustering (also called the connectedness or minimum method). we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster. In complete-linkage clustering (also called the diameter or maximum method), we consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster. In average-linkage clustering, we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster. This kind of hierarchical clustering is called agglomerative because it merges clusters iteratively.

The main problem with AHC, that, they are very slow with large amount of data provided and also very sensitive with halting criterion that is, by mistake it can merge valuable clusters into one cluster. Also they do not scale well. They can never undo what was previously done. With outliers it performs poorly [3].

4.2 K-Means Algorithm

This algorithm is based on the center locations .It first finds out the k cluster center location. Then each data point finds out which center is closest to it. Each center finds the centroid of the points and jumps to there. The main benefit of K-means algorithm is that, it is capable to produce overlapping clusters. Its main disadvantage is that it is most effective when the desired clusters are approximately spherical with respect to the similarity measure used. There is no reason to believe that documents should fall into approximately spherical clusters.

4.3 Suffix Tree Clustering (STC)

STC includes 2 main steps. First it searches for all sets of documents that share a common phrase. They are found by suffix tree data structure. In second step we merge these phrases into cluster. The merge process is dependent on the percentage of the documents that contain both phrases. It also allows overlapping clusters.STC uses simple cluster definition. Also, STC is a fast incremental linear time algorithm which makes it suitable for web search clustering. It is faster than K-Means. The main benefit of Suffix Tree Clustering is that it uses phrases to provide concise and meaningful descriptions of groups. But needs some thresholds for cluster formation and they turn out particularly difficult to tune. Its main disadvantage is it removes longer high quality phrases and use only shorter phrases. Finally, if a document does not include any of the extracted phrases or just some parts of them, it will not be included in the results although it may still be relevant.

4.4 Semantic Hierarchical Online clustering (SHOC)

The Semantic Online Hierarchical Clustering is a web search results clustering algorithm that uses variation of the Vector Space Model called Latent Semantic Indexing (LSI) and uses phrases in the process of clustering. Unlike STC, SHOC improves the quality of label.STC gives incomplete labels while SHOC gives complete phrases. With SHOC documents can belong to several clusters. SHOC includes two key concepts: Complete phrases and definition of continuous clusters. It should meet the three requirements: Semantic, Hierarchical, and Online. It has three steps: 1. Data collection and cleaning 2. Feature extraction and 3. Identifying and organizing clusters.

Problems with SHOC

The problem with SHOC is that it provides only vague comments on the values of thresholds of their algorithm and the method which is used to label the resulting clusters. It uses the singular value decomposition. So it may create unintuitive, random

continuous clusters. It might be because of the input snippets used in that [3].

4.5 Lingo Algorithm

The Lingo algorithm is used by the Carrot2 web searcher and is based on complete phrases and LSI. Lingo is an enhancement of SHOC and STC and unlike most of the algorithms, it first discover descriptive names for the clusters and then, assigns the documents into appropriate clusters. One disadvantage with this algorithm is that the topic separation phase usually requires algebraic transformations that demand a lot of computing time, using Singular Value Decomposition. The phases of lingo algorithm are described below [1, 2].

Phases of Lingo Algorithm

1. Preprocessing:
The pre processing phase includes stemming, stop words and stop labels. Stemming is the process of folding grammatical variations of words into their "base" forms. Carrot2 uses built in set of stemmers. Stop words include the terms that are meaningless in the language (i.e. "is", "this" in English).It is often desirable to filter out certain frequently occurring expressions that should not be considered as cluster. This resource provides means of avoiding such cluster labels.

2. Frequent Phrase Extraction:
The frequently occurring terms and phrases in documents are found in this phase. There are some predefined thresholds given. Frequency of the terms and phrases should exceed these threshhold values then and then it can be considered as a frequently occurred terms and phrases. The advanced method adds an extra step that involves finding the synonyms of the frequent terms and phases.

3. Cluster Label Induction:
This phase of lingo first computes the term document matrix for the frequent terms. After that it decomposes this term document matrix using singular value decomposition. Then using this decomposed matrix it finds the abstract concepts from document and then apply phrase matching. The abstract concept can then be used as cluster labels according to some thresholds.

4. Cluster Content Discovery:

This phase of lingo then assigns the content of the document or the input snippets to the clusters which are labeled in previous phase.

5. Final Cluster Formation:

Finally, the clusters are scored using label score and member count. Then clusters are sorted according to these cluster scores.

In Lingo, as input data, snippets are used. Snippet can be the most probable terms and phrases that describe the whole document or can the first few lines of the document. This is the main difference between lingo and other clustering algorithms. Also Lingo first finds the label of the cluster and then assigns the content to the cluster that is the description comes first approach.

Comparison of Web Document Clustering Algorithms

Algorithm	Cluster Diversity	Cluster labels	Scalability	Time Complexity	Advantages	Disadvantages
Agglomerative Hierarchical Clustering (AHC)	not very robust towards outliers	Most frequent terms	Low	Single link and group average: $O(n^2)$ Complete link: $O(n^3)$	Simple	-Slow when applied to large document collections. -Sensitive to halting criterion. -Poor performance in domains with many outliers.
K- means	Low, small (outlier) clusters rarely highlighted[8]	One-word only, may not always describe all documents in the cluster	Low, based on similar data structures as Lingo	O(nkt) (k:initial clusters, t: iterations)	-Efficient and simple. -Suitable for large datasets.	-Very sensitive to input parameters.
Suffix Tree Clustering (STC)	Low, small (outlier) clusters rarely highlighted[8]	Shorter, but still appropriate	High	O(n)	-Incremental -Uses phrases to provide concise and meaningful description of groups.	-Snippets usually introduce noise. -Snippets may not be a good description of a web page.
Semantic Online Hierarchical Clustering (SHOC)	Low	Label that describe the cluster	High	O(n)	-Uses Latent Semantic Indexing (LSI) and phrases in the process of clustering. -Uses suffix array to identify complete phrases. -Allows overlapping clusters. -Provides a method of ordering documents	-Provides only vague Comments on the values of thresholds of the algorithm and the method which is used to label the resulting clusters.

					within clusters.	
Lingo	High, many small (outlier) clusters highlighted[8]	Longer, often more descriptive	Low. For more than about 1000 documents, Lingo clustering will take a long time and large memory	O(n)	-Readable cluster Labels. Overlapping clusters. -Cluster accuracy.	-Unable to generate a Hierarchical structure of clusters. -The implementation of lingo is fairly computationally expensive.

Table 1 Comparison of Web Document Clustering algorithms

5. Conclusion

Clustering can increase the efficiency and the effectiveness of information retrieval. The fact that the user's query is not matched against each document separately, but against each cluster can lead to an increase in the effectiveness, as well as the efficiency, by returning more relevant and less non relevant documents. The organization and presentation of the pages in small and meaningful groups (usually followed by short descriptions or summaries of the contents of each group) gives the user the possibility to focus exactly on the subject of his interest and find the desired documents more quickly. Thus document clustering is very useful to retrieve information application in order to reduce the consuming time and get high precision and recall. This paper has presented comparison of various algorithms that support web document clustering.

6. References

[1] Stanislaw Osinki, "An Algorithm for Clustering of Web search results".

[2] Stanislaw Osinski, Dawid Weiss, "A Concept-Driven Algorithm for Clustering Search Results".

[3] K. Sridevi, R. Umarani, V.Selvi, "An Analysis of Web Document Clustering Algorithms".

[4] Michael Steinbach, George Karypis, Vipin Kumar, "A Comparison of Document Clustering Techniques".

[5] N. Oikonomakou, M. Vazirgiannis, "A Review of Web Document Clustering Approaches".

[6] Ahmed Sameh, Amar Kadray, "Semantic Web Search Results Clustering Using Lingo and WordNet".

[7] Stanislaw Osinski, "Dimensionality Reduction Techniques for Search Results clustering", Master Thesis, Department of Computer Science, The University of Sheffield, UK, 2004.

[8] http://download.carrot2.org/head/manual/index.html