International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.3, No.2, February 2014 DOI:10.15693/ijaist/2014.v3i2.111-114

UDH: UNL Based Document Clustering Using a Hybrid Clustering Approach

Mausumi Goswami
Assistant Professor/Department of CSE
Christ University, Bangalore, India

Bipul Syam Purkayastha
Professor/Department of Computer Science
Assam University, Silchar

Abstract— **Agglomerative hierarchical clustering and K-means are the two main approaches to document clustering. For K-means we may use a "standard" K-means algorithm and also a variant of K-means, "bisecting" K-means. Hierarchical clustering is often portrayed as the better quality clustering approach, but is limited because of its quadratic time complexity. In contrast, K-means and its variants have a time complexity which is linear in the number of documents, but are thought to produce inferior clusters. In traditional document representation methods, the feature vector representing the document is constructed from the frequency count of document terms. In this paper we propose a hybrid clustering technique. We generate feature vectors using Universal Networking Language (UNL) and then clustering of the documents using a hybrid clustering technique.**

Index terms - UNL, feature vector, hybrid approach

I. Introduction

In traditional document clustering methods, a document is considered a bag of words. The fact that the words may be semantically related, crucial information for clustering is not taken into account. The feature vector representing the document is constructed from the frequency count of document terms. To improve results, weights calculated from techniques like *Inverse Document Frequency (IDF)* and *Information Gain (IG)* are applied to the frequency count. These weights also are essentially statistical parameters and do not make use of any semantic information.

It may not be currently feasible to make use of the full meaning of a document; we can still extract semantic information from the properties of words, relations between words and/or the structure of a document. This information is then employed to classify and categorize the documents. Clustering has the advantage that a priori knowledge of categories is not required, and so the categorization process is unsupervised. The results of clustering could then be used to automatically formulate queries and search for other similar documents on the Web.

A. Document Representation using Term Frequency

A.1 Data preprocessing steps

This is the first part of feature extraction and it involves removal of stop words, stemming, followed by term weighting. The document is parsed through to find out the list of all the words. The next process in this step is to reduce the size of the list created by the parsing process, An easy way to comply with the conference paper generally using methods of stop words removal and stemming.

Stop words removal

This is the first step in preprocessing which will generate a list of terms that describes the document satisfactorily. The document is parsed through to find out the list of all the words. The stop words removal accounts to 20% to 30% of total words counts. Stop words are removed from each of the document by comparing the with the stop word list. This process reduces the number of words in the document significantly since these stop words are insignificant for search keywords. Stop words can be pre-specified list of words or they can depend on the context of the corpus.

Stemming

The next process in phase one after stop word removal is stemming. Stemming is process of linguistic normalization in which the variant forms of a word is reduced to a common form. For example: the word, connect has various forms such as connect, connection, connective, connected, etc., Stemming process reduces all these forms of words to a normalized word connect. Algorithm for stemming should be surveyed first and would be chosen there after.

B. Document Representation by using Tf-IDF (term frequency – inverse document frequency)

A Document is represented by a set of keywords/ terms extracted from the document. The collection or union of all set of terms is the set of terms that represents the entire collection and defines a 'space' such that each distinct term represents one dimension in that space. A term-document matrix can be encoded as a collection of n documents and m terms.

c.1 Tf–idf, term frequency–inverse document frequency, is a numerical statistic which reflects how important a word is to a document collection or corpus. It is used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others. Variations of the *tf–idf weighting scheme* are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including *text summarization and classification*.

Tf–idf is the product of two statistics, term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist. Following section some of the techniques are described.

Definition : term frequency by counting raw frequency

Term frequency tf(*t*,*d*) the simplest choice is to simply use the *raw frequency* of a term in a document, i.e. the number of times that term *t* occurs in document *d*. If we denote the raw frequency of *t* by f(*t*,*d*), then the simple tf scheme is tf(*t*,*d*) = f(*t*,*d*).

Definition : term frequency by Boolean frequencies

Boolean "frequencies": tf(*t*,*d*) = 1 if *t* occurs in *d* and 0 otherwise;

Definition : term frequency by logarithmically scaled frequency

Logarithmically scaled frequency: tf(*t*,*d*) = 1 + log f(*t*,*d*) (and 0 when f(*t*,*d*) = 0);

Definition : term frequency by normalized frequency

normalized frequency is used to prevent a bias towards longer documents, e.g. raw frequency divided by the maximum raw frequency of any term in the document:

$$\mathbf{tf}(t,d) = \frac{\mathbf{f}(t,d)}{\mathbf{max}\{\mathbf{f}(w,d) : w \in d\}}$$

The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient..

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

with

- $|D|$: cardinality of D, or the total number of documents in the corpus
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\mathrm{tf}(t,d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the formula to $1 + |\{d \in D : t \in d\}|$.

Mathematically the base of the log function does not matter and constitutes a constant multiplicative factor towards the overall result.

Then tf–idf is calculated as

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

$$= \mathrm{tf}(t,d) \ \mathrm{x} \ 1 + |\{d \in D : t \in d\}|.$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0.

II. RELATED WORK

Universal Networking Language (UNL) [Uchida, Zhu and Della 1995] is a semantic representation of a document, which expresses the document in the form of a graph. Information written in a natural language may be enconverted to UNL and the UNL can be deconverted into a target natural language. The UNL representation defines a semantic net [Woods 1985] like structure. The meaning is represented sentence by sentence in the form of a hyper graph having concepts as nodes and relations as directed arcs. Concepts are represented as character-strings called *Universal Words (UWs).*All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

The knowledge within a document is represented in three dimensions:

1. Word Knowledge is expressed by Universal Words (UWs), which are language independent. These UWs are restricted using constructs, which describe the sense of the word in the current context.

III. OBJECTIVES & OVERVIEW OF THE PROPOSED MECHANISM

In this section we provide a brief overview of hierarchical and partitional (K-means) clustering techniques. Hierarchical techniques produce a nested sequence of partitions, with a single, all inclusive cluster at the top and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level (or splitting a cluster from the next higher level). The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendogram. This tree graphically displays the merging process and the intermediate clusters. The dendogram at the right shows how four points can be merged into a single cluster. For document clustering, this dendogram provides a taxonomy, or hierarchical index. There are two basic approaches to generating a hierarchical clustering:

a) Agglomerative: Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.

b) Divisive: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

Agglomerative techniques are more common, and these are the techniques that we will compare to K-means and its variants [A. El-Hamdouchi and P. Willet 89] . We summarize the traditional agglomerative hierarchical clustering procedure as follows:

Simple Agglomerative Clustering Algorithm

1. Compute the similarity between all pairs of clusters, i.e., calculate a similarity matrix whose *ijth* entry gives the similarity between the *ith* and *jth* clusters.

2. Merge the most similar (closest) two clusters.
3. Update the similarity matrix to reflect the pairwise similarity between the new cluster and the original clusters.
4. Repeat steps 2 and 3 until only a single cluster remains.

Agglomerative hierarchical clustering and K-means are two clustering techniques that are commonly used for document clustering. Agglomerative hierarchical clustering is often portrayed as “better” than K-means, although slower. A widely known study, discussed in [Richard C. Dubes and Anil K. Jain88], indicated that agglomerative hierarchical clustering is superior to K-means, although we stress that these results were with non-document data. In the document domain, Scatter/Gather [Douglass R. Cutting, David R. Karger92], a document browsing system based on clustering, uses a hybrid approach involving both K-means and agglomerative hierarchical clustering. K-means is used because of its efficiency and agglomerative hierarchical clustering is used because of its quality. In contrast to hierarchical techniques, partitional clustering techniques create a one-level (un-nested) partitioning of the data points. If *K* is the desired number of clusters, then partitional approaches typically find all *K* clusters at once. Contrast this with traditional hierarchical schemes, which bisect a cluster to get two clusters or merge two clusters to get one. Of course, a hierarchical approach can be used to generate a flat partition of *K* clusters, and likewise, the repeated application of a partitional scheme can provide a hierarchical clustering. There are a number of partitional techniques, but we shall only describe the K-means algorithm which is widely used in document clustering. K-means is based on the idea that a center point can represent a cluster. In particular, for K-means we use the notion of a centroid, which is the mean or median point of a group of points.

The basic K-means clustering technique is presented below. Basic K-means Algorithm for finding *K* clusters.

1. Select *K* points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don’t change.

IV. PROPOSED METHOD

5.1 Document Vector Construction Using UNL Graph Links

In the UNL link method, instead of using the words as components for the document vector we use the Universal Words- which are concepts formed using English words and attaching restrictions to them- as the components of the vector. Since each Universal Word is disambiguated (for example the financial bank is represented as *bank (icl>financial institute)* and the river bank is represented as *bank (mod>river)* in UNL), multiple words in the document get automatically differentiated, thereby producing correct frequency count.

After this, each component of the document vector- that represents a different universal word (*i.e.*, a concept) is assigned the number of links incident on the node, considering the graph to be undirected. When a UW is not present in the UNL graph of the document then 0 is written in its position. The basic assumption behind this approach of counting the links is that *the more number of links to and from a universal word, the more is the importance of the word in the document.*

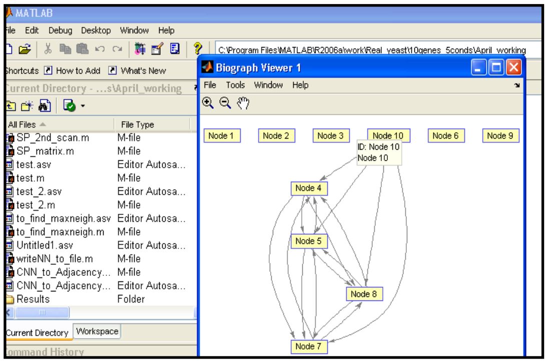
5.2 Clustering step for finding K clusters.

1. Pick a cluster to split.
2. Find 2 sub-clusters using the basic K-means algorithm. (Bisecting step)
3. Repeat step 2, the bisecting step, for ITER times and take the split that produces the clustering with the highest overall similarity.
4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

.

V. RESULTS

Vectors of documents were created using the term frequency, the UNL link methods. Then they are to be clustered using K-means algorithm. Bisecting K-means might work better than regular K-means. It is found, main reason for this is that bisecting K-means tends to produce clusters of relatively uniform size, while regular K-means is known to produce clusters of widely different sizes. Smaller clusters are often of higher quality, but this doesn’t contribute much to the overall quality measure since quality measures weight each cluster’s quality contribution by the cluster’s size. Larger clusters, on the other hand, tend to be of lower quality and make a large negative contribution to cluster quality. A snapshot of the intermediate result using the proposed method is shown below.



C:\Program Files\MATLAB\R2006a\work\Real_yeast\10genes_5conds\April_working

All Files	File Type
SP_2nd_scan.m	M-file
SP_matrix.m	M-file
test.asv	Editor Autosa...
test.m	M-file
test_2.asv	Editor Autosa...
test_2.m	M-file
to_find_maxneigh.asv	Editor Autosa...
to_find_maxneigh.m	M-file
Untitled1.asv	Editor Autosa...
writeNN_to_file.m	M-file
CNN_to_Adjacency...	M-file
CNN_to_Adjacency...	Editor Autosa...
Results	Folder

REFERENCES

[1] **Uchida H., Zhu M., Della Senta T.** *UNL: A Gift for a Millennium.* The United Nations University, 1995

http://www.unl.ias.unu.edu/publications/gm/index.html.

[2] Woods William A. *What's in a Link: Foundation for Semantic Networks.* in Readings in Knowledge Representation, R.J. Brachman and H.J.Levesque (ed.), Morgan Kaufmann Publishers, 1985.

[3] J.A Hartigan and M. A. Wong. *A k-means clusteringalgorithm.* Applied Statistics, 28:100--108, 1979.

[4] Richard C. Dubes and Anil K. Jain, *Algorithms for Clustering Data*, Prentice Hall, 1988.

[5] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey, *Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections*, SIGIR '92, Pages 318 – 329, 1992.

[6] A. El-Hamdouchi and P. Willet, Comparison of Hierarchic Agglomerative Clustering Methods forDocument Retrieval, The Computer Journal, Vol. 32, No. 3, 1989.

[7] A.P. Dempster, N.M. Laird, and D.B. Rubin. *Maximum Likelihood from Incomplete Data via the EM Algorithm*. Journal of the Royal Statistical Society, Series B (Methodological) , 39(1):1--38, 1977.

[8] Julio Gonzalo, Felisa Verdejo, Irina Chugur, Juan Cigarran *Indexing with WordNet synsets can Improve Text Retrieval*, Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP, Montreal.1998.

[9] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs NJ, U.S.A., 1988.

Authors Profile

Paste

Mausumi Goswami received the **B.E.** degree in computer science engineering, M.Tech in Information Technology from Tezpur University and also an MBA. Her research interest includes Data mining, Machine Learning, Information Retrieval, Bioinformatics, Natural Language Processing.

Bipul Syam Purkayastha received PhD degree in Mathematics from North Eastern Hill University, Shillong in 1997.Currently he is a Professor in Computer Science Department in Assam University, Silchar. His research interests include Artificial Intelligence, Natural Language Processing.

Networks.