Topic Tracking with Dynamic Topic Model and Topic-based Weighting Method

Xiaoyan Zhang
Department of Computer Science and Technology, School of Computer, National University of Defense Technology
No.137, Yanwachi Street, Changsha, Hunan 410073, P.R.China
Email: zhangxiaoyan@nudt.edu.cn

Ting Wang
Department of Computer Science and Technology, School of Computer, National University of Defense Technology
No.137, Yanwachi Street, Changsha, Hunan 410073, P.R.China
Email: tingwang@nudt.edu.cn

***Abstract*—In topic tracking, a topic is usually described by several stories. How to represent a topic is always an issue and a difficult problem in the research on topic tracking. To emphasis the topic in stories, we provide an improved topic-based *tf*idf* weighting method to measure the topical importance of the features in the representation model. To overcome the topic drift problem and filter the noise existed in the tracked topic description, a dynamic topic model is proposed based on the static model. It extends the initial topic model with the information from the incoming related stories and filters the noise using the latest unrelated story. The topic tracking systems are implemented on the TDT4 Chinese corpus. The experimental results indicate that both the new weighting method and the dynamic model can improve the tracking performance.**

***Index Terms*—topic tracking, topic drift, dynamic topic model, feature weighting**

I. INTRODUCTION

Topic Detection and Tracking (TDT) [1] refers to a variety of automatic techniques for discovering and threading together topically related material in a stream of data such as newswire or broadcast news. It could be quite valuable in many applications where people need timely and efficiently access to large quantities of information such as [2] and [3].

In TDT, topic is an important concept. A topic is defined as *a seed event or activity plus directly related events or activities* and usually described by a sequence of time-ordered stories. Since existing models are not special enough to emphasis the topic in stories, an improved topic-based *tf*idf* (*T-tf*idf*) is proposed to order the features in the representation model according to the topical importance. In addition, a topic may normally evolve with the passage of time, which leads to the topic drift phenomenon [4]. The focus of a topic may change dynamically. The noise in the topic description will also change dynamically in the development of a topic. The traditional static models are no longer suitable for representing the topic. In these conditions, a dynamic model is provided to improve the classical static vector model in two ways: dynamically adding on-topic information in the incoming related stories into the initial topic model; dynamically filtering the noise in the topic model using the incoming unrelated stories.

This paper uses the improvements for the topic tracking. As a main task of TDT, topic tracking is to associate incoming stories with a topic and find all the stories related to it in a stream of stories. The topic is pre-described by a few topically-related stories. In our topic tracking system, *T-tf*idf* weighting method measures the features of the tracked topic and all the single testing stories, and is also used for re-weighting the features in the incoming related stories when updating the topic model; the dynamically updating technology is used for the tracked topic model. The experimental results show that both improvements can improve the performance.

The rest of this paper is organized as follows: Section II gives the definition of topic tracking and its evaluation method; Section III introduces the related work and the motivation of this paper; Section IV describes the research framework of the tracking system; Section V is the static representation model for the tracked topic and the testing stories, mainly introducing the new weighting method; Section VI presents the dynamic topic model; the experimental results are given and discussed in Section VII; finally, Section VIII concludes the paper.

II. PROBLEM DEFINITION AND EVALUATION

In the definition of topic tracking, a tracking system is given a topic identified with a few related news stories and a stream of time-ordered news stories $S = < s_1, s_2, s_3, ..., s_n >$. The system is required to sequentially find all the news stories related to the topic in the stream. The tracking procedure can be formalized as follows:

- Construct a representation model with the given related stories for the tracked topic.
- For each incoming story s_i in the steam:
  - Construct the representation model for s_i.

Manuscript received May 20, 2009; revised June 30, 2009; accepted July 23, 2009.
Corresponding author: Xiaoyan Zhang

© 2010 ACADEMY PUBLISHER
doi:10.4304/jsw.5.5.482-489

- Compute the similarity score between the topic model and the story model. If the similarity is larger than a predefined threshold, then s_i is decided to be topically related to the tracked topic, otherwise, s_i is off the topic.

To evaluate the tracking system, the TDT 2003 evaluation method[1] is used. It produces an error detection cost due to the errors caused by the system. The cost is a linear combination of the miss probability and the false alarm probability. The formula is as follows.

$$C_{det} = C_{miss} \times P_{miss} \times P_{target} + C_{fa} \times P_{fa} \times P_{non\text{-}target} \quad (1)$$

C_{det} is the error detection cost. The smaller C_{det} is, the better the performance is. C_{miss}, C_{fa}, P_{target} and $P_{non\text{-}target}$ are predefined values, respectively set as 1, 0.1, 0.02 and 0.98. The former two are the costs of missing or false alarming a story about the tracked topic. The later two are the prior probabilities of whether a story is related to the tracked topic, or not. P_{miss} and P_{fa} are the conditional probabilities of missing and false alarming a related story.

After being normalized, the minimum value of $(C_{det})_{norm}$ is still 0. If the value of $(C_{det})_{norm}$ is 1, it means that the system is not better than the system that considers all the testing stories as related or not. The normalization is done as follows.

$$(C_{\det})_{norm} = \frac{C_{\det}}{\min(C_{miss} \times P_{t\arg et}, C_{fa} \times P_{non-t\arg et})} \quad (2)$$

For more than one topic tracking system, there are two reasonable methods for computing P_{miss} and P_{fa}: story-weighted and topic-weighted. The story-weighted method assigns equal weight to each decision for each story and accumulates errors over all topics. The topic-weighted method accumulates errors separately for each topic and then averages the error probabilities over topics, with equal weight assigned to each topic. The later one can usually provide better estimation of the performance. Therefore, the experimental results in this paper are all topic-weighted.

If the tracking system makes decisions with a predefined threshold, we take the generated $(C_{det})_{norm}$ as the current cost. By sweeping the threshold over the score range, the minimum $(C_{det})_{norm}$ that the system can acquired is obtained when the system makes decisions with the optimum threshold. These two values reflect the current performance and the optimum performance respectively. The current performance is considered to be the primary evaluation metric by standard TDT evaluation, since topic tracking is a kind of practical application-oriented research.

III. Related Work and Motivation

The representation of the tracked topic (a set of stories) or single stories is a fundamental problem in topic tracking. Most representation methods are borrowed from the related research fields. The weights of the features in these models are usually computed with the traditional methods such as *tf*idf* weighing[2], conditional probability[3] and generation probability[4]. Some improvements are also proposed: directly increasing the weights of some important features [5]; re-estimating the weight according to the prior joint probability of the feature and other topics in the training data [6]. However, these measures do not change the nature of existing weighting methods. The weight of a feature still measures its importance relative to the main content of a story, not the topic in the story; to a single story, not a set of stories. In this paper, an improved topic-based weighting method is proposed based on the traditional *tf*idf* weighting. We call it *T-tf*idf* weighting. With the new *T-tf*idf* weighting method, the features in the model can be ordered according to the topical importance.

For the representation of the tracked topic, many technologies have been provided to improve the topic model such as [7], [8], [9], [10]. However, the topic drift problem must be considered. The existing methods use the incoming related stories to adjust the initial topic model. For example, [11] directly takes the features with their weights in the incoming on-topic stories to update the topic model. In [12], the term frequencies of the topic model are refreshed by adding the term frequencies of the incoming related story weighted with a coefficient. [13] simplifies the previous method by setting the similarity score as the coefficient and gets better performance. However, the stories are all learned independently in these methods. Sometimes, some information in a related story is not important enough to be learned when being compared with that in other related stories. Besides, if some pseudo related stories are selected for learning, a lot of noise will flow into the topic model in the above methods. To overcome these limitations, this paper proposes a novel learning method for the incoming related stories. Before being extended into the topic model, all the incoming related stories together with their similarity scores will first be put into a set. Taking this set as a whole, the weights of the features in it are re-computed. Then some features are selected according to the weights and used to update the topic model. This method selects important information while taking all the related stories as a whole, and the noise in the pseudo related stories can also be avoided at the same time.

As for the dynamic noise in the topic model due to the topic drift, it has not been considered in previous research on topic tracking. To settle the problem, we first take the following assumption: if a feature appears in a story about a topic, it usually appears with a low weight or even does not appear in stories about other topics. With this consideration, the previous stories, which are unrelated and close to the tracked topic, are chosen to

[1] The 2003 Topic Detection and Tracking Task Definition and Evaluation Plan, http://www.itl.nist.gov/iaui/894.01/tests/tdt/tdt2003/evalplan.htm

[2] http://en.wikipedia.org/wiki/Tf-idf

[3] http://en.wikipedia.org/wiki/Conditional_probability

[4] http://nlp.stanford.edu/IR-book/html/htmledition/estimating-the-query-generation-probability-1.html

© 2010 ACADEMY PUBLISHER

locate the noise in the tracked topic model. In topic tracking, when comparing the tracked topic and a testing story, the information in both the topic model and the latest unrelated story model is most likely to be dissimilar with the story model. Therefore, the latest unrelated story is used to filter the noise in the topic model before the comparison. In our method, the topic model is always updated with the related stories first, and then the unrelated story.

IV. Research Framework

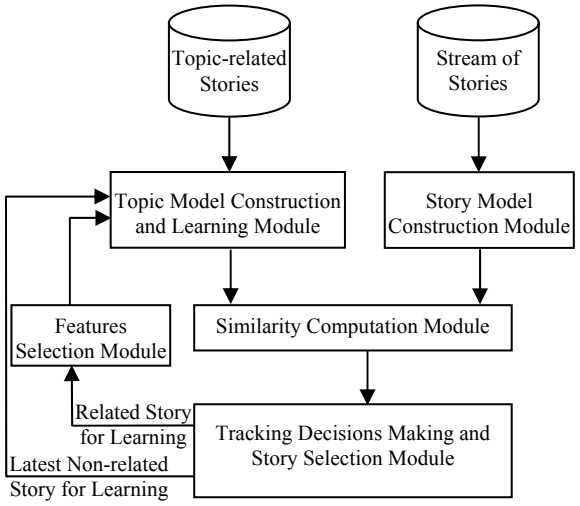

Figure 1. Framework of the tracking system

Figure 1 shows the framework of the tracking method. The initial topic model is constructed by the Topic Model Construction Module. For each comparison between the topic and the incoming testing story:

- First, the story model is constructed by the Story Model Construction Module.
- Second, the similarity between the current topic model and the story model is computed by the Similarity Computation Module.
- Third, the Tracking Decision Making Module decides whether the story describes the topic according to the similarity. If the similarity is larger than a predefined tracking threshold, the story is on the topic. Otherwise, the story is off topic.
  At the same time, the Story Selection in the same module will decide whether the story is used for learning.
  - If the story is related and to be learned, it will be put into the learning set. Features Selection Module extracts features according to the feature weights. In the Topic Model Learning Module, the current topic model rolls back to the initial model and learns the features.
  - If the story is unrelated and to be learned, it will replace the previous learned unrelated story in the Topic Model Learning Module. In the learning method, the weights of the features in the topic model are reduced when the features also occur in the current unrelated story.

In the above figure, *T-tf*idf* weighting method is used in Model Construction Modules and Features Selection Module.

V. Static Model

In this paper, we take the classical vector as the baseline static model [11], [14]. The static model represents the tracked topic and testing stories as vectors in feature space, where the coordinates represent the weights of the features. Each feature is a single word. For the topic model, the feature set is the union of the candidate feature sets from known topically related stories. The feature weights are computed with *T-tf*idf* weighting method. The similarity score is always computed with the cosine function. Because of the limited space, the cosine function [15] will not be introduced here. The focus of this section is on the feature weighting method.

A. Preprocessing

Preprocessing is the first step in creating a representation model. For each story, the preprocessor tokenizes and tags the text, and removes stop words. After that, a set of candidate terms is obtained for the vector model. If a word has more than one tag, it will represent multiple features. The length of the story is computed after the preprocessing. The tokenizer and tagger we use is ICTCLAS[5]. The stop word list consists of 507 words.

B. Feature Weighting

Feature weighting is an important component in the representation. Through changing the sources of some parameters in the traditional *tf*idf* weighting, the *T-tf*idf* weighting tries to measure the importance of a feature relative to the topic in a set of stories. The details of the *T-tf*idf* weighting method are as follows.

$TS = \{T_i \mid i = 1, ..., n\}$ is a set of known topics. Each topic $T_i = \{s_{ij} \mid j = 1, ..., m\}$ is described by a set of related stories, which can be considered as a big story putting all the related stories together. T_i is represented as a set of features after being preprocessed. In this paper, T_i is a topic to be tracked, and s_{ij} is a given related story. In the evaluation experiment of this paper, n is 70 and m is 4.

For a set of stories or a single story (a set containing only one story) T, it is represented as a set of features $\{f_k \mid k=1,..., t\}$ after being preprocessed. The *T-tf*idf* weight of f_k is computed as follows.

$$weight(f_k, T) = tf(f_k, T) * idf(f_k) \quad (3)$$

$$tf(f_k, T) = \frac{fre(f_k, T)}{fre(f_k, T) + 0.5 + 1.5\frac{dl(T)}{dl_{avg}(TS, T)}} \quad (4)$$

$$dl_{avg}(TS, T) = \frac{docSum(T)}{\sum_{T_i \in TS} docSum(T_i)} \times \sum_{T_i \in TS} dl(T_i) \quad (5)$$

[5] http://sewm.pku.edu.cn/QA/reference/ICTCLAS/FreeI-C TCLAS/

© 2010 ACADEMY PUBLISHER

$$idf(f_k) = \frac{\log(\frac{topicSum(TS)+0.5}{df(f_k,TS)})}{\log(topicSum(TS)+1)} \quad (6)$$

In the preceding formulae: $fre(f_k, T)$ is the frequency of f_k in T; $dl(T)$ is the sum length of all the stories in T; $dl_{avg}(TS, T)$ is the average length of TS, and also related with the number of stories in T; $docSum(T)$ is the number of stories in T; $topicSum(TS)$ is the number of topics in TS; $df(f_k, TS)$ is the number of topics containing f_k in TS. In *T-tf*idf* weighting, $tf(f_k, T)$ shows how much f_k represents T, and $idf(f_k)$ reflects the ability of f_k to distinguish different topics. Both of them estimate the importance of f_k to a set of stories about a same topic instead of a single story.

The main difference between *T-tf*idf* and *tf*idf* weighting methods is how to get the value of *idf* of a feature. The former one takes the training data as a set of story sets and *idf* is the number of sets containing the feature. The later one takes the data as a set of stories and *idf* is the number of stories containing the feature. *T-tf*idf* weighting takes a story set, not a single story, as the minimum statistics scope. Besides *tf*idf* weighting, the characteristics of TDT and the stories are also considered in *T-tf*idf* weighting. This method is more suitable for topic tracking, as will be demonstrated by our work.

VI. Dynamic Topic Model

In this paper, the dynamic topic model is updated in two ways as the tracked topic evolves: using the incoming related stories to extend the topic model, and using the incoming unrelated stories to filter the noise in the topic model. Since the incoming related stories are used to complement the topic description, the learning process should take into account all the related stories. However, since the unrelated stories are to filter the latest most likely noise in the topic model when being compared with the incoming testing story model, the located information should always come from the latest unrelated story. In addition, if we use all the unrelated stories, the topic model after filtering will gradually be generalized not to represent the tracked topic. The following will explain in detail how to dynamically update the topic model.

A. Updating with Related Stories

The first step of learning with related stories is story selection. In our investigation, it is observed that the number of related stories in the testing stream is usually far less than that of unrelated stories. Even if the false alarm probability is very small, there are still many pseudo related stories in the tracking result. With this consideration, the threshold for selecting related stories to update the topic model is not easy to set, and it should be large enough at least. We take the most frequently used empirical method: the selection threshold we use is 0.35, which is much larger than the tracking threshold (0.07). The tracking threshold for the system using the dynamic topic model is same as that for the system using the static model.

All the selected related stories will first be put into a learning set (*LS*), and a feature selection procedure works on the set as follows. *LS* is seen as a whole consisting of related stories weighted with their similarity scores. The feature weighting method is same as that for the tracked topic. The only difference between them is that the stories in *LS* is weighted with their similarities, instead of the constant weight 1 used in the tracked topic. That is to say, the feature frequency in a story of *LS* is multiplied with the similarity of the story before being used. This way can further help measuring how much the features in *LS* should be used for learning. After that, the first twenty features are selected according to the feature weights to update the tracked topic model. Twenty is still an empirical value here. It can be seen that the importance of selected features are global among all of the related stories. The noise in the pseudo related stories can also be eliminated in this procedure.

There are usually two methods [16] to extend the topic model with the selected features: the average method and the increment method. In our investigation, the former one is better and it is used in this paper. This method can add new features without losing information in the topic model. The learning method is as follows. For each selected feature f_i:

- If it also occurs in the topic model T as f_j, then the weight of f_j in T will be reset as the average value of the weight of f_i and the old weight of f_j in T;
- Otherwise, f_i together with its weight will be directly added into the topic model.

Note: if a new related story is added into the learning set, the feature re-weighting and selection procedure is called again; the topic model will first roll back to the initial model when learning the new set of related features, otherwise, some features may be accumulatively learned, which is not supposed to happen.

B. Updating with Unrelated Stories

Under the assumption that the information in the unrelated stories will not be so important when appearing in the tracked topic, the unrelated stories are used to locate and weaken the noise in the topic model. Compared with the related stories, the unrelated stories are easier to select. It is because the number of unrelated stories is always much larger than that of pseudo unrelated stories in the tracking results. The probability of selecting a real unrelated story from them is usually very high. Therefore, the threshold for selecting an unrelated story is less difficult. In this paper, the selection threshold is empirically set as one tenth of the tracking threshold. If the similarity between a story and the topic is larger than the selection threshold and smaller than the tracking threshold, the story is unrelated and used for learning.

Compared with other unrelated stories, these selected unrelated stories are close to the topic and can locate the noise in the topic model more accurately. According to the temporal property of a topic, we choose the latest unrelated story to locate the noise in the topic model. The information in both the topic model and the latest unrelated story model is most likely to be dissimilar with the incoming story model.

© 2010 ACADEMY PUBLISHER

The learning method for the unrelated story is as follows. For the features in both the latest unrelated story and the topic model, their weights in the topic model will be reduced. In our research, the reduction proportion is set empirically, which is 0.9 in this paper. It is notable that the topic model will undo the effect of using the previous unrelated story on the feature weights whenever encountering a new unrelated story.

VII. Experiments and Discussions

The implemented tracking systems are tested on the Chinese subset of TDT4 corpus[6]. The data supports for 70 topic tracking tasks. Each topic tracking is provided with a topic described with four related stories and a stream of testing stories. For the testing streams, there are 48 of them that contain at least one related story for their tracked topics, and 22 of them that have no related story for their tracked topics.

A. Experiments on T-tf*idf weighting method

To verify the effectiveness of the *T-tf*idf* weighting method, the following tracking methods are implemented. Table 1 shows their topic-weighted experimental results.

TABLE I
THE TOPIC-WEIGHTED EXPERIMENTAL RESULTS OF TRACKING SYSTEMSS USING STATIC TOPIC MODEL

		TTS-C	TTS-T
Current Performance	P_{miss}	0.0118	0.0202
	P_{fa}	0.0902	0.0156
	$(C_{det})_{norm}$	0.4538	0.0967
Optimum Performance	P_{miss}	0.0998	0.0421
	P_{fa}	0.0108	0.0070
	$(C_{det})_{norm}$	0.1527	0.0766

- Topic tracking with the static vector model and classical *tf*idf* weighting method (**TTS-C**);
- Topic tracking with the static vector model and *T-tf*idf* weighting method (**TTS-T**);

Compared with TTS-C, TTS-T reduces $(C_{det})_{norm}$ by 78.69% and 49.84% in the current and optimum performance. Except the current P_{miss} of TTS-C, other P_{miss} and P_{fa} all benefit from the improved *T-tf*idf* weighting method. *T-tf*idf* weight aims to measure the topical importance of the features in the model, which is the object of the representation in TDT. However, the more accurate model of the tracked topic and the testing story would cause a bigger current P_{miss} in TTS-T by contraries when the tracked topic drifts. The dynamic topic model verified in the next section is proposed to adjust the topic drift problem. Altogether, the improved weighting method is more suitable than the classical *tf*idf* method for representing the topic and stories in TDT. It's another usage of re-weighting the extended related features when dynamically updating the topic model will re-prove the effectiveness.

B. Experiments on the dynamic topic model

To verify the effectiveness of the dynamic topic model, we implemented the following tracking systems. Table 2 shows their topic-weighted experimental results.

[6] http://projects.ldc.upenn.edu/TDT4/

TABLE II
THE TOPIC-WEIGHTED EXPERIMENTAL RESULTS OF TRACKING SYSTEMS USING DYNAMIC TOPIC MODEL

		TTD-RL	TTD-RUL	Supervised TTD-RUL
Current Performance	P_{miss}	0.0194	0.0191	0.0130
	P_{fa}	0.0156	0.0152	0.0151
	$(C_{det})_{norm}$	0.0956	0.0937	0.0870
Optimum Performance	P_{miss}	0.0370	0.0405	0.0359
	P_{fa}	0.0081	0.0072	0.0060
	$(C_{det})_{norm}$	0.0765	0.0759	0.0654

- Topic tracking with dynamic topic model just using incoming related stories (**TTD-RL**);
- Topic tracking with dynamic topic model using both incoming related and unrelated stories (**TTD-RUL**);
- The tracking method is same as TTD-RUL, but the incoming stories are human selected, not above system automatically-selected (**Supervised TTD-RUL**);

Compared with TTS-T, the usage of the incoming related stories can reduce $(C_{det})_{norm}$ by 1.14% and 0.13% in the current and optimum performance. The impact of the usage of unrelated stories further decreases $(C_{det})_{norm}$ by 1.99% and 0.78% in the current and optimum performance. The former improvement focuses on reducing P_{miss}. The later one can reduce both P_{miss} and P_{fa}, which is to say the P_{miss} decreases at no cost of increasing P_{fa}. The improvements are few since the baseline method has already improved the performance remarkably by using the *T-tf*idf* weighting. Although the minimum $(C_{det})_{norm}$ of TTD-RL and TTD-RUL are a little decreased compared with that of TTS-T, the current performance (the primary evaluation metric) is improved obviously.

Supervised TTD-RUL indicate the upper bound of TTD-RUL. P_{miss} and P_{fa} in the current and minimum performance of the supervised method, especially P_{miss}, are further reduced compared with those values of the unsupervised TTD-RUL. In this condition, the linear-combined values of $(C_{det})_{norm}$ are reduced apparently. The current $(C_{det})_{norm}$ drops by 7.15% from 0.0937 to 0.0870 and the minimum $(C_{det})_{norm}$ drops by 13.83% from 0.0759 to 0.0654. The upper bound of the performance exhibits the great potential of our dynamic model. Besides the updating method of the topic model, the selection of the incoming stories should also be more carefully studied.

To further confirm the effectiveness of the proposed method, the current performance of the single topic tracking is also analyzed. The tracked topics are divided into two parts. One part contains the topics without related stories in the testing streams, consisting of 22 topics. For this part, only P_{fa} is generated in the evaluation results. Figure 2 shows the comparisons of these probabilities. The other part contains the topics with at least one related story in their testing streams, consisting of 48 topics. Figures 3, 4 and 5 show the detailed evaluation information for the individual topic tracking on $(C_{det})_{norm}$, P_{fa} and P_{miss}.

From Figure 2, it can be seen that P_{fa} of TTD-RL is same as that of TTS-T. It is because that the similarities of the pseudo related stories in the testing streams are all smaller than the learning threshold. That is to say, no

© 2010 ACADEMY PUBLISHER

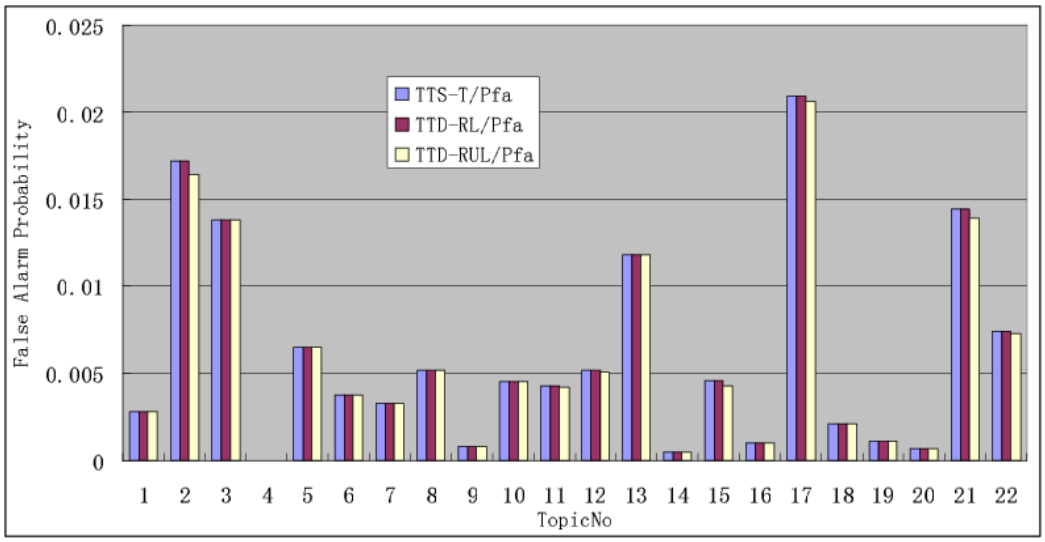

Figure 2. Pfa for the tracking systems without related stories in the testing streams

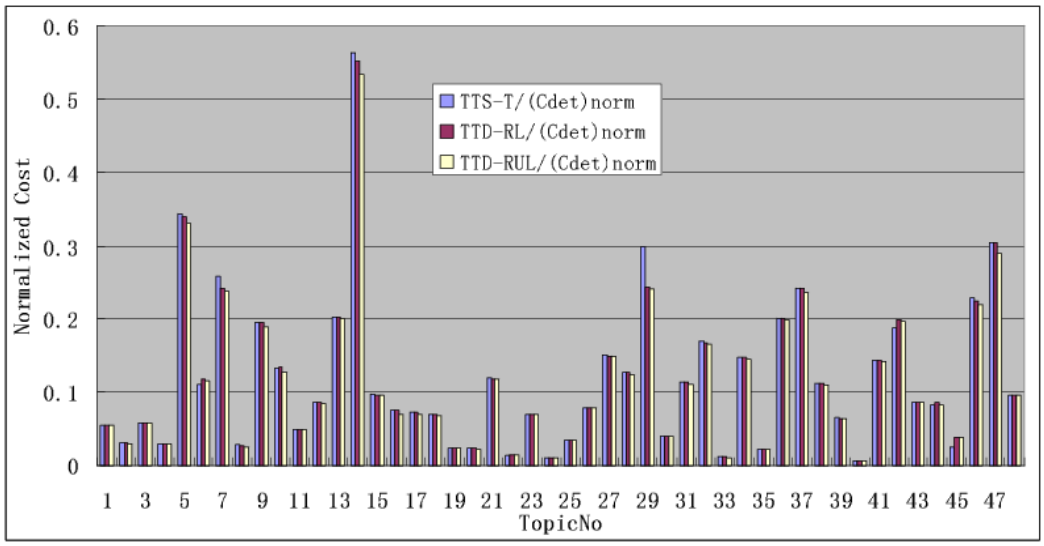

Figure 3. (Cdet)norm for the tracking systems with related stories in the testing streams

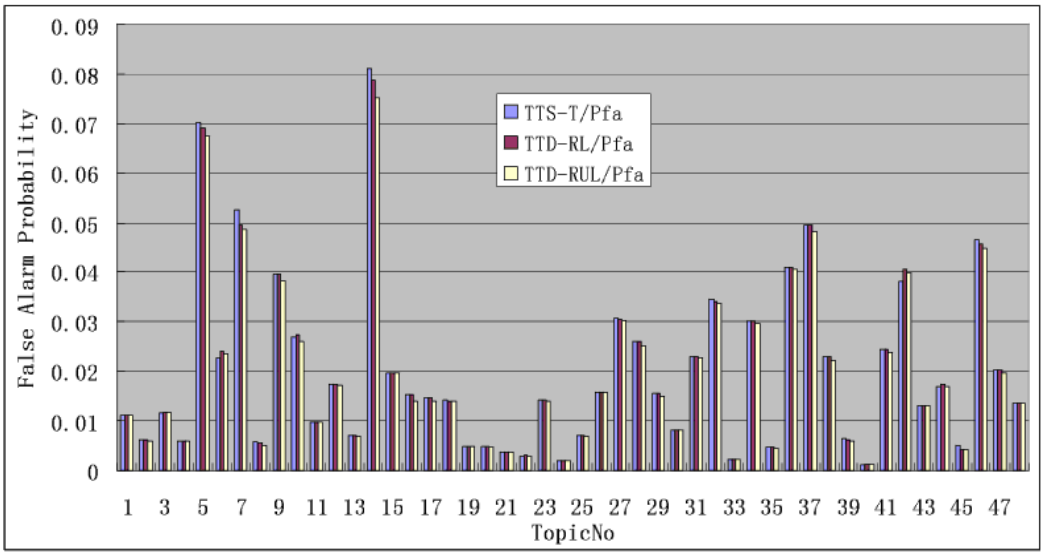

Figure 4. P_{fa} for the tracking systems with related stories in the testing streams

© 2010 ACADEMY PUBLISHER

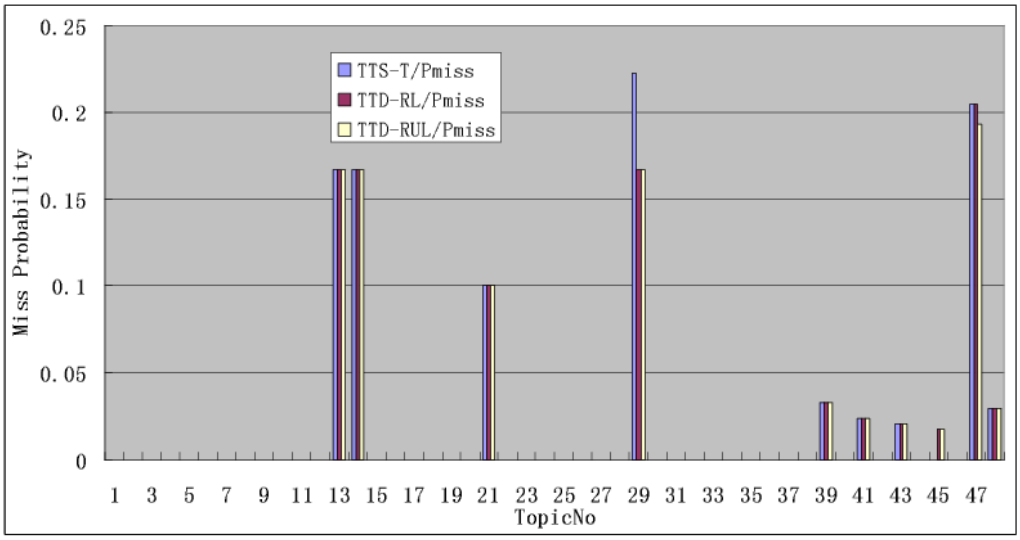

Figure 5. P_{miss} for the tracking systems with related stories in the testing streams

story is similar enough to be used for learning. In TTD-RUL, the unrelated stories are used to filter the noise in the topic model, which results in the reduction on the number of the pseudo related stories. P_{fa} decreases in the tracking results of 7 topics, and remains unchanged in the tracking results of the rest topics.

From Figure 3, we can see that TTD-RL has improved the tracking performance on almost all the topics compared with TTS-T, and TTD-RUL is even better. The improvement of TTD-RUL on $(C_{det})_{norm}$ is in 99%-confidence when being compared with both TTS-T and TTD-RL. Figure 4 shows that TTD-RL reduces P_{fa} on most of the topics, and TTD-RUL reduces these probabilities more than TTD-RL. Furthermore, compared with the number of topics with increasing P_{fa} in TTD-RL (9 topics), it decreases to 4 in TTD-RUL. In Figure 5, there are 38 topics on which P_{miss} remains zero and 7 topics on which P_{miss} remains unchanged in three systems. For the rest 3 topics, P_{miss} on topic 29 decreases from 0.2222 in TTS-T to 0.1667 in both TTD-RL and TTD-RUL, P_{miss} on topic 45 increases from 0 in TTS-T to 0.0197 in TTD-RL and TTD-RUL, P_{miss} on topic 47 decreases from 0.2045 in TTS-T and TTD-RL to 0.1932 in TTD-RUL. Altogether, based on TTS-T, the improvement of TTD- RUL on P_{miss} is 5.45% which is more than 3.96% obtained by TTD-RL.

C. Discussions

Besides the proposed method, the learning methods for the topic model in the related work are also implemented based on the static model in the TTS-T and evaluated on the same corpus. The learning and tracking thresholds in the following methods are same as those in TTS-T.

TABLE III
THE TOPIC-WEIGHTED EXPERIMENTAL RESULTS OF RELATED METHODS

		Method1	Method2	Method3
Current Performance	P_{miss}	0.0205	0.0194	0.0194
	P_{fa}	0.0162	0.0156	0.0156
	$(C_{det})_{norm}$	0.0997	0.0956	0.0956
Optimum Performance	P_{miss}	0.0377	0.0374	0.0405
	P_{fa}	0.0083	0.0080	0.0074
	$(C_{det})_{norm}$	0.0784	0.0768	0.0766

- Method1 [11]: All the features with their weights from each incoming related stories are extended into the topic model.
- Method2 [11]: The first twenty features with their weights are extracted from each incoming related stories, and then extended into the topic model.
- Method3 [13]: The models of the related stories are first weighted with the similarities between them and the topic model. Then, the first twenty features with their new weights are extracted from each model and extended into the topic model.

The incoming related stories are all accumulatively learned in the above methods. The learning method used to extend the information into the topic model is same as that in our proposed method. Table 3 shows their topic-weighted experimental results.

Compared Table 3 with Table 2, it can be seen that both the current $(C_{det})_{norm}$ and the minimum $(C_{det})_{norm}$ of Method1 are poorer than TTS-T because of the increase in P_{fa}. The reason for this is mainly on the noise in the stories for learning. Method2 and Method3 filter the noise and get almost the same improved current performance on P_{miss}, P_{fa} and $(C_{det})_{norm}$ as TTD-RL. For the optimum performance, the $(C_{det})_{norm}$ of Method2 is a little poorer than TTS-T still because of the increase in P_{fa}; The increase in P_{fa} of Method3 is less than that of Method2. However, since all the weights of the features for learning are compressed when filtering the noise, P_{miss} of Method3 is worse than that of Method2. Finally the minimum $(C_{det})_{norm}$ of Method3 is same as that of TTS-T and not improved. Therefore, it can be said that the noise filtering in Method2 and Method3 is still not handled well enough. Compared with the above related methods, TTD-RL extracts related information which is important globally among all the related stories and gets improved current and optimum performance. In addition, the usage of the unrelated stories during learning in TTD-RUL further improves the performance.

The construction of the dynamic model consists of two steps: locating related information and noise for the topic

© 2010 ACADEMY PUBLISHER

model; integrating the related information into the topic model and filtering the noise in the topic model. Both of them will affect the effectiveness of the topic model. However, currently our research mainly focus on the former step, while for the later step, just a simple method has been tried. How to better learn or filter the located information for the topic model will be the next step. In addition, the features used for the representation model still stay at the word level. The deeper information such as semantic information should be more efficient for the topic model. How to improve the topic model with the deeper information will also be our future work.

VIII. CONCLUSIONS

How to represent a topic is an important issue in topic tracking research. Though some classical models and improvements based on them have been used to represent a topic, there are still some characteristics that are ignored. After analyzing the topic characters and the deficiencies in the existing representation models, this paper first provides a topic-based *tf*idf* weighting to measure the topical importance of the features in the model, and then presents a dynamic topic model which focuses on two problems: the topic drift problem and the noise in the topic model. The experimental results indicate that the *T-tf*idf* weighting and the dynamic topic model can improve the tracking performance.

We also realize that there are some remaining problems in the dynamic topic model. Our method mainly focuses on the relevant information selection and the noise location. The approaches used to update the topic model with the related information and filter the noise in the topic model is still very simple and empirical. In addition, deeper information needs to be further extracted to enrich the representation model. We intend to explore these issues in the future work.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (60873097, 60403050); Program for New Century Excellent Talents in University (NCET-06-0926).

REFERENCES

[1] J. Allan, Ed., Introduction to Topic Detection and Tracking in Topic Detection and Tracking: Event-based Information Organization. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[2] K. K. Bun and M. Ishizuka, "Emerging topic tracking system in www," Jouranl of Knowledge-Based System, vol. 19, no. 3, pp. 164–171, 2006.

[3] G. Cselle, K. Albrecht, and R. Wattenhofer, "Buzztrack: Topic detection and tracking in email," in Proceedings of the 2007 International Conference on Intelligent User Interfaces, 2007, pp. 190–197.

[4] C. C. Aggarwal, Ed., Data Streams: Models and Algorithms. Charu Aggarwal Springer, 2007.

[5] J. Makkonen, H. Ahonen-Myka, and M. Salmenkivi, "Simple semantics in topic detection and tracking," Information Retrieval, vol. 7, no. 3-4, pp. 347–368, 2004.

[6] K. Zhang, J. zi Li, G. Wu, and K. hong Wang, "A new event detection model based on term reweighting," Journal of Software, vol. 19, no. 4, pp. 817–828, 2008.

[7] B. li Li, W. jie Li, and Q. Lu, "Topic tracking with time granularity reasoning," ACM Transactions on Asian Language Information Processing, vol. 5, no. 4, pp. 388–412, 2006.

[8] S. Lee and H. joon Kim, "News keyword extraction for topic tracking," in Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management, 2008.

[9] W. Zheng, Y. Zhang, Y. Hong, J. Fan, and T. Liu, "Topic tracking based on keywords dependency profile," in Proceedings of the 4th Asia Infomation Retrieval Symposium, 2008, pp. 129–140.

[10] B. li Li, W. jie Li, Q. Lu, and M. Wu, "Profile-based event tracking," in Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 631–632.

[11] J. Allan, A. Bolivar, M. Connell, S. Cronen-Townsend, A. Feng, F. Feng, F. Kumaran, L. Larkey, V. Lavrenko, and H. Raghavan, "Umass tdt 2003 research summary," in Proceedings of TDT workshop, 2003.

[12] Y.-Y. Lo and J.-L. Gauvain, "The limsi topic tracking system for tdt2001," in Proceedings of DARPA Topic Detection and Tracking Workshop, 2001.

[13] N. li Ma, Y. ming Yang, and M. Rogati, "Applying clir techniques to event tracking," in Proceedings of the First Asia Information Retrieval Symposium. LNCS 3411, Springer-Verlag, 2004, pp. 24–35.

[14] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan, "Umass at tdt 2004," in Proceedings of TDT2004 Workshop, 2004.

[15] J. Allan, V. Lavrenko, D. Malin, and R. Swan, "Detections, bounds, and timelines: Umass and tdt–3," in Proceedings of Topic Detection and Tracking (TDT–3), 2000, pp. 167–174.

[16] X. yan Zhang, T. Wang, and H. wang Chen, "Story link detection based on dynamic information extending," in Proceedings of the Third International Joint Conference on Natural Language Processing, 2008, pp. 40–47.

Xiaoyan Zhang was born in Henan of P.R.China in 1981. She received her B.S. degree and M.S. degree in Computer Science and Technology from National University of Defense Technology at Changsha of P.R.China respectively in 2002 and 2004, and currently is a Ph.D. candidate in the same place. Her major interests lie in topic detection and tracking and information retrieval.

Ms. Zhang is a student member of China Computer Federation.

Ting Wang was born in Changsha of P.R.China in 1970. He received his B.S. degree and Ph.D. degree in Computer Science and Technology from National University of Defense Technology respectively in 1992 and 1997. He is currently a professor and doctoral supervisor at National University of Defense Technology. He has published more than 40 papers and undertaken four projects as the principal investigators. His research interests include natural language processing and computer software.

Prof. Wang is a member of China Computer Federation.

© 2010 ACADEMY PUBLISHER