Frequent Itemset Based Hierarchical Document Clustering Using Wikipedia as External Knowledge

Kiran G V R, Ravi Shankar, and Vikram Pudi

International Institute of Information Technology, Hyderabad
{kiran_gvr,krs_reddy}@students.iiit.ac.in, vikram@iiit.ac.in

Abstract. High dimensionality is a major challenge in document clustering. Some of the recent algorithms address this problem by using frequent itemsets for clustering. But, most of these algorithms neglect the semantic relationship between the words. On the other hand there are algorithms that take care of the semantic relations between the words by making use of external knowledge contained in WordNet, Mesh, Wikipedia, etc but do not handle the high dimensionality. In this paper we present an efficient solution that addresses both these problems. We propose a hierarchical clustering algorithm using closed frequent itemsets that use Wikipedia as an external knowledge to enhance the document representation. We evaluate our methods based on F-Score on standard datasets and show our results to be better than existing approaches.

Keywords: Frequent itemsets, Document clustering, Wikipedia, Ontology, TF-IDF.

1 Introduction

A recent trend in clustering documents is the use of frequent itemsets. These methods handle the high dimensionality of the data by considering only the words which are frequent for clustering. A frequent itemset is a set of words which occur together frequently and are good candidates for clusters. Many algorithms in this category consider the entire set of frequent itemsets for clustering, which may lead to redundant clusters.

Most approaches performing document clustering do not consider the semantic relationship between the words. Thus if two documents talking about the same topic do that using different words (which may be synonyms), these algorithms can not find the similarity between them and may cluster them into two different clusters. A simple solution to this problem is to use an ontology to enhance the document representation.

Topic detection is a problem very closely related to that of document clustering. Intuitively, the goal here is to determine the set of all topics that are contained in a document. A topic is not necessarily the same as a keyword, but can be defined by a set of related keywords. The problem of topic detection therefore reduces to finding sets of related keywords in a document collection.

R. Setchi et al. (Eds.): KES 2010, Part II, LNAI 6277, pp. 11–20, 2010.

The contributions of this paper are: (1) First, we show a strong formal connection between the dual problems of document clustering and topic detection when seen in the context of frequent itemset mining. (2) Second, we propose an efficient document clustering algorithm based on frequent itemset mining concepts that has the following attractive features: (a) It handles high dimensional data – up to thousands of dimensions. (b) It allows the use of external knowledge such as that contained in Wikipedia to handle semantic relations between words. (c) Achieves a compact clustering using concepts from generalized frequent itemsets [1]. (d) It provides meaningful labels to the clusters. While there are several document clustering algorithms that also have some of the above features, ours is unique in that it combines all of them.

The rest of the paper is organized as follows: Section 2 describes the itemset based document clustering problem. Section 3 talks about the related work. Section 4 briefly describes our entire algorithm. Section 5 details our approaches to reduce document duplication and Section 6 presents the experiments that support our approaches. Finally we conclude our paper in Section 7.

2 Itemset-Based Document Clustering Problem

In this section, we formally describe the problem of itemset-based clustering of documents. Our formulation captures the essence of related methods (described in Section 3) in an elegant manner. In particular, we show that the dual problems of document clustering and topic detection are related very closely when seen in the context of frequent itemset mining.

Intuitively, the document clustering problem is to cluster text documents by using the idea that *similar documents share many common keywords*. Alternatively, the topic detection problem is to group related keywords together into meaningful topics using the idea that *similar keywords are present in the same documents*. Both of these problems are naturally solved by utilizing frequent itemset mining as follows.

For the first problem, keywords in documents are treated as *items* and the documents (treated as sets of keywords) are analogous to transactions in a market-basket dataset. This forms a transaction space, that we refer to as **doc-space** as illustrated below, where the d_i's are documents and w_{ij}'s are keywords.

d_1 – $[w_{11}, w_{21}, w_{31}, w_{41}, \ldots]$
d_2 – $[w_{12}, w_{22}, w_{32}, w_{42}, \ldots]$
d_3 – $[w_{13}, w_{23}, w_{33}, w_{43}, \ldots]$
...

Then, in this doc-space, frequent combinations of keywords (i.e., frequent itemsets) that are common to a group of documents convey that those documents are similar to each other and thereby help in defining clusters. e.g: if (a, b, c) is a frequent itemset of keywords, then (d_1, d_3, d_4) which are the documents that contain these keywords form a cluster.

For the second problem, documents themselves are treated as *items* and the keywords are analogous to transactions – the set of documents that contain a

keyword is the transaction for that keyword. This forms a transaction space, that we refer to as **topic space** as illustrated below, where the d_i's are documents and w_{ij}'s are keywords.

w_{11} – $[d_{i1}, d_{j1}, d_{k1}, \ldots]$
w_{12} – $[d_{i2}, d_{j2}, d_{k2}, \ldots]$
w_{13} – $[d_{i3}, d_{j3}, d_{k3}, \ldots]$
...

Then, in this topic-space frequent combinations of documents (i.e., frequent itemsets) that are common to a group of keywords convey that those keywords are similar to each other and thereby help in defining topics.

In this context, the following lemma formally captures the relationship between the doc-space and topic-space representations.

Lemma 1. *If $(w_1, w_2, w_3, \ldots)$ is frequent in the doc space, $(d_i, d_j, d_k, \ldots)$, the documents containing these words will also be frequent in the topic space and vice versa.*

Proof: Let $W = (w_1, w_2, \ldots, w_n)$ be frequent in the doc space and $D = \{d_1, d_2, \ldots, d_m\}$ be the corresponding set of documents for these frequent words.

$\Longrightarrow$ In the topic space, $(d_1, d_2, d_2, \ldots, d_m)$ occurs in each transaction $w_i \in W$.

$\Longrightarrow$ Each $d_i \in D$ occurs in atleast n transactions.

$\Longrightarrow$ This means D is frequent in the topic space for all minimum supports $\leq n$, where n is the length of W. □

The above analogies between document clustering, topic detection and frequent itemset mining are summarized in Table 1.

Table 1. Document Clustering and Topic Detection

Frequent Itemset Mining	Document Clustering	Topic Detection
Item	Keyword	Document
Transaction	Document	Keyword
Frequent Itemset	Document Cluster	Topic

3 Related Work

Document clustering has been an interesting topic of study since a long time. Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [2] is one of the best algorithms for agglomerative clustering [3]. K-Means and its family of algorithms have also been extensively used in document clustering. Bisecting K-means is the best one in this family for partitional clustering [3]. Clustering using frequent itemsets has been a topic of extensive research in recent times. Frequent Term based Clustering (HFTC) [4] has been the first algorithm in this regard. But HFTC was not scalable and Fung, et al came up with Hierarchical Document Clustering using Frequent itemsets (FIHC) [5] which outperforms

HFTC. It provides a hierarchical clustering with labels to the clusters. Some of the drawbacks of FIHC include (i)using all the frequent itemsets to get the clustering (number of frequent itemsets may be very large and redundant) (ii) Not comparable with previous methods like UPGMA and Bisecting K-means in terms of clustering quality. [6] (iii) Use hard clustering (each document can belong to at most one cluster), etc. Then Yu, et al came up with a much more efficient algorithm using closed frequent itemsets for clustering(TDC) [7]. They also provide a method for estimating the support correctly. But they use closed itemsets which also may be redundant.

Recently Hasan H Malik, et al. proposed Hierarchical Clustering using Closed Interesting Itemsets, (which we refer to as HCCI) [6] which is the current state of the art in clustering using frequent itemsets. They further reduce the closed itemsets produced by using only the closed itemsets which are "interesting". They use Mutual Information, Added Value, Chi Square, etc as interestingness measures of the closed itemsets. They show that this provides significant dimensionality reduction over closed itemsets and as a result an increase in cluster quality and performance. The major drawback of using the various interestingness measures provided by them is that there might be a loss of information when we reduce the number of closed itemsets. We discuss the drawbacks in the score functions used by them in Section 5.1 and propose improvements that overcome these drawbacks.

Research is also being done about improving the clustering quality by using an ontology to enhance the document representation. Some of the most commonly available ontologies include WordNet, MESH, etc. Several works [8,9,10] have been done to include these ontologies to enhance document representation by replacing the words with their synonyms or the concepts related to them. But all these methods have a very limited coverage. It can also happen that addition of new words could bring in noise into the document or while replacing the original content, there might be some information loss. Existing knowledge repositories like Wikipedia and ODP(open directory project) can be used as background knowledge. Gabrilovich and Markovitch [11,12] propose a method to improve text classification performance by enriching document representation with Wikipedia concepts. Further, [13,14] present a framework for using Wikipedia concepts and categories in document clustering.

Out of the existing knowledge repositories, we chose wikipedia because it captures a wide range of domains, is frequently updated, is ontologically well structured, and is less noisy. [11,12]

4 Our Approach : Overview

In our approach, we first apply any simple frequent itemset mining algorithm like Apriori on all the documents to mine the frequent itemsets. We used the approach given in TDC [7] to calculate the best support. Since words which occur frequently need not necessarily mean they are important, we use the concept of generalized closed frequent itemsets [1] to filter out the redundant frequent

itemsets. Even though the number of generalized closed frequent itemsets are very less compared to the number of frequent itemsets, we guarantee that there is very little loss of information in our method.

Using the generalized closed frequent itemsets obtained, we construct the initial clusters (as mentioned in Section 2). These initial clusters have a lot of overlap between them. We use two approaches to reduce the overlapping and get the final clusters. The first approach, proposed in TDC uses tf-idf scores of the words to give scores to each document in a cluster. This can help us in deciding which cluster the document best belongs to.

The second method uses Wikipedia as an ontology to enrich the document representation. We take each document and enrich its content using the outlinks and categories from Wikipedia. Then we calculate the score of each document in the initial clusters. This score is used to find out the best clusters to which a document belongs. We then introduce a new method to generate labels to clusters using Wikipedia categories which we find to be more interesting than the existing methods which use frequent itemsets as labels to clusters. Our experiments show that the approach using Wikipedia gives better clustering results than the TF-IDF approach because of the use of external knowledge. We also show that our clustering quality is better than most of the existing approaches (Section 6).

Typically frequent itemset mining produces too many frequent itemsets. This is a problem because frequent itemsets correspond to the cluster candidates and the number of clusters need to be small in number. To alleviate this problem, we use Generalized Closed frequent itemsets [1] which are much fewer in number. The approach guarantees that the actual frequent itemsets with their supports can be calculated back within a deterministic factor range.

5 Reducing Document Duplication

The initial clusters have a lot of overlaps between themselves leading to a single document being present in multiple clusters. In order to reduce document duplication we propose two different approaches as described in the next section. In this section, we propose two methods for reducing the document duplication from the initial clusters. Section 5.1 covers the TF-IDF approach and Section 5.2 explains the Wikipedia approach. The basic idea of clustering is to produce disjoint groups of objects/entities. But in the case of document clustering, this idea of disjointness doesn't hold in all cases. This is because a document may belong to multiple clusters, e.g. a document about Arnold Schwarzenegger, may belong to clusters like actors, politics, movies, etc. But allowing a document to be present in too many clusters might lead to redundant clusters. So, we put a limit on the number of clusters a document can belong to using the score functions proposed below.

5.1 TF-IDF Approach

We used a document duplication removal and score calculation method similar to TDC [7], which uses the documents TF-IDF vector (using frequent 1-itemsets

only) and adds the term frequencies of individual items that exist in the itemset. We incorporated a few major changes which help in improving the performance. TDC limits document duplication to a maximum of *max_dup* clusters(the maximum number of clusters in which a document can occur), a used defined parameter. A major drawback of this approach is that they are using the same *max_dup* for all the documents. But in a general scenario, all documents need not be equally important. So, instead we use a threshold which is the percentage of clusters in which a document can be present. In our experiments, we observed that setting the value of *max_dup* between 2-4% gives us good results.

For calculating the score TDC and HCCI [6] use the following approach:

$$Score(d, T) = \sum_{t \in T} (d \times t) \tag{1}$$

where $d \times t$ denotes the tf-idf score of each t in T, the frequent itemset in document d.
e.g: if (a, b, c) is a frequent itemset and (d_1, d_3, d_4) is the corresponding document set, score of d_1 in (a, b, c) is the sum of tf-idf scores of a, b, c in d_1 respectively. The top *max_dup* documents having the highest scores are put into their respective clusters by this score.

But this method has a drawback that longer frequent itemsets would get a higher score. We can solve this problem by using Eq. 2 instead.

$$Score(d, T) = \sum_{t \in T} (d \times t) \,/\, length(T) \tag{2}$$

5.2 Using Wikipedia

Definition 1 (Wikipedia Categories). *Wikipedia contains preclassified topic labels to each document. Each document of Wikipedia belongs to at least one such topic. These can be accessed at the bottom of any Wikipedia page. e.g:* $(Federer)^{category} = \{$*World No. 1 tennis players, 21st-century male tennis players, Wimbledon Champions, ...*$\}$*, etc.*

Definition 2 (Wikipedia Outlinks). *We define outlinks of Wikipedia to be the words which have an internal hyperlink to a different Wikipedia document. e.g: A page on Federer in Wikipedia has outlinks like* $(Federer)^{outlinks} = \{$*tennis, world no. 1, Switzerland, Grand slam, ...*$\}$*, etc. Outlinks indicate the major topics which are present in a document.*

5.2.1 Mapping Documents to Wikipedia Outlinks and Categories

The mapping process is divided into four steps:

1. Extracting Wikipedia outlinks and categories from each page of Wikipedia.
2. Building an inverted index separately for categories and outlinks consisting of the categories/outlinks and the list of wikipedia pages.
3. Constructing a multi level index on these inverted index for fast access.

4. Now, we take a dataset (like Re0, Hitec, etc) and for each word in a document of the dataset, we check if a Wikipedia page exists and find the outlinks and categories of that page.

5.2.2 Document Duplication Removal Using Wikipedia

We now have the initial clusters and we have to assign each document to only a max_dup clusters. To do this, we compute the score of each document(d) in an initial cluster(C) as follows:

$$Score(d, C) = \sum_{d_i \in C} sim\,(d, d_i)\,/size(C) \tag{3}$$

where d_i is the document that belongs to a cluster C, where

$$sim(d_i, d_j) = csim(d_i, d_j)^{word} + \alpha * csim(d_i, d_j)^{outlinks} + \beta * csim(d_i, d_j)^{category} \tag{4}$$

and $csim(d_i, d_j)^{word}, csim(d_i, d_j)^{outlinks}, csim(d_i, d_j)^{category}$ indicates the cosine similarity between words, outlinks and categories in the documents d_i, d_j respectively. A similarity measure similar to Eqn. (4) was proposed in [13]. In Eqn. (3), we divide by $size(C_j)$ for the same reason as explained in Eqn. (2). An advantage of using Eqn. (3) is that we compute the similarity of documents unlike in Eqn. (2) where we calculate the similarity of a frequent itemset and a document.

In our experiments we found that setting the values of α and β such that $\alpha > 1$ and $\beta < 1$ yield good clusters because:

(i) α indicates the similarity between the outlinks which are nothing but the words which contain hyperlinks to other pages in that document. So, they need to have a higher weight than ordinary words. In our experiments, we found that setting α between 1.1 and 1.9 gives us good results.

(ii) β indicates the similarity between the categories of Wikipedia which are generalizations of the topics in a document. So, if the categories of two documents match, we cannot be sure as to whether the two documents talk about the same topic as opposed to the case where two words are the same. In our experiments, we found that setting β between 0.5 and 1 gives us good results.

5.3 Generation of Hierarchy and Labeling of Clusters

We generate the hierarchy of clusters using a method similar to the one used in HCCI [6], except that we use generalized closed itemsets instead of closed interesting itemsets. Many previous approaches like FIHC [5], TDC [7], HCCI [6] propose a method of creating labels to clusters. The labeling given to a cluster is the frequent itemset to which the documents in the cluster belong to. We propose a new method for labeling of clusters using Wikipedia categories as labels which we find to be more interesting and efficient. Using frequent itemsets as labels to clusters is a good approach but it has the problem that there can be frequent itemsets of length > 30, some words being not so important, etc. In

the approach we propose, we take the Wikipedia categories of all the documents of a cluster and find out the top k frequently occurring categories and assign them as the labels to the cluster. We find this method more interesting because categories in Wikipedia represent a more general sense of the documents in the cluster (like the parent nodes in a hierarchy).

6 Experimental Evaluation

We performed extensive experiments on 5 standard datasets of varying characteristics like number of documents, number of classes, etc. on both our approaches. The datasets we used and their properties are explained in Table 2. We also compared our approach to the existing state of the art approaches like UPGMA [2], Bisecting K means [3], Closed itemset clustering methods like TDC [7] and HCCI [6]. We used F-score measure to compute the quality of clustering. The quality of our clustering solution was determined by analyzing how documents of different classes are distributed in the nodes of the hierarchical tree produced by our algorithm on various datasets. A perfect clustering solution will be one in which every class has a corresponding cluster containing exactly the same documents in the resulting hierarchical tree.

The results of our approaches have been discussed in Section 6.2 and 6.3. Our results are better than the existing algorithms because of the following reasons: (i) We use Generalized closed frequent itemsets which guarantee minimum loss of information, at the same time, reduce the number of closed itemsets; (ii) we propose improvements to the score functions provided by the existing algorithms and also propose new score functions; (iii) we make use of knowledge contained in Wikipedia to enhance the clustering.

Table 2. Datasets

Dataset	no. of classes	no. of attributes	no. of docs	Source
Hitech	6	13170	2301	San Jose Mercury news
Wap	20	8460	1560	WebACE
Re0	13	2886	1504	Reuters-21578
Ohscal	10	11162	11465	Ohsumed-233445
K1a	6	2340	13879	WebACE

6.1 Datasets

Wikipedia releases periodic dumps of its data[1]. We used the latest dump consisting of 2 million documents having a total size of 20GB. The data was present in XML format. We processed the data and extracted the categories and outlinks out of them. Datasets for clustering were obtained from Cluto clustering toolkit [15]. The results of various algorithms like UPGMA, Bisecting K means,

[1] http://download.wikipedia.org

FIHC [5], etc were taken from the results reported in HCCI [6]. HCCI [6] presents different approaches for clustering, we have taken the best scores out of all of them. We used CHARM toolkit for mining closed frequent itemsets.

6.2 Evaluation of TF-IDF Approach

Our TF-IDF approach stated in Section 5.1 reduces the document duplication and produces the final clusters. In this approach, we propose improvements to the existing methods used in TDC and HCCI. We can see from the results in Table 3 that our approach performs better than TDC for all the datasets. Compared to HCCI, our approach performs better for 3 datasets Hitech, Wap, Ohscal, and comparitively for the remaining datasets.

Table 3. Comparision of F-Score using our approaches

Dataset	UPGMA	Bisecting k-means	FIHC	TDC	HCCI	Tf-idf	Wikipedia
Hitech	0.499	0.561	0.458	0.57	0.559	0.578	**0.591**
Wap	0.640	0.638	0.391	0.47	0.663	0.665	**0.681**
Re0	0.548	0.59	0.529	0.57	**0.701**	0.645	0.594
Ohscal	0.399	0.493	0.325	N/A	0.547	0.583	**0.626**
K1a	0.646	0.634	0.398	N/A	0.654	0.626	**0.672**

6.3 Evaluation of Wikipedia Approach

The Wikipedia approach proposed in Section 5.2 uses background knowledge from Wikipedia to enhance the document representation. We expect better results(than the TF-IDF approach) for this approach because of use of an ontology. The results in Table 3 illustrate that our approach performs better than all other existing approaches for 4 datasets (Hitech, Wap, Ohscal, K1a). The drawback in our approach is that it might not be of great use for datasets which do not have sufficient coverage in Wikipedia i.e. the documents in this dataset do not have corresponding pages in Wikipedia(like Re0).

7 Conclusions and Future Work

In this paper, we presented a hierarchical algorithm to cluster documents using frequent itemsets and Wikipedia as background knowledge. We used generalized closed frequent itemsets to construct the initial clusters. Then we proposed two methods using, 1) TF-IDF 2) Wikipedia as external knowledge, to remove the document duplication and construct the final clusters. In addition to these, we proposed a method for labeling of clusters. We evaluated our approaches on five standard datasets and found that our results are better than the current state of the art methods. In future, these ideas can be extended to other areas like evolutionary clustering, data streams, etc by using incremental frequent itemsets.

References

1. Pudi, V., Haritsa, J.R.: Generalized Closed Itemsets for Association Rule Mining. In: Proc. of IEEE Conf. on Data Engineering (2003)
2. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. In: An introduction to Cluster Analysis. John Wiley & Sons, Inc., Chichester (1990)
3. Zhao, Y., Karypis, G.: Evaluation of Hierarchial Clustering Algorithms for Document Datasets. In: Proc. of Intl. Conf. on Information and Knowledge Management (2002)
4. Beil, F., Ester, M., Xu, X.: Frequent Term-based Text Clustering. In: Proc. of Intl. Conf. on Knowledge Discovery and Data Mining (2002)
5. Fung, B., Wang, K., Ester, M.: Hierarchical Document Clustering using Frequent Itemsets. In: Proc. of SIAM Intl. Conf. on Data Mining (2003)
6. Malik, H.H., Kender, J.R.: High Quality, Efficient Hierarchical Document Clustering Using Closed Interesting Itemsets. In: Proc. of IEEE Intl. Conf. on Data Mining (2006)
7. Yu, H., Searsmith, D., Li, X., Han, J.: Scalable Construction of Topic Directory with Nonparametric Closed Termset Mining. In: Proc. of Fourth IEEE Intl. Conf. on Data Mining (2004)
8. Hotho, A., Staab, S., et al.: Wordnet Improves Text Document Clustering. In: The 26th Annual Intl. ACM SIGIR Conf. on Proc. of Semantic Web Workshop (2003)
9. Hotho, A., Maedche, A., Staab, S.: Text Clustering Based on Good Aggregations. In: Proc. of IEEE Intl. Conf. on Data Mining (2001)
10. Zhang, X., Jing, L., Hu, X., et al.: A Comparative Study of Ontology Based Term Similarity Measures on Document Clustering. In: Proc. of 12th Intl. Conf. on Database Systems for Advanced Applications (2007)
11. Gabrilovich, E., Markovitch, S.: Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In: Proc. of The 21st National Conf. on Artificial Intelligence (2006)
12. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In: Proc. of The 20th Intl. Joint Conf. on Artificial Intelligence (2007)
13. Hu, X., Zhang, X., Lu, C., et al.: Exploiting Wikipedia as External Knowledge for Document Clustering. In: Proc. of Knowledge Discovery and Data Mining (2009)
14. Hu, J., Fang, L., Cao, Y., et al.: Enhancing Text Clustering by Leveraging Wikipedia Semantics. In: Proc. of 31st Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (2008)
15. Cluto: `http://glaros.dtc.umn.edu/gkhome/views/cluto`