HLJIT at TREC 2017 Real-Time Summarization

Zhongyuan Han[1,*], Song Li[1,2], Leilei Kong[1], Liuyang Tian[1,2], Haoliang Qi[1]

hanzhongyuan@gmail.com

[1]School of Computer Science and Technology, Heilongjiang Institute of Technology
Harbin, Heilongjiang, China, 150050

[2]School of Computer Science and Technology, Harbin Engineering University
Harbin, Heilongjiang, China, 150001

Abstract

This paper describes the approaches used at the TREC 2017 Real-Time Summarization. This task contains two scenarios: push notifications and email digest. For the scenario of push notifications, three filtering models, which are based on the hyperlink-extended retrieval model, the Learning to Rank and the hybrid filtering model, are proposed to filter the relevant tweets for a given topic. A novelty verification method is given for further filter the tweets for push notification. For the scenario of email digest, three ranking models, the hyperlink-extended retrieval model, the retrieval model based on learning to rank, and the personal retrieval model, are presented to rank the relevant tweets. Similarly, a novelty verification is proposed for filtering the redundant tweets. The evaluation results of TREC 2017 Real-Time Summarization show that the performance of our models is competitive.

1. Introduction

The evaluation of TREC 2017 Real-Time Summarization contains two scenarios: push notifications (Scenario A) and email digest (Scenario B). Scenario A requires pushing the relevant tweets that concentrating on the different aspects of one thing in real-time, while 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 of computing the results in a relatively short time after the day ends on the condition of not using the future evidences.

Focused on the problem of Real-Time Summarization, three filtering models based on retrieval models or online classification models are developed to decide the relevant tweets in Scenario A, while the Scenario B is viewed as a retrieval task and three different retrieval models were exploited for email digest.

This paper is organized as follows: Section 2 introduces our methods for Scenario A. Section 3 depicts the detailed methods applied in Scenario B. Section 4 reports experimental setting and results. And Section 5 gives the conclusion.

2. Scenario A: Push Notifications

For the task of Push Notifications, we present three filtering models: the filtering model based on the hyperlink-extended retrieval model (denoted as HLJIT_testRun1_06), the filtering model based on Learning to Rank (denoted as HLJIT_testRun2_07) and the hybrid filtering model incorporating the retrieval model and the classification model (denoted as HLJIT_testRun3_08). The first two models exploit the idea of retrieval models and the third one adopts the idea of online classification model. The tweets within a certain period is firstly ranked by the proposed filtering model, then the sorted tweets are further filtered using a novelty verification method. Additionally, the assessment by the mobile assessors are only considered by the proposed hybrid filtering model. In this section, we first describe the filtering models, then introduce the method of novelty verification.

2.1 Filtering model based on the hyperlink-extended retrieval model

The first filtering model we adopt is the hyperlink-extended retrieval model. In microblog retrieval, the content linked by the URLs provided more important information for a microblog. The proposed hyperlink-extended model combines the content of microblogs and the embedded hyperlinks webpages using a ranking function based on language model[1].

Given a tweet D and a topic T, the relevant score is calculated as follow:

$$LM(T,D) = P(T_{text}|T)QL(\Theta_T,\Theta_{text}) + P(T_{URL}|T)QL(\Theta_T,\Theta_{URL}) \tag{1}$$

where $P(T_{text}|T)$ denotes the probability that the user's information needs might be satisfied by the microblog's text, while $P(T_{URL}|T)$ denotes the probability that the user's information needs are more likely to meet by the hyperlink documents. Then the similarity of the language model of topic and the language model of document is estimated by using the following equation:

$$QL(\Theta_T,\Theta_D) = \prod_{i=1}^{n} P(q_i \mid \Theta_D)^{P(q_i|\Theta_T)} \tag{2}$$

where Θ_T and Θ_D are language model of topic and document respectively, q_i *is the word in topic.*

The language model of topic Θ_T is estimated by integrating the title field and the description field in each topic as follows:

$$\Theta_T = P(w \mid T) = (1-\alpha)P(w|T_{\text{title}}) + \alpha P(w|T_{description}) \tag{3}$$

T_{title} and $T_{description}$ are the text in title field and description field respectively. $P(w|T_{title})$ and $P(w|T_{description})$ are the probabilities that the term w occurs within the title or the description by using the Maximum Likelihood Estimation.

The document model Θ_D is estimated using Maximum Likelihood Estimation and considers unseen words through probability smoothing by using Dirichlet smoothing method [2], shown in Eq. (4):

$$\Theta_D = P(w \mid D) = \frac{c(w;D) + \mu P(w \mid C)}{|D| + \mu} \tag{4}$$

where $c(w,D)$ is the term frequency of w in document D, $P(w|C)$ is the probability of w in corpus C, and $|D|$ is the total number of words in the document D.

2.2 Filtering model based on Learning to Rank

The second filtering model is performed based on learning to rank algorithm. The algorithm of ListNet[3] is adopted for ranking the tweets. We rely on the features of text similarity and the scores of language models for ListNet model. All features are presented in Table 1.

Table 1. Features used for learning to rank algorithm

1	$Jaccard(T_{\text{title}}, D_{\text{text}})$	11	$\text{Cos}(T_{\text{description}}, D_{\text{text}})$
2	$Jaccard(T_{\text{title}}, D_{URL})$	12	$\text{Cos}(T_{\text{description}}, D_{\text{URL}})$
3	$Jaccard(T_{\text{title}} \cup T_{\text{description}}, D_{\text{text}})$	13	$LM(T,D)$ where $\alpha=0$, $P(T_{URL}\|T)=0$
4	$Jaccard(T_{\text{title}} \cup T_{\text{description}}, D_{\text{URL}})$	14	$LM(T,D)$ where $\alpha=0$, $P(T_{URL}\|T)=1$
5	$Jaccard(T_{\text{description}}, D_{\text{text}})$	15	$LM(T,D)$ where $\alpha=0.4$, $P(T_{URL}\|T)=0$
6	$Jaccard(T_{\text{description}}, D_{URL})$	16	$LM(T,D)$ where $\alpha=0.4$, $P(T_{URL}\|T)=1$
7	$\text{Cos}(T_{\text{title}}, D_{\text{text}})$	17	$LM(T,D)$ where $\alpha=1$, $P(T_{URL}\|T)=0$
8	$\text{Cos}(T_{\text{title}}, D_{URL})$	18	$LM(T,D)$ where $\alpha=1$, $P(T_{URL}\|T)=1$
9	$\text{Cos}(0.6T_{title} + 0.4T_{descrition}, D_{text})$	19	A Tweet consists of URL for 1, otherwise for 0.
10	$\text{Cos}(0.6T_{title} + 0.4T_{descrition}, D_{URL})$		

where *Jaccard* (*T, D*) is the Jaccard coefficient of *T* and *D*, *Cos*(*T, D*) is the Cosine similarity of *T* and *D*, and *LM*(*T, D*) is the score defined in Eq.(1).

Then the evaluation metric MAP is selected to optimize on training data, and the Gradient Descent is used to update the parameters of model.

2.3 The hybrid filtering model

The third filtering model is a hybrid model adopted to estimate the relevance between the topic *T* and the Tweet *D*. The hybrid model incorporates the retrieval model (language model) and the classification model (logistic regression), and uses the retrieval model as prior knowledge to revise the hyper plane of classification. Specifically, we built a relevance estimation model $h_T(D)$ for each topic:

$$h_T(D) = \frac{e^{(1-\delta)\sum(wx+b)+\delta(LM(T,D)-\gamma)}}{1+e^{(1-\delta)\sum(wx+b)+\delta(LM(T,D)-\gamma)}} \tag{5}$$

where δ is a controlling parameter, x is the term vector of tweet *D*, w is the weight vector, *b* is the bias, *LM*(*T,D*) is the similarity score and *γ* is a threshold computed by Eq.(6):

$$\gamma(t) = find\_kth(NRM(t)) \tag{6}$$

where *NRM(t)* is the number of the relevant tweets at the time *t* and $find\_kth(\cdot)$ is a function returning the k-th max similarity score. We set the *k=100* in this task.

The tweet *D* will be judged as the relevant tweet if $h_T(D)>0.5$, and the relevant tweets accumulated within half a day are ranked according to the score of $h_T(D)$. The online filtering model is updated according to assessment by the assessors. The updating details were described in REF. [4].

2.4 Novelty Verification for Push Notifications

For guaranteeing the pushed tweets not talking about the same thing, we perform a novelty Verification.

Novelty Verification uses Cosine similarity to check the novelty. Specifically, the Cosine similarity is used to compare candidate tweets from the filtering models mentioned above sequentially with those in push pool.

For HLJIT_testRun1_06 and HLJIT_testRun2_07, only the first tweet having a similarity score lower than a certain threshold (0.7 is used in our method) is viewed as the valuable one. The novelty tweet at the top of the list will be pushed. If there is no any satisfied tweet, then the top1-ranked tweet will be pushed.

For HLJIT_testRun3_08, the push number K is set as half of N (the number of relevant tweets in the current list) if $N<10$. Otherwise, the K is set by the zoom logistic function which maps N into [1,10]. Then the Tweet will be pushed if it has passed the Novelty Verification.

3. Scenario B: Email Digest

We regard the task of Email Digest as a problem of relevant tweet retrieval and propose three models, the hyperlink-extended retrieval model (denoted as qFB_url), the model based on learning to rank (denoted as HLJIT_l2r), and the personal retrieval model (denoted as HLJIT_rank_svm), to rank the relevant tweets. Similarly, a novelty verification is operated on the ranking list.

3.1 The hyperlink-extended retrieval model (qFB_url)

A retrieval model based on hyperlink-extended model described in 2.1 is exploited to rank the relevant tweet. The difference is that we use the Relevance Model [5] for query language modeling. The 50 top-ranked feedback documents searched by Google search engine are used to query expansion. We select top 10 tweets posted in a day as the relevant ones and send them to the Novelty Verification in batch after the day ends. The model for a given topic is estimated by Eq. (7)

$$\Theta_{T_R} = P(w \mid T) = (1-\beta)P(w|T_{\text{title}}) + \beta P(w|T_R) \tag{7}$$

where $P(w|T_R)$ is estimated by Relevance Model described in Ref. [5].

3.2 The retrieval model based on Learning to Rank (HLJIT_l2r)

The learning to rank method based on ListNet is also used in the task of Email Digest. On the basis of the ranking model described in 2.2, two new features, $QL(\Theta_{T_R}, \Theta_{text})$ and $QL(\Theta_{T_R}, \Theta_{URL})$, are added to the proposed ranking model, where $QL()$ is the score of language model which taking the query expansion into account (described in 3.1)

Additionally, all the tweets that has been judged as relevant ones by the mobile RTS evaluation broker are sent to the novelty verification.

3.3 The personal retrieval model (HLJIT_rank_svm)

In this scheme, the assessment by the mobile assessors are regarded as a person's feedback, which represent the user's interest, and RankSVM[6] is used to learn the personal retrieval model. For each topic, we exploit the assessment by the mobile assessors to train a ranking model. The pairwise-based RankSVM is adopted as the learning algorithm, while the terms in tweets are selected as the features.

Additionally, all the tweets that has been judged as relevant ones by the mobile RTS evaluation broker are sent to the Novelty Verification.

3.4 Novelty Verification for Email Digest

The similarity estimation in Novelty Verification for Email Digest also adopts the Cosine similarity.

In qFB_url, tweets in the list ranked by Relevant Tweet Retrieval are checked by a novelty verification sequentially, all the tweets that go through the novelty verification are pushed to the RTS evaluation broker.

In HLJIT_l2r and HLJIT_rank_svm, we select *K* novel tweets sequentially and push them to the RTS evaluation broker. *K* is determined according to the assessment by the mobile assessors from the RTS evaluation broker on that day. If there are any relevant tweets in the assessment, K=10, otherwise, *K* is set a value in [3,5] according the number of the irrelevant tweets in the assessment.

4. Experiments

4.1 Data Set

We download 38,199,201 tweets by using the official API to listen the tweets stream. In these tweets, we get 29,255,621 effective tweets which contains 8,962,062 tweets written in English. Then, total 3,596,304 tweets are remained after the following processing. Firstly, the trash tweets are abandoned according to the following rules proposed in [7,8].

1) The number of ASIIC characters (0-128) is less than 80%;
2) The length of text is less than 20 (characters);
3) The number of HashTags is more than 4;
4) Non-English characters are more than 35%.

Secondly, further preprocessing operation are performed according to the following rules.

1) Only the tweets which contain at least one word in the topic title field is selected;
2) RT tag, user_mentions and stop words are removed from tweet text;
3) Porter stemming are used.
4) The webpage of the URL is downloaded;

4.2 Parameters setting

All the parameters used in the proposed models are showed in Table 2.

Table 2. Parameters setting

	Run Id	Model	Parameters
Scenario A	testRun1	LM	μ=100, P(TURL|T)=0.1, α=0.5, period=1 day
	testRun2	LISTNET	Learning rate=0.01, the number of epochs to train=100, the number of hidden layers=1, the number of hidden nodes per layer=10, Metric to optimize on the training data=MAP, period=1 day
	testRun3	LR	γ=0.5 The number of epochs to train=50 Learning rate= 0.005, period=half day
Scenario B	qFB_url	LM-FB	μ=100, P(TURL|T)=0.2, β=0.7, fbTermNum=20
	HLJIT_l2r	LISTNET	Same as testRun2
	HLJIT_rank_svm	RANK SVM	c=2

4.3 Experimental Results

Table 3 shows the experimental results of scenario A runs by the mobile assessors, Table 4 shows the experimental results of scenario A runs by NIST assessors, Table 5 shows the results of scenario B runs by NIST assessors. In the table 5, the language model with original query and tweet(LM), the language model with query expansion and original tweet(qFB), and the hyperlink-extended model with original query (URL) are reported to show the effect of query expansion and document expansion with hyperlink-extended.

Table 3. Evaluation of scenario A runs by the mobile assessors.

run	relevant	redundant	Not relevant	unjudged
HLJIT-testRun1-06	847	173	1479	153
HLJIT-testRun2-07	1018	178	1494	106
HLJIT-testRun3-08	1027	196	1694	168

Table 4. Evaluation of scenario A runs by NIST assessors

runtag	EGp	EG1	nCGp	nCG1
HLJIT-testRun1-06	0.3318	0.1811	0.261	0.1102
HLJIT-testRun2-07	0.363	0.2088	0.2808	0.1266
HLJIT-testRun3-08	0.2426	0.1832	0.242	0.1826

Table 5. Evaluation of scenario B runs by NIST assessors

runtag	nDCGp	nDCG1
LM	0.2906	0.2289
qFB	0.3267	0.2651
URL	0.3283	0.2725
qFB_url	0.3501	0.291
HLJIT_rank_svm	0.2697	0.2376
HLJIT_l2r	0.3107	0.2778

From table 3, it can be seen that the HLJIT-testRun3-08 found the most relevant microblog, but the number of irrelevant microblogs is much more than HLJIT-testRun2-07. Relatively, HLJIT-testRun2-07 is better. The result of table 4 also cites this. The one-side feedback makes the low number of the relevant documents returned by mobile assessors. This may be the reason that the HLJIT-testRun3-08 does not fully play its role.

As can be seen from table 5, the qFB_url has the highest score, which indicates that query expansion and URL information alleviate the problem of short text matching and achieve better performance. The HLJIT_rank_svm score is the lowest, which may be due to the fact that the number of relevant tweets in the training set(returned by the mobile assessors) is too small to learn an efficient ranking model.

5. Conclusion

In this paper, we have introduced the key aspects of the proposed models for TREC 2017 Real-Time Summarization task.

Three filtering models have been proposed for the scenarios of push notifications. The content linked by the URLs is attempted to estimate the similarity of the topic and the document. The model based on learning to rank is also considered in the proposed models. Combining with the novelty verification strategies, the model based on the ListNet achieved 0.363 on the main measure metrics EG-p (the highest EG-p score of the proposed three models). The online filtering model is also used for the scenarios of push notifications and the assessment by the mobile assessors are used to update the filtering model.

For the scenarios of email digest, we deem it as a retrieval task and three ranking-based methods are attempted. The model based on hyperlink-extended retrieval model achieved the highest nDCG@10-p for the richer extended content.

From the experimental results, it is obvious that vocabulary mismatch is the main problem for the short query and short document. Query expansion and hyperlink-extended model have the obvious effect on improving the performance and achieved the best results. Although the hybrid filtering model and the personal retrieval model did not get good results in the evaluation, we still believe that they have a certain potential. The future work will be further explored how to use the feedback in these two aspects.

Acknowledgment

This work is supported by the Social Science Foundation of Heilongjiang Province (No.16XWB02).

References

[1] Z Han, M Yang, L Kong, H Qi, S Li. A hyperlink-extended language model for microblog retrieval[J]. International Journal of Database Theory and Application. 2015, 8(6):89-100.

[2] MacKay D J C, Peto L C B. A hierarchical Dirichlet language model[J]. Natural language engineering, 1995, 1(3): 289-308.

[3] Z. Cao, T. Qin, T.Y. Liu, M. Tsai and H. Li. Learning to Rank: From Pairwise Approach to Listwise Approach[C]. ICML 2007.

[4] Z Han, M Yang, L Kong, H Qi, S Li. A Hybrid Model for Microblog Real-time Filtering[J]. Chinese Journal of Electronics. 2016, 25(3):432-440.

[5] V. Lavrenko and W. B. Croft. Relevance-based language models. Proceedings of the 24 th annual international ACM SIGIR conference, 120-127, 2001

[6] Joachims T. Training linear SVMs in linear time[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2006:217-226.

[7] Luchen Tan, Adam Roegiest and Charles L.A. Clarke. University of Waterloo at TREC 2015 Microblog Track. In the Twenty-Fourth Text REtrieval Conference (TREC 2015) Proceedings. NIST, 2015

[8] H Tan, D Luo, W Li. PolyU at TREC 2016 Real-time Summarization. In the Twenty-Fifth Text REtrieval Conference (TREC 2016) Proceedings. NIST, 2016