HLJIT at TREC 2016: The Approaches Based on Document Language Model for Real-Time Summarization Track

Song Li[1,2], Zhenyuan Hao[1], Zhongyuan Han[1,*], Leilei Kong[2,1], Haoliang Qi[1]
hanzhongyuan@gmail.com
[1] *School of Computer Science and Technology, Heilongjiang Institute of Technology, Harbin, China;*
[2] *College of Information and Communication Engineering, Harbin Engineering University, Harbin, China*

Abstract

The paper describes the technology of *HLJIT* for TREC 2016 Real-Time Summarization Track for microblog. Three summarization approaches under the language model framework, the traditional language model, the temporal document language model and the hyperlink-extended language model, are proposed.

1 Introduction

Microblog is one of the most powerful online communications media for people to learn what is happening around the world today. After a hotspot happens, people post microblogs to disseminate information in real time and express their views by using limited words quickly and freely. However, in the microblog search system and information filtering system, the information users obtained has a lot of repetition, which makes difficult for users to follow up the latest development of event. Microblog real-time summarization system is designed to solve the problem of information redundancy, and pushes the latest developments information to the user in real-time.

Addressing on the issue of real-time summarization, TREC presents the Real-Time Summarization Track in 2016 which contains two scenarios: push notifications (Scenario A) and email digest (Scenario B). Scenario A requires pushing relevant microblogs in real-time, and the pushed microblogs should not say the same thing. Scenario B identifies a batch of up to 100 ranked tweets per day for per interest profile, it is expected that the systems have the abilities to compute the results in a relatively short time after the day ends on the condition of without the future evidence.

Under the language model framework, three real-time summarization methods are presented in this paper. For scenario A, we adopt temporal filtering strategy to determine the relevance between the topic and the current microblog, if the microblog passed the novelty verification, it will be pushed. For scenario B, the microblogs, which come from the top 100 candidate relevant microblogs as the relevant microblogs, through the novelty verification are

submitted.

2 Real-Time Summarization Track Framework

Pushing the non-redundant microblogs in real-time is the task for the Real-Time Summarization Track. This section introduces a framework of a real-time summary system, and then gives the detailed explanations for each part in the proposed framework.

For the user interest files, the framework uses a preprocessing process to calculate the similarity between the user interest and the microblogs according to the interest topic as soon as the system getting the user interest files.

When the new microblogs arrived, the following processes are implemented step by step. Firstly, we do some preprocessing operation on each microblog. The *Preprocess* is described in 2.1 in detail. Secondly, the *Fast Filter* chooses the relevant microblogs candidate. Thirdly, the *Relevance Estimation* calculates the relevance between the topic and the microblog and saves the results in the *Candidate Relevant Tweet Pool*. Fourthly, the relevance is verified by the *Relevance Verification*. Fifthly, the *Novelty Verification* checks the redundancy of the microblogs. Finally, the microblog is pushed to the RTS evaluation broker, and saved it to the rushed tweet pool.

The difference between the scenario A and scenario B is that, after the *Relevance Ranking* ranking the candidate relevant microblogs in *Candidate Tweet Pool* of the day that the microblog posted, the Top 100 candidate Tweets will be selected as the relevant microblogs without relevance verification.

Fig 1 shows a framework of the proposed Real-Time Summarization Track.

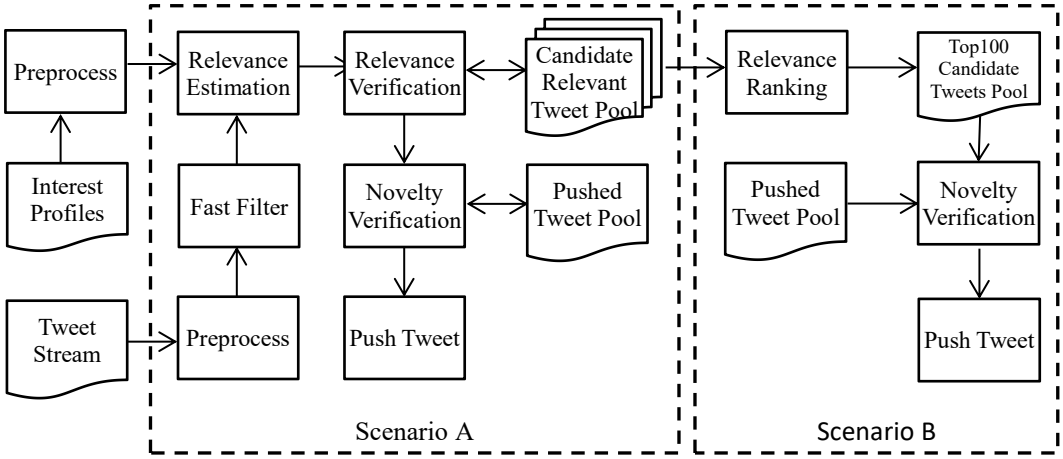

Fig 1. Real-Time Summarization Track System Framework

2.1 Preprocess

In the preprocessing, the topic and the tweet stream are preprocessed as follows:

1) Removing the tweets that are marked as the Non-English tweets.

2) Filtering the Non-English characters.

3) Stop words are removed. The other words are stemmed using the Porter Algorithm.

2.2 Fast Filter

To reduce the calculation time, the *Fast Filter* removes the microblogs that do not contain any keywords which occurs in the user profile. And then if the original tweet of the retweet M1 or the retweet that has the same original tweet had been pushed, the microblog M1 will be discarded.

2.3 Relevance Verification

Relevance Verification aims to determine whether the microblog is related to the topic or not. We assume that the similarity scores of the relevant microblogs are higher than those of the irrelevant ones. Therefore, the score of the n-th microblog is an ideal threshold if n is the number of relevant microblogs. The idea comes from Ref. [1] and [2]. Because there are no any feedbacks, we set n as a constant.

2.4 Novelty Verification

The Novelty Verification is exploited to remove the microblogs that talking the same things with the pushed microblogs. The Vector Space Model(VSM) is employed to achieve the redundancy degree between the two microblogs. If the similarity between the new microblog and any microblog in pushed pool is greater than 0.5, the new microblog is regarded as a redundant one. The VSM is showed as follows:

$$VSM(\vec{T},\vec{D}) = \frac{\vec{T} \bullet \vec{D}}{\|\vec{T}\| \times \|\vec{D}\|} \tag{1}$$

where $\vec{\mathrm{T}}$ and $\vec{\mathrm{D}}$ are the vector of the topic and the microblog document.

3 Relevance Estimation

In the framework depicted in Fig 1, the component *Relevance Estimation* is the most important kernel. The language model for information retrieval is employed to estimate the relevance between the uses interest profile (represented by a topic) and the micorblogs.

In the classic language model framework, query and document are modeled as query model θ_Q and document model θ_D. Specifically, the Kullback-Leibler divergence is used to measure the difference between θ_Q and θ_D as follows[3]:

$$\mathrm{KL}(\theta_Q,\theta_D) = \sum_{w \subset V} \mathrm{P}(w \mid \theta_Q) \log \frac{\mathrm{P}(w \mid \theta_Q)}{\mathrm{P}(w \mid \theta_D)} \tag{2}$$

where V is the vocabulary, w is the word in V. $P(w|\theta_Q)$ and $P(w|\theta_D)$ are the word w's distribution in query model θ_Q and document model θ_D. θ_Q and θ_D are usually constructed as the uni-gram language model.

Query model θ_Q is regularly estimated by maximum likelihood estimate method as follows:

$$P(\mathrm{w} \mid \theta_{\mathrm{Q}}) = \frac{c(w,Q)}{|Q|} \tag{3}$$

where $c(w,Q)$ is the term frequency of w in query Q, and $|Q|$ is the total number of words in query Q.

Three document language models are selected to build the document model.

3.1 Document Language Model

$P(w|\theta_D)$ is estimated by maximum likelihood estimate and smoothing technology, such as Dirichlet smooth, which is used to address the zero probability problem[4]. By applying Dirichlet smoothing method, $P(w|\theta_D)$ can be described as:

$$\mathrm{P}(\mathrm{w} \mid \theta_{\mathrm{D}}) = \frac{\mathrm{c}(w,\mathrm{D}) + \mu \mathrm{P}(w \mid C)}{|\mathrm{D}| + \mu} \tag{4}$$

3.2 Temporal Document Language Model

The relevant microblogs are often posted in a certain period centrally, the program uses current m history microblogs as the smooth data, and adopts two-stage smooth method to estimate the microblog model:

$$P(\mathrm{w} \mid \theta_{\mathrm{D}}) = \frac{\mathrm{c}(w,\mathrm{D}) + \mu P(w \mid C_t)}{|\mathrm{D}| + \mu} + \lambda P(w \mid C) \tag{5}$$

C_t is the set of recent m history microblogs, $P(w|C_t)$ is the probability of word w in C_t.

3.3 Hyperlink-Extended Language Model

In microblog retrieval, the content linked by URLs is one of the most important information of a microblog. We present a Hyperlink-extended model[5] for microblog retrieval that combines content of microblogs and the content of embedded hyperlinks web-pages using a probabilistic ranking function based on language model.

$$P(\mathrm{w} \mid \theta_{\mathrm{Du}}) = \frac{\mathrm{c}(w,\mathrm{D_u}) + \mu P(w \mid C_{\mathrm{html}})}{|\mathrm{D_u}| + \mu} \tag{6}$$

For Scenario A, we do the relevance verification using $\mathrm{KL}(\theta_{\mathrm{Q}} \| \theta_{\mathrm{Du}})$ and $\mathrm{KL}(\theta_{\mathrm{Q}} \| \theta_{\mathrm{D}})$ respectively, following the work in [6].

For Scenario B, we do the relevance verification as

$$\mathrm{P}(D \mid Q) \propto -\mathrm{P}(Q_m \mid Q) KL(\theta_{Q_m}, \theta_{D_m}) - \mathrm{P}(Q_u \mid Q) KL(\theta_{Q_u}, \theta_{D_u}) + \frac{\ln P(D_u)}{|\mathrm{Q}|} \tag{18}$$

where the first part is the product of $P(Q_m|Q)$ and KL divergence of microblog text D_m and query Q_m , the second is the product of $P(Q_u|Q)$ and KL divergence of linked document text

D_u and query Q_u, and the last part can be understand to represent a confidence levels of hyper-linked document. There are two ways to estimate $P(D_u)$: if the microblog D contains links and we can get the linked pages content, we set $P(D_u)=1$; otherwise, we give it a very small value. The $P(Q_m|Q)$ and $P(Q_u|Q)$ is simply set as 1.

4 Results

We submit three runs for each scenario: traditional Language Model(LM), Temporal Document Language Model (TDLM), the Hyperlink-Extended Language Model (HELM).

The parameters are set following Ref. [6]. In detail, the run LM adopts the smooth parameter μ=100; the run TDLM sets the parameters μ=100 and λ=0.5, and the run HELM adopts the smooth parameters μ=100 in $P(w|\theta_D)$ and μ=3000 in $P(w|\theta_{Du})$. We set the microblog relevance verification parameter n =10 and the HTML relevance verification parameter n=300 in all runs. The performances of the submitted runs for scenario A and B are shown in Table 1 and Table 2.

Table 1. Performance of the submitted runs for scenario A

model	EG-1	EG-0	nCG-1	nCG-0	GMP0.33	GMP0.5	GMP0.66
LM	0.2085	0.0246	0.2018	0.0178	-0.4070	-0.2929	-0.1854
TDLM	0.2276	0.0383	0.2283	0.0390	-0.3698	-0.2576	-0.1520
HELM	0.1752	0.0109	0.1788	0.0145	-0.3256	-0.2357	-0.1511

Table 2. Performance of submitted runs for scenario B

model	nDCG-1	nDCG-0
LM	0.1155	0.1155
TDLM	0.1145	0.1145
HELM	0.0638	0.0638

5 Conclusions

In this paper, the systems for TREC 2016 Real-Time Summarization Track are introduced. A Real-Time Summarization Track System Framework is proposed. The traditional Document Language Model, the Temporal Document Language Model and Hyperlink-Extended Language Model are utilized to estimate the document model. The experimental results of TREC 2016 Real-Time Summarization Track are reported. Overall, the approaches still need further improvement for a perfect solution to tweet document modeling.

Acknowledgment

This work is supported by Research Project of Heilongjiang Provincial Department of Education (No. 12541677).

References

[1] Han Zhongyuan, Yang Muyun, Kong Leilei, Qi Haoliang, Li Sheng. A Temporal Microblog Filtering Model.

International Journal of Grid and Distributed Computing, 2016, 9(1): 89-98.

[2] Han Zhongyuan, Yang Muyun, Kong Leilei, Qi Haoliang, Li Sheng. A Hybrid Model to Real-time Microblog Filtering. Chinese Journal of Electronics. 2016, 25(3):432-440.

[3] Lafferty J. and Zhai C. Document language models, query models, and risk minimization for information retrieval. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. 2011:111-119.

[4] C. Zhai, J. Lafferty. A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems. 2004, 22(2):179-214.

[5] Han Zhongyuan, Yang Muyun, Kong Leilei, Qi Haoliang, Li Sheng. A Hyperlink-extended language model for microblog retrieval. International Journal of Database Theory and Application. 2015, 8(6):89-100.

[6] Han Zhongyuan, Li Xuwei, Yang Muyun, Qi Haoliang, Li Sheng, Zhao Tiejun. Hit at TREC 2012 microblog track. Proceedings of Text REtrieval Conference. 2012, 3.