A Frequent Term-Based Multiple Clustering Approach for Text Documents

Hai-Tao Zheng, Hao Chen, and Shu-Qin Gong

Tsinghua-Southampton Web Science Laboratory
Graduate School at Shenzhen, Tsinghua University Shenzhen, China
zheng.haitao@sz.tsinghua.edu.cn,
{jerrychen1990,gongshuqin90}@gmail.com

Abstract. With the boom of web and social network, the amount of generated text data has increased enormously. On one hand, although text clustering methods are applicable to classify text data and facilitate data mining work such as information retrieval and recommendation, inadequate aspects are still evident. Especially, most existing text clustering methods provide either a hard partitioned or a hierarchical result, which cannot describe the data from various perspectives. On the other hand, multiple clustering approaches, which are proposed to classify data with various perspectives, meet several challenges such as high time complexity and incomprehensible results while applied to text documents. In this paper, we propose a frequent term-based multiple clustering approach for text documents. Our approach classifies text documents with various perspectives and provides a semantic explanation for each cluster. Through a series of experiments, we prove that our method is more scalable and provides more comprehensible results than traditional multiple clustering methods such as OSCLU and ASCLU while applied to text documents. In addition, we also found that our approach achieves a better clustering quality than existing text clustering approaches like FTC.

Keywords: Multiple clustering, Frequent term, Text documents.

1 Introduction

Data mining in database provides data owners with new information and patterns in their data. Clustering is a traditional data mining task for automatically grouping data. However, groups may be hidden in different perspectives of the data. An item may belong to different groups with different perspectives. Traditional clustering approaches only provide either a hard partitioned result or a hierarchical result. In these clustering results, an item can only belong to one group. To discover hidden groups in various perspectives, we need to apply multiple clustering approaches. Multiple clustering methods can assign one item to different groups with respect to different perspectives. Generally speaking, multiple clustering approaches have to deal with two challenges including high time complexity and redundant result.

L. Chen et al. (Eds.): APWeb 2014, LNCS 8709, pp. 602–609, 2014.

On the other hand, as the web continues to grow rapidly, huge number of text documents have been generated. To organize and do data mining work on these text documents, text clustering becomes a very important application of clustering algorithms. However, compared to other applications of clustering, three major challenges including high dimensionality, large data and incomprehensible results should be addressed for text clustering:

Although applying a multiple clustering approach to text documents can help us significantly while doing text mining tasks, to our best knowledge there is no existing feasible multiple clustering approach for text documents since now. The high dimensionality of text documents makes multiple clustering approaches not scalable while applied to text documents. In this paper, we propose the first feasible multiple clustering approach for text documents called FTMTC(frequent term-based multiple text clustering approach). FTMTC represents a cluster with a set of terms to deal with the high dimensionality challenge. We also introduce WordNet[1] to improve the quality of redundancy removal process.

This paper is structured as follows: We review existing text clustering and multiple clustering approaches in section 2. In section 3, we introduce a series of notations to define the problem we are going to solve. We describe details of FTMTC with sequence charts in section4. Then, we prove our approach is feasible and outstanding with a series of experiments in section 5. Finally, we make a conclusion and introduce our future work in section 6.

2 Related Work

2.1 Multiple Clustering Approaches

The main difference between multiple clustering approaches and traditional clustering approaches is that multiple clustering's result contains clusters discovered with various perspectives. Clusters in multiple clustering result can overlap to each other while clusters in traditional clustering result can't. Traditional multiple clustering approaches tend to generate a quite large amount of clusters. The result contains a lot of redundant clusters. OSCLU[2] is a recent proposed non-redundant multiple clustering approach, which is based on the idea that a pair of clusters which share more than a certain amount of overlapped dimensions and items should be regarded as similar to each other. ASCLU[3] applies OSCLU to an alternative clustering way.

2.2 Text Clustering Approaches

Most text clustering approaches rely on a *vector-space model*, in which, each text document d is represented by a vector of frequencies of all terms: $d = (tf_1, tf_2, \ldots, tf_m)$. Based on this model, standard clustering approaches like k-means[4] can be applied to text documents directly. But they can't handel the high dimensional and incomprehensible result challenges well.

In this paper, we propose a multiple clustering solution for text documents based on frequent term model. This model can help us get avoid of the high dimensionality challenge of text documents.

3 Problem Definition

For consistent notations in the following sections, we define some notations here. First of all, we make a formal definition of our problem: Given a set of text documents $DS = \{d_1, d_2, \ldots, d_m\}$ as input, let $T_{all} = \{t_1, t_2, \ldots, t_k\}$ denote all the terms that appear in DS and $T(d)$ denote terms that appear in d. Our target is to generate a set of clusters $R = \{C_1, C_2, \ldots, C_n\}$. In this procedure, three main challenges need to be addressed:

Challenge 1: Incomprehensible Results: Traditional clustering results do not provide explanations for clusters. To give each cluster a explanation, we associate each cluster in R with a term set. To associate terms with documents, we introduce the following definitions: As document d contains a set of terms, a term t can also "cover" a set of documents. We define the set of documents in DS that contain term t as $Cover(t)$:

$$Cover(t) = \{d \in DS | t \in T(d)\} \tag{1}$$

The "cover" of a term set $T = \{t_1, t_2, \ldots, t_k\}$ is defined as the intersection of all terms in T:

$$Cover(T) = \bigcap_{i=1}^{k} Cover(t_i) \tag{2}$$

So, if $Cover(T)$ is the documents grouped by a cluster, T will give an explanation for the cluster.

Challenge 2 High Dimensional Data: To deal with the high dimensional challenge, we control the number of term sets that are associated to clusters. We only associate frequent term sets to clusters. We can judge whether a term set T is a frequent set with $Cover(T)$, we define the set of all frequent term sets as $FTS(DS)$:

$$FTS(DS) = \{T \subseteq T_{all} \big| |Cover(T)| \geq \alpha * |DS|\} \tag{3}$$

Where α is the threshold of frequent term set. So, a cluster is composed with a frequent term set T as explanation and a document set D as members.

$$C = (T, D) \tag{4}$$

Where $T \in FTS(DS)$ and $D = Cover(T)$.

Challenge 3 Redundant Clustering Results: To prevent a redundant clustering result, the size of R should be reasonable. Each cluster in R should bring novel information. We will introduce a cluster picking algorithm in section 4 to handle this challenge.

4 Frequent-Term Based Multiple Text Clustering Approach

Based on the notations above, we propose a multiple clustering approach for text documents called FTMTC(Frequent-term based multiple text clustering). Generally speaking, FTMTC is composed of three steps as shown in Fig. 1:

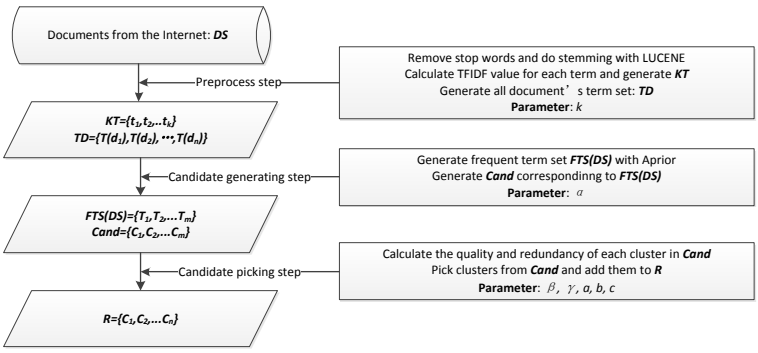

Fig. 1. Sequence diagram of FTMTC

4.1 Preprocess Step

To preprocess the document data, we conduct several steps including stop words removing, stemming and indexing. First of all, a stop word list[1] is employed to remove the stop words. Secondly, we apply Poter stemming algorithm for word stemming. To process the document efficiently, we apply a tool named Lucene to build index files for the documents.

For an efficient algorithm, we extract k important words from T_{all} as key terms. The key term set is noted as KT. Since nouns with high TFIDF value tend to be representative in general, we pick nouns with high TFIDF from T_{all} and add them to KT.

4.2 Candidate Generating Step

In this step, we generate $FTS(DS)$ with Aprior algorithm[5] algorithm. For each T in $FTS(DS)$, we build a corresponding cluster $C = (T, Cover(T))$ and add it to the candidate set $Cand$. Since term sets with more terms tend to cover less documents than those with less terms, we set the document coverage threshold as follows: Assuming that α is the threshold of frequent term set with one term, the threshold of a frequent term set with N terms will be $\alpha \times 0.9^{(N-1)}$

4.3 Candidate Picking Step

In this step, we pick clusters from $Cand$ and add them to the result set R gradually. First of all, we rank clusters in $Cand$ based on clusters' quality. Generally speaking, clusters with large number of documents or terms tend to have high quality. Besides, if the terms are closely related to the documents, the cluster's quality is high. We judge the relationship between terms and documents with average TFIDF value. Therefore, we define the the quality of a cluster

[1] `http://www.ranks.nl/stopwords`

$C(T, D)$ as $Quality(C) = |D|^a \times |T|^b \times AVG_{TFIDF}(C)^c$. where $a + b + c = 1$ and $AVG_{TFIDF}(C)$ denotes the average TFIDF value between documents and terms.

$$AVG_{TFIDF}(C) = \frac{1}{m \times n} * \sum_{i=1}^{m} \sum_{j=1}^{n} TFIDF(t_j, d_i) \tag{5}$$

Where m denotes the size of D and n denotes the size of T

As clusters in $Cand$ are sorted by $Quality(C)$ in descending order, we remove redundant clusters from $Cand$ to deal with challenge 3. Inspired by OSCLU, we consider clusters either have dissimilar term sets or group dissimilar documents to be non-redundant to each other.

Obviously we can define the similarity between two term sets with overlap percentage. However, terms contain semantic meanings. It makes similarity between term sets more complex than similarity between mathematic vectors. For example, term set {"USA", "president", "history"} and term set {"America", "chairman", "past"} share no term, but they do represent similar concepts.

To adapt the redundancy definition to text clustering, we introduce WordNet as external knowledge. WordNet is a lexical database which can be used to calculate the similarity between two terms. We use Jiang and Conrath's word similarity algorithm JNC[6] to judge the similarity between terms. We define semantic similarity between two term sets $T = \{t_1, t_2, \ldots, t_n\}$ and $\hat{T} = \{\hat{t_1}, \hat{t_2}, \ldots, \hat{t_m}\}$ as $Similarity(T, \hat{T})$:

$$Similarity(T, \hat{T}) = \frac{1}{2n} \times \sum_{i=1}^{n} \max_{\hat{t_j} \in \hat{T}} JNC(t_i, \hat{t_j}) + \frac{1}{2m} \times \sum_{i=1}^{m} \max_{t_j \in T} JNC(\hat{t_i}, t_j) \tag{6}$$

With a similarity threshold β, we can get a group of clusters in R that are similar to a given cluster C. The similar group of $C(T, D)$ in R with threshold β is defined as:

$$SimGroup_\beta(C, R) = \{C_i \in R \backslash C | Similarity(T, T_i) \geq \beta\} \tag{7}$$

Although C has similar term set with clusters in $SimGroup_\beta(C, R)$, if C groups dissimilar documents, we also consider C as non-redundant to clusters in R. Given a cluster set $CS = \{C_1, C_2, \ldots, C_n\}$, we define the coverage of CS's document as $Coverage(CS) = \bigcup_{i=1}^{n} D_i$, where D_i denotes the document set of C_i

At last, we define an interest value $Interest(C, R)$ to judge whether C is novel to R. Given a threshold γ, we add C to R if $Interest(C, R)$ is larger than γ. Since we already sort $Cand$ with regard to cluster's quality by descending order, it's obvious that our algorithm is a greedy algorithm and thus can maximize the summation of clusters' qualities under the premise that R is none-redundant. We define the interest of $C = (T, D)$ to R as $Interest(C, R)$:

$$Interest(C, R) = \frac{|D \backslash Coverage(SimGroup_\beta(C, R))|}{|D|} \tag{8}$$

5 Experiments

5.1 Experiment Setup

To build a multi-label data set, we download 4505 biography pages from Wikipedia with two different perspectives. The biography pages are downloaded from four country categories and three occupation categories.

We measure clustering results with three aspects. First of all, we list the clustering result to prove it covers categories with different perspectives in section 5.2. Secondly, we evaluate the scalability of our algorithm in section 5.3. At last, in section 5.4, we evaluate the quality of clustering result with multiple clustering evaluation measurements introduced in [7], including purity, entropy and F1-value.

5.2 Experiment Result

Table 1 shows the result of FTMTC. It handles Challenge 1 and Challenge 2 well. The term set associated to a cluster explains the cluster's topic well. We mark categories in nationality perspective with black font and mark categories in occupation perspective with normal font. We found the result covers every known category in two perspectives. Besides, we are glad to see that FTMTC also can discover clusters we do not know in advance like"War" and detailed category like "Swim". We mark them with italics font.

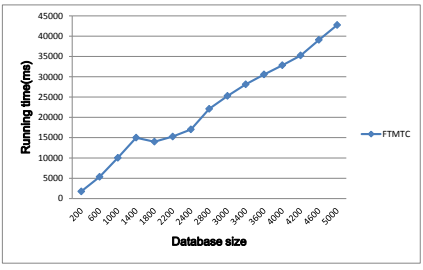

Fig. 2. FTMTC's scalability

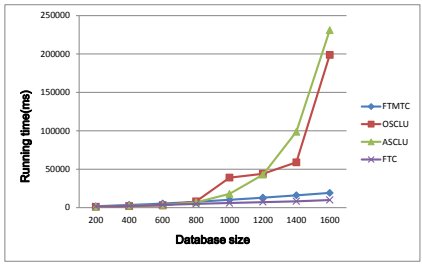

Fig. 3. running time Vs size

5.3 Scalability Evaluation

Multiple clustering approaches that are based on term vector model's running time grows fast as the database grows. Since FTMTC applies similar redundancy removal process with OSCLU and ASCLU, we compare FTMTC with OSCLU, ASCLU and FTC. We do experiments as the database's size grows from 100 to 5000 (add 200 documents each time). Each time, we run the clustering algorithms ten times and calculate the average running time. From Fig.2 and Fig.3, we can see that FTMTC and FTC's running time grows linearly as the database's size grows while OSCLU and ASCLU's running time grows exponentially. It's obvious that FTMTC outperforms OSCLU and ASCLU with regard to scalability.

Table 1. Clustering result

ID	Term list	Size	Category
1	[aquatics,champion,Europe,olympics,swim]	169	Athlete, *Swim*
2	[China,football,league]	298	**China**, Athlete, *Football*
3	[football,league,France,nation,team]	217	**France**, Athlete, *Football*
4	[Olympics,swim]	331	Athlete, *Swim*
5	[France,Paris]	741	**France**
6	[cup,football,Germany,nation,team]	149	**Germany**, Athlete, *Football*
7	[China]	858	**China**
8	[news,publisher]	875	Writer
9	[basketball,champion,coach,season]	129	Athlete, *Basketball*
10	[Germany]	916	**Germany**
11	[writer,Europe]	992	Writer
12	[America,gold,summer]	468	**USA**, Athlete
13	[election,party,state]	153	Politician
14	[book,publisher]	329	Writer
15	[California]	329	**USA**
16	[Shanghai]	258	**China**
17	[man]	879	*Man*
18	[mayor,Paris]	268	**France**, Politician
19	[president]	384	Politician
20	[war]	266	*War*

5.4 Clustering Quality Evaluation

Since there is no existing multiple clustering approaches for text documents, we compare FTMTC with FTC, OSCLU and ASCLU on multi-label text documents. We choose FTC as the baseline because FTMTC shares the same cluster definition with it. We choose multi-label text documents as test set to focus on discovering clusters in various perspectives.

We compare clustering quality with different database sizes. . For a fair comparison, we set the FTMTC's key word number equals to FTC's. From Fig.4 to Fig.6 we can see that FTMTC can obviously outperform FTC with regard to purity, F1 value and entropy.

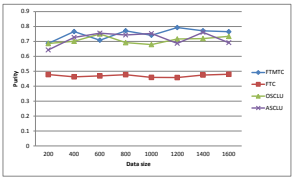

Fig. 4. Purity vs size

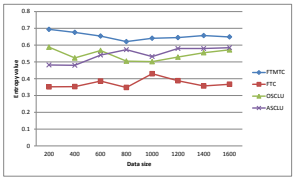

Fig. 5. Entropy vs size

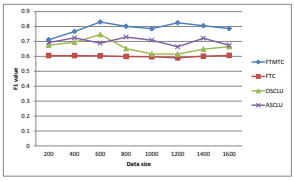

Fig. 6. F1 value vs size

6 Conclusion and Future Work

In this paper, we propose a feasible multiple clustering approach for text documents based on a frequent term model. We also introduce WordNet as external knowledge to help removing results' redundancy. With a series of experiments, we prove that FTMTC can provide an understandable clustering result which contains clusters in various perspectives. FTMTC can also excavate hidden and more detailed clusters, which helps many tasks of data mining. With comparison, we prove that FTMTC is more scalable than traditional multiple clustering approaches and achieves a better clustering result than FTC while applied to multi-label text documents. In the future, we will exploit more external knowledge, such as Cyc Ontology[2] and Wikipedia[3], to improve our clustering results.

Acknowledgments. This research is supported by the 863 project of China (2013AA013300), National Natural Science Foundation of China (Grant No. 61375054) and Tsinghua University Initiative Scientific Research Program(20131089256).

References

1. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
2. Günnemann, S., Müller, E., Färber, I., Seidl, T.: Detection of orthogonal concepts in subspaces of high dimensional data. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1317–1326. ACM (2009)
3. Günnemann, S., Färber, I., Müller, E., Seidl, T.: Asclu: Alternative subspace clustering. In: MultiClust at KDD. Citeseer (2010)
4. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 881–892 (2002)
5. Agrawal, R., Ramakrishnan, Srikant, o.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB, vol. 1215, pp. 487–499 (1994)
6. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008 (1997)
7. Müller, E., Günnemann, S., Assent, I., Seidl, T.: Evaluating clustering in subspace projections of high dimensional data. Proceedings of the VLDB Endowment 2(1), 1270–1281 (2009)

[2] `http://www.cyc.com/`
[3] `https://www.wikipedia.org/`