Vector Representation of Words for Detecting Topic Trends over Short Texts

Liyian He*, Yajun Du and Lei Zhang
School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
*Corresponding author

I. INTRODUCTION

Researchers have shown an increased interest recently in analysing user intent, hot topics and social behaviors among these social platforms such as Twitter, Facebook and Microblog. Topic models, as a class of newly developed machine learning techniques, are employed to identify the underlying topics from text corpus. These topic models such as LDA [1] and PLSA [2], partly consider the semantic relatedness between words by incorporate topic information into a multinomial distribution over documents and words. Statistical techniques (e.g., Gibbs sampling) are then employed to identify the underlying topic distribution of each document as well as word distribution of each topic, based on the high-order word co-occurrence patterns [3,4]. However, these original topic models encounter a large performance degradation over short texts because of limited word co-occurrence information in short texts. Based on these models’ variants and others’ emerging technologies, such as neural network language models [5,6], several major heuristic strategies have been adopted to solve the data sparsity problem in short texts. Some researchers aim to aggregate a subset of short texts to form a long texts. And then topic models are applied over these long texts [7]. Other ingenious methods, such as MA-LDA [8] and MB-LDA [9], take some features into consideration and incorporate these features into LDA. It can outperforms the baseline of LDA empirically. Specifically, with the recent development in neural network techniques, Word embedding, first introduced in [10], have been successfully applied in language models and many NLP tasks. The combination of topic model and neural network language model prove to be a feasible way to enhance semantic relatedness [11].

Nowadays, people not only want to know what kind of topics can be extract from these text contents but also desire to gain the temporal dynamics of these topics. This calls for a more effective model that can track the evolution of trends over time in the analysis of user behavioral data. Many models have been proposed to incorporate temporal dynamics into topic models. The typical example in this category is Dynamic Topic Model (DTM), proposed by Blei [12]. Another model called Trend Detection Model (TDM), introduces a latent trend class variable into each document. However, a limitation of these methods is that word co-occurrence information that can be captured by these models because of data sparsity.

In this paper, we propose a novel model named topic trends detection (TTD) model. It contains an optimized topic model, which incorporates a term weighting pattern into a standard LDA model. Unlike existing models in which just incorporate time parameters and observed variables into Probabilistic generative model such as DMM [15,13,14], we combine our topic model with neural network language model in an unsupervised learning fashion which enables us to effectively and accurately detect the topic trends. In addition, the background knowledge about word semantic relatedness learned from a series of relevant short texts can be easily exploited to improve topic modeling by word embeddings. Through extensive experiments on several short text collections, we show our model’s effectiveness and accurateness over non-trivial baselines. The main contributions of this paper are summarized as follows:

(1) we propose an effective and feasible model which can not only extract latent topics over short texts correctly but also track temporal dynamic trends of these topics.

(2) we firstly present a newly topic model to generate more discriminative topics from aggregates of short texts accurately.

(3) Demonstrate the effectiveness of the model as compared to baseline models by experimenting on several microblog text collections.

The remainder of this article is organised as follows. In section 2, we review some previous work on extracting topics over short texts and tracking the topic trends over time. In section 3, we formalise our optimized topic model and TTD models. Our main experiments and results are presented in Section 4. The conclusions are discussed in Section 5.
II. RELATED WORK

In this section, we review three directions of related work. First, we summarize several classical topic models which try to infer topics for short texts. Second, we discuss the ingenious strategy of advanced models with word embeddings.

Topic models posit that each document is expressed as a mixture of topics, which are designed for categorical data. These topic proportions of standard topic model such as LDA [1], are drawn once per document, and the topics are shared across the corpus. This model is widely used to extract latent topic from text corpus. Another variant based on LDA called DMM shows that each short text is sampled from a single topic, known as mixture of unigrams or Dirichlet Multinomial Mixture (DMM) [15,13,14]. Blei et al. propose a correlated topic model (CTM) [16] that treats each document’s topic assignment as a multinomial random variable drawn from logistic normal prior. Yan et al. propose a novel way for modeling topics in short texts, referred as bitext topic model (BTM) [17]. It learns the topics by directly modeling the generation of word co-occurrence patterns. However, there is still a problem that these methods can not fully capture the semantic relatedness between the words. With the recent development in neural network techniques, vector space representations of words have contributed incremental improvements in researching word semantic relatedness [18,19,14].

Vector space representations of words, first introduced in [10], have been shown to capture lexico-semantic regularities in language. Words with similar syntactic and semantic properties are found to be close to each other in the embedding space [20,21]. According to the distributional hypothesis [22], words occurring in similar contexts tend to have similar meaning. Inspired by recent work in efficient word representation learning, Pennington et al.[6] introduce a new global log-bilinear regression model which contains a global matrix and a local context window. It is able to count data while simultaneously obtaining the meaningful linear substructures so as to leverage efficient statistics. Recently, an efficient architecture called fastText[26] is proposed for text classification. Because of several rank classifiers, it can achieve a fast loss approximation for training and evaluation. Liu et al. propose a model called Topical Word Embeddings (TWE)[23], which employ latent topic models to assign topics for each word in the text corpus, and learn topical word embeddings based on both words and their topics.

Multi-prototype vector space models [24] were proposed to cluster contexts of a word into groups, then generate a distinct prototype vector for each cluster. Since few of literatures research on the effects of selecting different vector space representations of words in short text modeling, we will discuss it in our experiment.

III. TOPIC TREND DETECTION(TTD) MODEL

TTD model is built to track topic trends for short texts based on a novel topic model and a effective neural network language model. Given a short document collections belong to a certain event, an optimized topic model we proposed extracts several topic terms or itemsets according to its weighting pattern and conditional probabilities. As shown in Fig.1, the word embeddings utilized in TTD is pre-learned using the state-of-the-art word embedding techniques from current date collections. Next, we present the details of the proposed model TTD.

A. Pre-process Stage

As is known to us, the standard LDA model can’t perform well over short texts due to the data sparsity. we consider it from two aspects. The first intuitive solution to this problem is pooling method: merging related short texts together and presenting them as a single document to stand topic model. In this paper, we put up with a novel short text-pooling scheme that we aggregate short texts in a certain period of time to form a longer pesudo-document. Our topic model is then applied over these pesudo-documents. Since this method specifically designed for short text contents, it will eliminate the additional unavailable metadata absolutely and enhance the correlation between these documents. On the other hand, we restrict the document-topic distribution like DMM model [13], such that each short text is sampled from a single topic.

B. An optimized topic model

For most LDA based topic models, the words with high probabilities in topic’s word distributions are usually chosen to represent topics. There is a common phenomenon that several different topics usually have similar topic-word distribution. That means, these words most likely represent general concepts or common concepts of the topics and cannot distinctively represent the topics. In the following work of detecting topic trends, the same words will lead to the same related words because the high dimensional representation of a certain word is fixed. In addition, the words in topic representations generated by LDA are usually single words. These single words provide too limited information about the relationships between the words and too limited semantic meaning to make the topics understandable. Accordingly, we extend LDA model by incorporating the weighting pattern to generate a distinctive
topic word or itemset. In this paper, we exploit TFIDF weight as a part of weighting pattern for our model.

Shown in Fig.2, the proposed topic model extends the Latent Dirichlet Allocation (LDA) model by incorporating the weighting pattern. The details of Generation process for our topic model is described in Algorithm 1. $M$ is the total number of documents, $K$ is the number of latent topics, $\alpha$, $\beta$ is Dirichlet parameters, $\varphi_k$ is word distribution for topic $k$, $\vartheta_m$ is topic distribution for document $m$, $N_m$ is the length of document $m$, $Z_{mn}$ is topic index of $n_{th}$ word in document $m$, $\omega_{mn}$ is the $n^{th}$ word of $m^{th}$ document. For the topical document, the word feature distribution over topic $k$, denoted as $td_k$ is generated based on their tf-idf scores, which are calculated by (1). $td_{i,j}$ is the $j^{th}$ term of $i^{th}$ document. $tf(td_{i,j})$ is the frequency of term $td_{i,j}$ in the $i^{th}$ document, $m_i$ is the $i^{th}$ document that consists of the words for a particular topic, $|m_i|$ is the count of terms in $m_i$, $N(td_{i,j})$ is the count of $td_{i,j}$ appearing in $m_i$.

In (2), Symbol $\gamma$ means that the current counting $i$ is excluded from the total counting. $\alpha$ and $\beta$ Dirichlet priors, $n_{k,-\gamma}$ is the number of times that word $w$ is assigned to topic $k$ and exclude the current counting. $V$ represents the vocabulary size and $K$ is the number of pre-defined latent topics. $n_{k,-\gamma}$ is the number of times that topic $k$ is assigned to topic document $m$ and exclude the current counting. The first ratio expresses the probability of word $i$ under topic $k$ as (3), and the second ratio expresses the probability of topic $k$ in document $m$ as (4). Specially, these counts are the only information necessary for computing the full conditional distribution, allowing the algorithm to be implemented efficiently by caching the relatively small set of nonzero counts. The biggest difference between the traditional topic model and the optimized topic model we proposed is that our model is built upon the weighting pattern. For each document, we get the word distribution over topic $k$. And then, we sort the words belong to each topic in TF-IDF value of these words as Table 1. The basic idea of the proposed method is to use the frequent itemsets generated from each document $D_i$ to represent topic $k$. For a given minimal support threshold $\xi$, where $F(w)$ is the frequency of $w$ which is the number of itemsets in $D_i$ that contain $w$. For example of $D_2$, set a minimal support threshold $\xi = 2$, all the frequent itemsets generated from $D_2$ are given in Table 2. We can notice that $\{w_1\}$, $\{w_3\}$ and $\{w_1, w_3\}$ are the dominant topics for Document 2. Comparing with the term based topic representation, itemsets represent the correlative words that carry more concrete and identifiable meaning. For example, "Houston Rockets" is more concrete than just one word "Houston" or "Rockets". In this paper, we treat itemsets as the dominant topics for the aggregates of short texts. For tracking trends of complete events, we aggregate the microblog texts of one event in a time period to a document and then obtain the the dominant topics from Topic model. Finally, we extract detailed information using word embeddings.
C. Auxiliary Word Embeddings

Although our topic model can extract discriminative terms in short texts to some extent, this prior preference for semantic coherence is not encoded in the model, and any such observation of semantic coherence found in the inferred topic distributions is, in some sense, accidental. In other words, two words having strong semantically relatedness but rarely co-occurring is, in some sense, accidental. In other words, Given two words having strong semantically relatedness but rarely co-occurring in a short text corpus, these models can not fully capture the semantic relatedness between the two words. At end of our (TTD) model, we utilize neural network language model for exploiting the contiguity of semantically similar words in the embedding space. In the experiment, we use Google’s Word2Vec toolkit with Skip-gram algorithm because Skip-gram algorithm can perform well on small date collections and catch the distinctive word semantic relatedness knowledge. As mentioned in section 3.2, we set minimal support $\xi = 3$ and choose the topic itemsets from top 10 words with high probabilities in topic’s word distributions. For all the methods in comparison, the hyper-parameter $\beta$ used in baseline LDA and DMM models was set to 0.01. We set the hyper-parameter $\alpha = 0.1$ as this can improve performance relative to the standard setting $\alpha = \frac{50}{T}$. Since microblog texts have the characteristic of immediacy, the release time of every microblog is considered as a certain point in time among the event evolution. We apply aggregate pattern to all of models for the fairness of comparison.

C. Evaluation by Topic Coherence

The topics generated by each model are evaluated by the topic coherence metric. Conventionally, topic models are evaluated by perplexity. However, there is a limitation that perplexity can not reflect the semantic coherence of a topic. Accordingly, it can sometimes be contrary to human judgments. Topic coherence measures the extent that the most probable words of a topic tend to co-occur together within the same documents. It proved to be a better metric to assess topic quality.

Here, we use the PMI-Score proposed in [25] to calculate topic coherence. PMI has been studied extensively in the context of collocation extraction. Given a topic $k$ and its top $T$ words with highest probabilities $w_1, \ldots, w_T$, the PMI-Score of $k$ is:

$$\text{PMI}(k) = \sum_{i=1}^{T} \sum_{j=1}^{T} \frac{f(w_i, w_j)}{f(w_i) f(w_j)} \log \frac{f(w_i, w_j)}{f(w_i) f(w_j)}$$

where $f(w_i)$ is the frequency of word $w_i$, and $f(w_i, w_j)$ is the frequency of the co-occurrence of $w_i$ and $w_j$.
\[ PMI(k) = \frac{2}{T(T-1)} \sum_{i,j \in T} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \] (5)

In (5), \( p(w_i) \) refers to the probability that word \( w_i \) appears in a document, and \( p(w_i, w_j) \) means the probability that words \( w_i \) and \( w_j \) appear in the same document. The overall topic coherence for each model is the averaged PMI-Score over all learnt topics. A higher topic coherence indicates the better learnt topics. Note that it need an external corpus, we use 3 hundred thousand Sina Microblog texts for our experimental datasets since microblog language has its own distinctive features. Fig.3 shows PMI-Score computed for the baseline LDA, DMM and TTD models on the microblog dataset with number of top words per topic \( T = \{5,10,15\} \) and number of topics \( K = \{30,50,70\} \), respectively. As we can see, our model has a better performance in topic coherence across all settings. That is because we incorporate a novel weighting pattern into the standard topic model. It can detect more representative words. DMM model is the second best model in most cases, and always outperforms LDA. Since it is a specific model for short texts which assumes each short document has only one topic. Compared with other methods, baseline LDA model has a poor performance in topic coherence for the limited word co-occurrence information, which impedes the generation of discriminative document-topic distributions.

Fig.4 presents the trends of PMI score in different number of topics vary from 30 to 70. We make the following observations. As selecting more Top-N words, the PMI score by every model shows descending trend in the same number of Topics. The reason is that choosing more words means the implication of more relevant topics. From (5) we can see, it increases more word pairs and reduce the scores of PMI. For learning vector representations of words and evaluating topic coherence, we used jieba, a python package segmentation tool, to segment the words in the microblog datasets. As is known to us, microblog language is completely informal. The accuracy in Chinese word segmentation might cause some performance variations.

**D. The Comparison of Choosing Different Word Representations**

Through extensive experiments, we observe that different vector space representations of words and learning approaches will generate different relevant words. In view of this, we will discuss the influences on topic coherence caused by choosing different word representations. We choose other two popular vector representations of words: Glove and fastText, and to do comparative analysis on topic coherence with our word representations which is based on word2vec. We train 50-dimensional word vectors from the aggregations of five events in Table 3 respectively for these three models. We set the parameter lr(learning rate)=0.05, min_count(minimal number of word occurrences)=1, w2vsize of the context window)=5, max_vocab size= 400000 respectively and we choose Topic-5 words to calculate PMI-Score to guarantee the fairness of comparative experiment.
Table 4 report the topic coherence metric of selecting different initial word representations on the aggregation of five event datasets in Table 3. We make the following observations. Specifically, Word2vec performs slightly better than other two models based on the same number of topics. Glove incorporates a global matrix into a new global log-bilinear regression model so as to leverages statistical information fully and it may consistently outperforms word2vec in some word analogy tasks. Unfortunately, it is not not applicable to extract topic words. Because of a global matrix, Glove will selectively focusing on parts of words which frequently in short texts and high-frequency words will be assigned to more significant weights. This phenomenon is useless for our purpose because usually the distinctive words appear in relatively lower frequency in short texts. fastText, which is designed for text classification, is similar to word2vec in structure, and unlike word2vec, fastText is a simple linear model with rank constraint. At the first stage of fastText, the word representations obtained from a initial weight matrix over words are averaged into a text representation, and it is prone to missing some word properties. As fastText is proposed for text classification, it has its own superiority in text classification, for instance, training efficiency. It is able to train on a billion words within ten minutes using a standard CPU, which is faster than word2vec. Since our datasets are not big enough, fastText model may not optimally exhibit the result.

E. Trend Detection in Microblog

Ultimately, we are interested in utilizing our method over microblog events to detect emerging trends in realtime. Since Sina Microblog has the property of immediacy, we treat each day as an unit to analysis the trend evolution by using TTD. Here, we display our TTD model applying in five microblog events in Table 4. As shown in Table 4, we took the $R(M)$ as the ratio of the amount of microblog texts that day to the amount of microblog texts of the whole event to discover the level of public concerns. Topic words are also the itemsets obtained from our model. Related words indicate the top 5 words obtained from Goolge’s Word2Vec toolkit with Skip-gram algorithm that have the highest similarity compare to topic itemsets. From above results, note that the amount of Microblog collections in the second day usually has a higher proportion. When a hot event emerged firstly, the Microblog amount should increase quickly in a short time. That means more people begin to concentrate on a specialized topic and discuss about it. And then, Microblog amount in next days decreases in some extent, which indicates that people have acquired the detailed information and they reduce the overall concern about the current event unless another relevant event appears. If a hot event is inconclusive in the near future, it will keep relatively higher attention in the end like #德事件#. That because the Microblog text collections of #德事件# are not according to the event occurrence time but the time after. For #


