Author Model and Negative Feedback Methods on TREC 2011 Microblog Track

Rui Li[1,2], Bingjie Wei[1,2], Kai Lu[1,2], Bin Wang[1]
[1]Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China, 100190
[2]Graduate University of Chinese Academy of Sciences
Beijing, China, 100190
Email: {lirui, weibingjie, lukai, wangbin}@ict.ac.cn

Abstract

This paper gives an overview of our work (the ICTIR group) in microblog track of TREC 2011 for tweets retrieval. The basic query likelihood model with smoothing is the fundamental method in our approaches, we also consider other factors: the author information and the negative feedback. Firstly, we classify all queries into three categories, construct refined feedback in different ways to reform them; Secondly, extremely short tweets lead to poor clustering performance, the author topic models are trained for tweets expansion and smoothing. Finally, we train negative feedback model to reduce noise impacts in our microblog search task. Experimental results show that our methods could improve the retrieval performance greatly.

Keywords: Microblog Retrieval; twitter; short text; topic modeling; negative feedback;

1 Introduction

Lots of people spend more and more time on microblog in the daily lives, many of us obviously like to use this convenient way to share our lives and communicate with our friends. The dataset published in the microblog track is from twitter.com. Microblog is different from traditional web pages, social

media and social networks. For example, in twitter, the tweets that user commonly write are extremely short, the average length is about 11. That is to say, if we use the traditional methods and technique in information retrieval directly, it would be suffered from severe problem of sparse. Moreover, microblogs usually contain some interesting features. You can "mention" other users in your tweets by adding their user names. "Retweet" are also supported to simply reproduce other users' tweets you like on your own page. User can add URLs in their tweets, which are then redirected by twitter.com. Hashtag, as a kind of tag, can be regarded as topics that user might be interested in and discussed in the tweets. That means user might tag the tweets himself by using hashtags. How to incorporate all these features to improve our retrieval results is a considerable problem. In addition, informal texts are found in vast majority of tweets. Tweets in twitter.com are filled with abbreviation, deformation, even emotions. It brings great natural language processing problems.

The microblog track has a defined task: to improve p@30 performance for all retrieval results of all the given queries. Each query is represented as "topics" that contains query id as primary key, query submission time, query content and the last tweet id before the query submitted. "Interesting" but "newer" relevant tweets should be ranked higher in rank list.

In our work, we first reform the query by feedback words which are gathered and extracted by several ways. Then, all tweets published by the same author are collected together to form a new dataset. After that, We could train topic models of authors/users [1]. At the last, we train negative feedback models with different parameters for different kinds of queries.

The rest of the paper is organized as follows: Section 2 shows our preparation for retrieval task and the baseline we use to get the initial retrieval sets. The main author model we used is presented in section 3. Section 4 introduces the negative feedback model. Section 5 describe our experiments, including the parameters we use and the details of the experimental results, and finally our conclusions and future work are given in section 6.

2 Preparation

The microblog dataset, released by TREC 2011, needs to be preprocessed before we use it. That is because it contains many informal and non-English texts. Furthermore, the "mention", "retweet", hashtag and URL in tweets

should be processed before we index them. The main steps are as follows:

1. If the tweet starts with "RT" or the tweet status code equals to a specific number, it is a reproduced tweet from another and is repetitive. In that case, we just ignore this tweet. If the tweet contains "RT", we only keep the words before "RT".

2. User's name that tweets mentioned, hashtags and URLs that tweets contained are all extracted as useful features.

3. Emotions and stop words are filtered.

4. Some transformational lexicons are restored, such as "goooooood!!!" changed by "good".

5. Tweets written in non-English language are filtered.

6. Porter stemmer is used for stemming.

2.1 Baseline

Because some models which are used to rerank the tweets have high computation complexity, we only sort a subset of the corpus. All the queries are used to run a simply retrieval model (bm25) to get the initial result sets. Each query contributes about top 10, 000 relevant tweets at most, we collect 471, 830 tweets (50 queries) in total at the end.

First, we collect feedback documents for each search topic returned by search engines; There are some restrictions when we search, such as we only retrieve the documents whose released time is closed to the query submission time (but before 2011.2.8). The number of the feedback documents is about 10∼20. Second, key words are extracted automatically (TextRank), then participants are invited to choose manually about 10 key words for each query; In details, all the key words are selected by 3 people respectively. Third, the baseline result sets are obtained by running a simple search task with the expanded queries. The rank list for each query retrieved by probability model (BM25) and kl divergence (KL) model.

2.2 Author Information Extraction

The baseline is a common retrieval method. As we mentioned that, it suffers from serious noise problem caused by the short text and irregular expression of words. The tweets are presented arbitrary and the average length is about 11. For example, a tweet which talks about "world cup" should be considered

relevant to the query "soccer fifa", although there are no common words between them.

In our dataset, we find that, the twitterer always publish more than several tweets that are all have similar meaning. If a user is interested in a specific topic, he may write or retweet many relevant messages on his microblogs. For this basic reason, we can estimate the author's model and then use it to improve the recall rate. The users who published more than five tweets, are considered as containing useful information. To estimate the author's model, we considered all tweets issued by the same author (user) as one document. There are totally about 3.6 million authors in our dataset. We then could get a distribution over words for each author. Recently, topic models are found especially useful to measure documents' semantic relations. In topic model, topics are distributions on words. Similarities between author and query can be computed by the distance (query likelihood, KL divergence and so on) between their distributions over words and topics. So in our method, we train topic models for each author, and then compute their ranking scores on given query by the author model in section 3.

2.3 Feedback Construction

According to paper [13], queries can be classified as: celebrity, social event and common queries. For different types of queries, we adopt different expansion strategies to reconstruct it. For celebrity query, its motivation is mainly to find the breaking news about a particular person or a public institution (such as "Oprah Winfrey" and "White Stripes"), rather than to learn more about a particular aspect of that person. So the feedback documents and key words we choose for celebrity query expansion contain more "breaking news" words. These words can represent the event's different part and something the searcher wants to find, such as: people's name in the event, when and where the event broke out and so on. We achieve this by selecting key words by experienced participants and adding them into initial query. For social event query, its motivation is different from the celebrity query's, what to find has been to a certain extent settled and the scope is narrower. The feedback documents could contain more kinds of statements or comments, and synonym words could be expanded. For the third type query, which is searching for specific topics, for example: "organic farming requirements"? Even we find that it occupies a small percentage in all queries, we do query expansion by extracting useful information from wikipedia and wordnet. Examples of

Table 1: The words selected from relevant documents

Toyota Recall	Mexico drug war	Thorpe return in 2012 Olympics
lexus	cartels	comeback
safety	violence	London
pedal	border	swimming
tundra	police	Ian Thorpe
fuel	tijuana	gold medal
pipe	fight	world
crack	traffick	championship
leak	conflict	welcome
avensis	government	Phelps
defect	calderon	australia

selected words can be seen in table 1.

3 The Author Model

With the feedback documents and expanded queries we constructed before, we introduce a retrieval model that could integrate the author information in this section. Because there is no user profile in the corpus, we just compute the similarities between authors' tweets and queries, the similarity can also be called the author's ranking score. It contains two parts: model of tweets and model of topics. Latent Dirichlet Allocation (LDA) are used to train the author's topic model. The score then can be computed according to the query likelihood retrieval model. All of these can be seen in formula $1 \sim 4$:

$$S(A,Q) = (1-\lambda)S_{tweet}(A,Q) + \lambda S_{topic}(A,Q) \tag{1}$$

$$\begin{aligned} S_{tweet}(A,Q) &= \frac{1}{|A|}\sum_{t_w \in A} S(t_w, q) \\ &= \frac{1}{|A|}\sum_{t_w \in A}\sum_{w \in V} C(w,q)\log p(w|t_w) \end{aligned} \tag{2}$$

$$S_{topic}(A, Q) = \sum_{i=1}^{K} p(\theta_i|A) S_\theta(Q, \theta_i) \tag{3}$$

$$S_\theta(Q, \theta_i) = \sum_{w \in V} C(w, q) \log p(w|\theta_i) \tag{4}$$

Where S denotes the ranking scores, A denotes author, $|A|$ denotes the number of author's tweets , Q is the query, t_w is the tweet, K is the topic number, $C(w, q)$ is the count of word w in query q; S_{tweet}, S_{topic} and S_θ mean the author's scores on tweets, the author's scores on topics, and the scores of each topic respectively. As we see, language modeling approaches are used by us to compute the ranking scores. One things need to notice is $p(w|t_w)$ in formula 2 are estimated using Dirichlet Prior smoothing (DIR) and Jelinek-Mercer smoothing (JM) for comparison. The parameters after selected are 8∼20 (DIR) and 0.5 (JM). The author model is then used in our experiment as a smoothing model (the smoothing parameter is 0.2) for tweet expansion.

4 Negative Feedback

We use negative feedback to improve the ranking effects. The main reason is that the model we used before is mainly used for common query retrieval tasks. However, in our dataset, some of queries are difficult. That means if the query is processed by the same method used before, the result will not so good. One of the reason is that the queries or the feedback documents contain noise words which could influence the results greatly. One query example is "Kubica crash", the expanded query after feedback contains the word "burn". According to this, in the result we find that lots of tweets that contain "crash crash burn" are recalled. Unfortunately, "crash crash burn" is the lyrics of a popular song, with a large number of tweets related to it. Thus, the relevant tweets are crowded out from the retrieval list. In this section, we represent our negative feedback method to cope with this problem.

After ranking methods we mentioned in section 3, we then collect negative feedback document sets for each query. The negative feedback model is trained from the non-relevant tweets, the corresponding negative feedback score can be computed as formula 5:

Table 2: Average P@30 and MAP

	BL	BL+FB	AT+FB	AT+FB+NFB
P@30	0.2025	0.2654	0.3986	0.4075
MAP	0.1233	0.1632	0.2469	0.2986

$$S(D,Q) = \sum_{w \in V} [p(w|\theta_q) - \alpha p(w|\theta_N)] \log p(w|t_w) \tag{5}$$

Where θ_N is the negative feedback model, $p(w|\theta_N)$ is the non-relevant words probabilities. The negative feedback model could punish the words in non-relevant tweets. Notice that $p(w|\theta_N)$ is estimated by the partly selected words from non-relevant tweets but all the non-relevant tweets to avoid excessive negative feedback. So our methods could take advantage of the author model for tweets, the feedback model and negative feedback model for queries.

5 Experiment and Analysis

Consistent with the official method, our main evaluation indicators are P@30 and MAP. The experiments are designed to answer two questions: How useful the author's information is in the retrieval task? How does the noise problem could be alleviated or solved?

5.1 Author Information

In our experiment, there are 3, 673, 968 authors in total, and we just keep the author who publish no less than 5 tweets to train topic model. That is, the users who published more than five tweets are thought as containing useful information. The toolkit we used to run our baseline method is lemur. The comparison of author model and baseline method can be seen at table 2, more detailed information can be seen in Table 3. Where BL is the baseline method, FB denotes feedback, NFB denotes the negative feedback, and AT means the author topic model.

Table 3: Average P@30, MAP and R-prec over all submitted runs(under the two relevance criteria)

	All			High		
Runs	P@30	MAP	R-prec	P@30	MAP	R-prec
baseline	0.0769	0.0722	0.0959	0.0182	0.0351	0.0271
run1	0.3823	0.1747	0.2352	0.1202	0.1043	0.1323
run1fix	0.3986	0.2469	0.3019	0.1354	0.2353	0.2517
run2	0.4075	0.2986	0.3571	0.1414	0.2598	0.2622

In table 2, we see that the author model works well, both the baseline and author model improves greatly after feedback. This means the author's information is useful, the improvement might come from two parts: First, the author's other tweets are used to expand the tweets longer, to some extent work around the sparse issue. Second, adding author's topic aims at dealing with the words mismatch problem, some semantically related tweets maybe written by same author.

5.2 Negative Feedback

Non-relevant tweets labeled by 3 group participants (for training, cross validation and test dataset) in the rank list are used to predict the query difficulty. Each query's top 15 tweets in retrieval list are labeled. In our experiment, if the number of non-relevant tweets is no less than 5, 9 and 12, we considered the query as a common query, difficult query and highly difficult query respectively. After that, we train the negative feedback model we described before, and for the three kinds of queries we trained different parameters respectively. In table 3, this baseline run we submitted is different from "BL" we mentioned before, because at that time there is no labeled data for training parameters, the run1 and run1fix in our submitted runs both make use of author model and relevance feedback, but with different parameters. Run2 have considered the negative feedback method we described before. So in table 2 and 3, we could see that, negative feedback could improve the result we retrieved before. That means the tweets suffers from too severe noise problem to make a simple query expansion directly. Our negative feedback method works better than other methods we have

implemented.

6 Conclusion and Future Work

As some papers mentioned [3] [4] [5], there are some features we could use to improve the performance of retrieval: user information, hashtag, URL. Need to say, user profile and user's friends are both extraordinary useful information. We have not use these features, partly because the dataset released does not include them.

However, more importantly, the word co-occurrence in tweets is still extremely sparse; Sparsity and noise should be solved by using better methods. Moreover, it is reasonable to believe that several kinds of external resources could enhance the effectiveness of retrieval model. Our future work will lay emphasis on training receiver's model and using external resources to improve the retrieval effects; The structure of microblog is considerable contents that is worthy of research.

References

[1] Daniel Ramage, Susan Dumais, and Dan Liebling, "Characterizing microblogs with topic models", ICWSM, 2010.

[2] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas, "Short Text Classification in Twitter to Improve Information Filtering", Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010.

[3] Rinkesh Nagmoti, Ankur Teredesai, and Martine De Cock, "Ranking Approaches for Microblog Search," 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 1, pp.153-157, 2010.

[4] Kamran Massoudi, Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp, "Incorporating Query Expansion and Quality Indicators in Searching Microblog Posts", ECIR, pp.19-21, April 2011.

[5] Miles Efron, "Hashtag Retrieval in a Microblogging Environment", Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010.

[6] A. Moore, "Statistical data mining tutorials," Ph.D. dissertation, CMU, Pittsburgh, May 2000. [Online]. Available: http://www.autonlab.org/tutorials/

[7] Liangjie Hong, and Brian D. Davison, "Empirical Study of Topic Modeling in Twitter", 1st Workshop on Social Media Analytics (SOMA 10), July 25, 2010, Washington, DC, USA. ACM.

[8] Wouter Weerkamp, Krisztian Balog, and Maarten de Rijke, "A Generative Blog Post Retrieval Model that Uses Query Expansion based on External Collections", Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language, 2009.

[9] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete, "Information Credibility on Twitter", Proceedings of the 20th international conference on World wide web, March 28CApril 1, 2011.

[10] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou and Heung-Yeung Shum, "An Empirical Study on Learning to Rank of Tweets", Proceedings of the 23rd International Conference on Computational Linguistics, 2010.

[11] Ravali Pochampally, Vasudeva Varma, "User context as a source of topic retrieval in twitter", Proceeding of the 34rd international ACM SIGIR conference on Research and development in information retrieval, 2011.

[12] Gordon V. Cormack, Jos Mara Gmez, Hidalgo, Enrique Puertas Snz, "Spam filtering for short messages", Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 2007.

[13] Jaime Teevan, Daniel Ramage, and Merredith Ringel Morris, "TwitterSearch: a comparison of microblog search and web search", Proceedings of the fourth ACM international conference on Web search, pp. 35-44, 2011.