CDW: A Text Clustering Model for Diverse Versions Discovery*

Rong Xiao, Liang Kong and Yan Zhang, Min Wang
1Department of Machine Intelligence, Peking University, Beijing 100871, China
2Key Laboratory on Machine Perception, Ministry of Education, Beijing 100871, China
xrsmile@gmail.com, kongliang@pku.edu.cn, zhy@cis.pku.edu.cn, wangmin@ucap.com.cn

Abstract—The development of information technology brings numerous online news and events to our daily life. One big problem of such information explosion is, many times there are diverse descriptions for one incident which make people confused. Although previous researches have provided various algorithms to detect and track events, few of them focus on uncovering the diversified versions of an event. In this paper, we propose a novel algorithm which is capable of discovering different versions of one event according to the news reports. We map documents to the topic layer to get the information of each topic. Then we extract the highly-differentiated words of each topic to cluster the documents. Compared with previous work, the accuracy of our algorithm is much higher. Experiments conducted on two data sets show that our algorithm is effective and outperforms various related algorithms, including classical methods such as K-means and LDA.

I. INTRODUCTION

In this age of information, countless amounts of information rush into people’s lives in every minute. A wide variety of media channels offers every one of us an immeasurably large pool of information to browse, in which we know what happens in each corner of the world. Some news reports are inherently objective, like the news of Harry Potter’s premiere, football fans turbulence in Moscow, EU’s call on its members’ mutual assistance in the crisis, etc. Since such kind of news always records or pictures some clear facts, the difference between various versions of reports is little. However, another kind of news may arouse debates since the fact behind the incident is hardly to be completely uncovered. The sinking of Cheonan, a warship of South Korea, is a hotly discussed topic. The media offers us contradictory explanations, including North Korea’s offense, United States’ plot, result of the fighting between South and North, etc. We call this kind of incidents as diverse-version events.

With the help of the techniques of Topic Detection and Tracking (TDT) [1], many websites are now able to provide users new application in organizing news messages. News events are classified into different topics with automatic and immediate update, thus, users are enable to browse all related reports on one specific event very conveniently. By TDT, users can find out how events develop or how situations change through various reportages. However, simple classification and organization are not enough to meet the users’ demand on the information of a diverse versions event. Since we can hardly see readers invest a big effort to distinguish each version of an event from a huge information base by themselves, a novel algorithm which function to find out each version and its content will be very useful.

However, few researches have focused on the discovery of diverse versions as far as we know. Diversified versions discovery is a challenging and valuable research subject. Although simple clustering can serve for this objective, it has a lot of limits. Since news reports about the same event often have high similarities, simple clustering methods cannot separate different “voices” effectively. So, we need to first filter the common description of the subject content, and then find out the different “voices”, which can be represented by the highly-differentiated words. After extracting these highly-differentiated words, we can cluster documents depending on them.

In this paper, we propose a novel algorithm that is capable of discovering different versions of one event. We map documents to different topics to get the information of each topic. Then we extract the highly-differentiated words of each topic to cluster the documents. We aim to develop a text clustering model to solve this problem, and operate by (1) mapping documents to topic layer, extracting topic features and filtering common topics; (2) finding out the representative topics and vectoring documents based on them; (3) finishing the clustering task. In comparison with previous work on diverse versions discovery, the accuracy of our algorithm is higher. We conduct our experiment on two data sets and evaluate the effectiveness of our method in a pairwise judgement task. The results completely demonstrate that our method is effective and outperforms various related algorithms. To sum up, our main contributions include:

- We develop a new clustering model for diverse versions discovery which is based on extracting highly-differentiated words to cluster the documents. Meanwhile, we also introduce an effective method to find out highly-differentiated words in documents collection.
- We build the relationship between documents and topic layer, that is, mapping documents to different topics and then extracting features and clustering on topic layers.

* Supported by NSFC with Grant No. 61073081, National Key Technology R&D Pillar Program in the 11th Five-year Plan of China with Research No. 2009BAH47B05
† Corresponding author
We introduce a method of evaluation for diversified versions discovery and conduct the experiments on real data sets.

The rest of this paper is organized as follows. In Section 2, we revisit the related work. In Section 3, we enumerate the desiderata for an algorithm for detecting multiple meaningful versions. Section 4 introduces our proposed approach in detail. Experiments in Section 5 demonstrate the performance of our approach. Section 6 concludes this paper.

II. RELATED WORK

Diversified versions discovery is based on topic detection and tracking. Topic Detection and Tracking (TDT) is a multi-site research project, to develop core technologies for a news understanding systems [2]. Specifically, TDT systems discover the topical structure in unsegmented streams of news reporting as it appears across multiple media and in different languages [3]. Topic detection and tracking research has been extensively studied in previous work. Pilot experiments in retrospectively and incrementally clustering of text documents have been done as a part of event detection task initiative [4] and query document like retrieval [5]. Topic tracking also stimulates many researches because events are dynamic and evolutionary characteries studied in [6]. Nevertheless, our work is distinct from these previous works. We focus on discovering diversified versions rather than whether events are found.

In our model, we need to employ a method to extract key words in the feature extraction process. Sun et al put forward a method that serves to collect keywords considering the relation between a sentence and its elements [7].

K-means [8] is a classical clustering algorithm. Latent Dirichlet Allocation (LDA) [9] is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. LDA was first presented as a graphical model for topic discovery [10]. [14] introduced a novel word clustering algorithm based on LDA, while our algorithm which clusters based on LDA performs better than it. In [11], a model DVD is presented to produce the discovery of diversified versions of an event based on graph. They firstly use an iterative algorithm on a bipartite graph integrating co-occurrence and semantics to filter the popular words to reduce the tight correlation between documents in a specific event. After that, they employ a communities discovery algorithm to construct version centroid and classifies the documents to multiple versions under Rocchio Classification framework [12].

Our work is related with above mentioned work, however, to the best of our knowledge, our approach has different characteristics from previous work.

III. PRODUCING MULTIPLE VERSIONS

In this Section, we enumerate the desiderata for an algorithm for detecting multiple meaningful versions of an event. By meaningful, we mean that each of these versions should be qualitatively strong. Before we formalize the concept, let us introduce some notation. Let $D = \{d_1, ..., d_n\}$ be a set of $n$ documents about a specific event, where each document $d_i, i = 1 : n$, is represented by a bag of words $w_1, w_2, ..., w_d$. We want to detect $m$ different versions $V = \{v_1, ..., v_m\}$ to describe the event, where each version $v_i, i = 1 : m$, is represented by a word distribution. To discover multiple versions of an event is to detect different aspects about an event and give users different views about an event.

To detect multiple meaningful versions, an algorithm should satisfy the following criteria [13]:

- **Multiplicity**: Given the input of a set of document $D$, our algorithm can detect $m$ (where $m > 1$) different versions, $V_i, i = 1 : m$, without changing the similarity function.

- **Distinctivity**: Each version $V_i, i = 1 : m$, should be distinctively different. We mean that two versions are highly dissimilar.

- **Quality**: Each version $V_i, i = 1 : m$, should be qualitatively strong with respect to the similarity function.

It is easy to prove that our algorithm can satisfy the above criteria on detecting multiple versions of a given event.

IV. OUR APPROACH: CDW

A. System Framework for CDW

The intuitive way to discover diverse versions of an event is to cluster documents. However, the abundant semantic association across documents will be neglected in this way. In this paper, we will try to solve this problem by a two-part approach. First, we find the highly-differentiated characteristics of different versions. Second, we vectorize the documents using these characteristics and then cluster the documents. We call this approach CDW, short for “Clustering by highly-differentiated words”. Specifically, the CDW approach can be broken into the following three steps.

![Fig. 1. The framework of CDW approach](image)
1) Defining characteristic words: Each document can be regarded as a bag of words. Unfiltered words are probably responsible for curse of dimensionality, especially for those large data sets. Our design of algorithm uses word frequency filter and popular words extraction to revise this problem. Thus, we secure the highly-differentiated characteristics by screening the noise, and lay a foundation for the discovery of diverse versions.

2) Constructing processed documents: In this step, we denote all existing documents with the refined characteristics we get in the first step.

3) Text clustering: We can obtain the vectors of the processed documents through the above procedures. Further, we use K-means method to perform a clustering in order to get the final diverse versions.

Now we elaborate the above three steps with following specifications. Section B introduces our method of extracting highly-differentiated characteristics. Since the last two steps are both implemented on topic layer, we put them together in Section C.

B. Detection for Highly-differentiated Characteristics

A word is a most basic unit to construct a document. However, curse of dimensionality is likely to occur if we use all the words included in the documents. So we have to reduce the dimensionality by extracting the highly-differentiated words, the most effective words to differentiate different versions of an event, in the first place.

First, we use a word frequency filter to process the common words. Usually, the words with high frequency of occurrences in the entire data set are very likely to exist repeatedly in most of the documents. This kind of words may involve with the name, subjects, objects, time and location of occurrence of the event, etc. These words contribute little to discriminate different versions due to their prevalence. On the contrary, taking such words into consideration will result in distorted similarity among different versions. To solve this problem, we first count the number of files in which a particular word appears, and define this number as the frequency of the word. Then we determine a threshold value \( d \), and remove the words whose frequency is larger than \( d \).

Second, we utilize LDA to reduce the dimensionality of the set of characteristics and map the documents to different topics. The mapping allows us to collect the features of each cluster by providing information on each topic. Based on LDA, we draw the probability distribution between the topics and each document. According to the maximum membership principle, each document is categorized into the topic which has the highest probability.

Third, we extract the popular words. We now take the popular words in each cluster as the most representative words for their topic. We approach the keywords extraction with a modified version of [7], which can be divided into two steps. First, we sort the sentences depending on their significance value and gather a set of significant sentences. Second, we compute the significance of each word according to the importance of the sentences it belongs to. Thereby an undirected word-graph \( G \) can be constructed.

Definition 1: \( \text{support}(i, j) \) denotes the supportiveness between word \( i \) and word \( j \), and can be defined as

\[
\text{support}(i, j) = \sum_{i,j \in p} \text{SRank}(s_p)
\]

where \( p \) represents the set of words in the sentence \( s_p \) and \( \text{SRank}(s_p) \) represents the importance of \( s_p \). Therefore, the weight of the edge between word \( i \) and word \( j \) in graph \( G \) is set to \( \text{support}(i, j) \).

Applying ranking algorithms such as PageRank [15] to the graph \( G \), we will get the importance value for each word and then select some keywords for each cluster. Finally we merge all the keywords from all of the clusters into a set, which is of highly-differentiated characteristics.

C. Vectorizing and Clustering on Topic Layer

Following the acquisition of the set of highly-differentiated characteristics, we vectorize the documents to construct processed documents. The basic idea in vectorization is to calculate the TF-IDF [16] value of each word so as to formalize the characteristic. Then, a clustering method, K-means, is performed on the vector space.

Figure 2 shows the mapping process between documents and topics, as well as the vectorizing and clustering process on topic layer. From Fig. 2 we can see that, the model of CDW is organized as follows: (1) mapping documents to topic layer, extracting topic features and filtering common topics; (2) finding out the representative topics and vectoring documents based on them; (3) finishing the clustering task.

V. EXPERIMENTS

A. Data Sets

We employ two data sets to show the effectiveness of our CDW approach. One is about the “Cheonan sinking” (including 533 documents, called as CS), and the other is about
“Sean Lien Shot” (including 391 documents, called as LSW). “Cheonan sinking” is the sinking of the South Korean warship Cheonan and hence causes a bitter controversy about who sank it. “Sean Lien Shot” discusses the truth of Sean Lien Shot and the motivation of the murderer. We crawl documents from the mainstream news web portals, such as Reuters, BBC, MSNBC, NYTimes, to build the data set. In the following sub-sections, we firstly propose the evaluation of event diversified version discovery, and then we describe the experimental designs and discuss the results.

B. Evaluation

Since there is no existing standard evaluation for diversified versions discovery, we give our own evaluation metric. The attempt to manually observe all the documents and classify them into different versions is almost impossible. Hence, we use a pairwise judgement task to evaluate the effectiveness.

A pairwise judgement task determines whether a pair of documents is of the same version. First we construct the pairwise standard test sets. We randomly sample 200 pairs of documents for CS and 150 pairs of documents for LSW. Afterwards, a group of volunteers will observe each and every pair of documents and determine if the pair of documents belongs to the same version by voting. If the voting results is ambiguous, the pair of documents will be discarded and a new pair will be added. Finally, we can obtain the pairwise test set, denoted as $T_c = \{<< d_{i1}, d_{i2} >> , v_i = 1 \}$, $d_{i1} \in D^c, d_{i2} \in D^c$, where $v_i = 1$ indicates that $d_{i1}$ and $d_{i2}$ belong to the same version and $v_i = 0$ indicates otherwise situation. Specifically, the pairwise test sets of CS and LSW are constructed and abbreviated to $T_{cs}$ and $T_{lsw}$ respectively.

In our evaluation, the Precision criterion is used to fulfill the pairwise test task. For pairwise documents $< d_{i1}, d_{i2} >$, a judge $v'_i$ is given by each related algorithms. Therefore, we can define Precision in the pairwise test task as $P_{score}$, that is:

$$P_{score} = \frac{\sum_{<d_{i1}, d_{i2} > \in T_c} \bigoplus v_i \bigoplus v'_i}{|T_c|}$$

where $|T_c|$ represents the size of the pairwise test set for event $\varepsilon$. $\bigoplus$ is XNOR.

C. Parameter Tuning

In order to set application parameters, we start our experiment from examining their influence. There are three application parameters.

Firstly, we examine the influence of factor $\alpha$ and $\beta$ under specific $K$. $\alpha$ presents the percentage of words with high frequency. We vary $\alpha$ from 0% to 10% with the step of 1%. $\beta$ presents the percentage of selected popular words. We vary $\beta$ from 10% to 20% with the step of 1%. We check the $P_{score}$ when these two parameters change in Fig.3 and get a best $\alpha$ and $\beta$ value pair ($\alpha_{best}$ and $\beta_{best}$). The best $P_{score}$ is achieved when $\alpha = 4\%$ and $\beta = 15\%$ in CS and $\alpha = 3\%$ and $\beta = 13\%$ in LSW. Furthermore, based on the process of tuning, the $P_{score}$ is decreasing with $\alpha$ when $\alpha$ is more than 10%. As for $\beta$, the $P_{score}$ is increasing left to the interval and decreasing right to the interval.

Fig. 3. $\alpha, \beta$ tuning under specific $K$ in CS and LSW

The parameter $K$ tuning procedure is listed in Table 1. $K$ is the number of clusters in LDA and K-means, which is also the number of versions of the final results. We first forecast a interval of $K$ by browsing some news report about the specific event. Then we check the $P_{score}$ to get a best $K$. Table 1 illustrates that both high and low $K$ significantly reduce the $P_{score}$. So, the best $P_{score}$ is achieved when $K = 5$ in CS and $K = 4$ in LSW.

D. Performance and Discussion

In this section, we study quantitatively the effectiveness of the proposed CDW. Comparisons against related algorithms are also conducted. The related algorithms studied include:
- **K-means**: clusters the documents based on the text similarity.
- **LDA**: clusters the documents based on the word distribution.
- **DVD**: an iterative algorithm on a bipartite graph. With a community discovery algorithm and under Rocchio Classification framework, it classifies the documents to multiple versions.
- **CDW-lda**: implements a special version of CDW without using LDA.
- **CDW-pw**: implements a special version of CDW only using frequency of words instead of extracting popular words.

We compare the performance of CDW with related algorithms in the two built pairwise test sets, $T_{cs}$ and $T_{lsw}$. The $P_{score}$ performance of the pairwise judge task in $T_{cs}$ and $T_{lsw}$ is demonstrated in Fig. 4. From Fig. 4, we can see that the classic K-means and LDA are not satisfactory. Since DVD only uses the hierarchical relationships between words and ignore the information in text, it does not perform well either. CDW-lda and CDW outperform other related algorithms, which demonstrates that the popular words filtering is important and effective. Moreover, CDW performs little better than CDW-lda, which performs better than CDW-pw. It indicates that the mapping relationship between documents and topics can improve the performance.

VI. CONCLUSION

In this paper, we present an event diversified versions discovery model, which helps in quickly learning from multilateral description of a specific event. We use the highly-differentiated words to cluster the documents and introduce the concept of topic layer. The algorithm can be broken into three steps. First we use word frequency filter and popular words extraction to secure the highly-differentiated characteristics. Then we denote all existing documents with the highly-differentiated characteristics. Further, we use the K-means method to cluster documents in order to get the final diverse versions. Experiments on two real data sets show that our approach is effective.

While our work in this paper outperforms related algorithms, there is still some room for further improvement. An interesting direction is to explore automatic classification schemes so that we can present more trustable versions. Moreover, although our approach is capable of discovering different versions of one event, it would be more practical to extract content information of each version in a more systematic manner.

REFERENCES