Fast NLP-based Pattern Matching in Real Time Tweet Recommendation

Zheng Gao, John Wolohan
School of Informatics, Computing, and Engineering
Indiana Unviersity Bloomington
gao27@indiana.edu, jwolohan@indiana.edu

***Abstract*—Social media users are willing to obtain information from online social streaming services. Everyday people receive news notifications from mobile devices and figure out information which are new and interesting to them. Therefore, it is necessary to learn a recommendation mechanism to see how to attract users attention most by providing most useful news or information to them. This year, TREC (Text REtrieval Conference) offers a Real Time Summarization track to explore online user reading preference on Twitter, one of the largest social media platform so far, to figure out recommendation patterns best suitable for users.**

***Keywords*—real-time summarization, social media recommendation, ad-hoc information retrieval**

I. Introduction

With the exploration of social media, everyday people are facing with numerous news from all kinds of online streaming services. It causes a huge problem for online users to extract the real needed news as they are drowned in huge amount of irrelevant noisy news. Therefore, it is necessary to come up with a model that can help to filter out useful news in a quickly and accurate manner. Moreover, as information updates so fast, time issue also needs to be considered. Yesterday's hot new may be no longer interested by users today, such as sports news; duplicate news are also not trigger users' interest because they have already got the information. Considering these, understanding how to make real time recommendations to users accurately has huge potentiality and still needs a long way to go.

In order to copy with this challenge, since 2016, TREC conference merged previous the Microblog (MB) track, which ran from 2010 to 2015, and the Temporal Summarization (TS) track, which ran from 2013 to 2015 together to form a new track named as 'Real Time Summarization' (RTS). The overall goal for this merged track is to find the most interesting tweets for users in a timely manner. In this year's RTS track, there are two scenarios in total including Scenario A (push notifications) and Scenario B (email digest) [7]:

- **Scenario A (push notifications):** Tweets which are regarded as relevant and novel to users' interest profile will be pushed to the user as notifications in a timely manner via TREC RTS evaluation broker via a REST API. Then these notifications will be immediately routed to the mobile device of a group of human assessors. The human assessors can therefore judge the notifications accuracy as well as time efficiency. As users are not willing to receive too many notifications everyday, there is a upper bound limitation of the number of notifications.
- **Scenario B (email digest)):** In this scenario, given a user's interest profile, we need to generate a daily tweet recommendation list for him/ her. The list should be no more than 100 tweets per day and should be push to users in a short time after a day is over ideally. As the same as Scenario A, the accuracy of the generated daily tweet recommendation list is also based on the human assessors' judgment. We can consider previous tweets influence on users but can't involve in future tweets. i.e. when we generate recommendation tweet list, we can consider today's tweets but can never use statistics from tomorrow's tweets.

The whole tasks started from July 29 2017, UTC 00:00 to August 5 2017, UTC 23:59:59. During this period, we 'listened ' to the Twitter sample stream using the Twitter streaming API to get sampled real time posted tweets. We are also offered a batch of topics represented as users' interest profile. all our models are built based on these two datasets.

Our contribution of this track is threefold: First of all, we build up a NLP-based model to match streaming tweets with user's interest profile and recommend tweets back to users in real time; second, by leveraging information retrieval language models, we generate a daily tweet ranking list based on user's interest profile as well and recommend a daily tweet digest to users; third, the model we build up can be generalized to other social media dataset and applied to other online platforms as well.

The rest of the paper are organized in following structures: Section 2 reviews previous work in the RTS track last year; Section 3 explains the details of our models for both scenarios; Section 4 shows the evaluation result from human assessors via a bunch of cutting edge evaluation metrics; Section 5 points out the limitation of our current work and future improvement.

II. Related work

In order to construct better model for solving the track tasks, we look into previous work as well as combine our previous related work together to come up with novel strategies to cope with Tweet real time summarization.

As a tweet content is short and less than 140 words, one basic approach is tweet content expansion. [1] uses Google

retrieved search pages as external resources to enrich tweet content, which turns out a positive result for tweet recommendation. [3] also uses Google Search Engine retrieved pages as an external resource for query expansion. Moreover, it also consider the web links embedded in tweet content and use their source page content to expand tweet content as weel. In previous work for solving RTS tasks, [11] considers text categorization as well as text clustering via classifiers such as SVM and non-negative matrix factorization to minimize the error classification rate. And [13; 8; 4] also consider clustering structures as an external latent support for tweet content pattern recognition.

Besides query expansion, [2] uses JS-divergence to estimate relevance and redundancy between tweets and topics. It also applies a hybrid TF-IDF strategy to compute the salience score for tweets towards topics. [6] divides the whole process into an offline part and an online part. In offline part, it trains a relevance measurement model and a redundancy detection model. And the online part calculates 11 different features to represent both tweets and topics as the input for the model built up in offline part. [9] defines a set of filtering functions to filter out the most relevant and salient tweets towards a given topic. [12] however uses a KL divergence language model with both Jelinek-Mercer Smoothing and Dirichlet Smoothing as similarity algorithms as well as parameter tuning process to finalize its language model. [5] also builds up a set of features based on bag-of-word model and semantic structure of sentences. Afterwards it applies a Time-adjusted Dynamic Threshold-based model to generate recommendations for given topics. [10] leverages an idf -based term weighting scheme as well as Jaccard similarity method to work on lexical level of tweet contents first. Based on a time dependent evaluation function, it therefore filters out the most relevant tweets towards topics.

III. Methods

A. Scenario A

In Scenario A, it requires everyday participants can push at most 10 tweets for users given a topic due to users are not willing to receive too many notifications everyday. Hence, there are three challenges existed. One challenge is timeliness. Users are not willing to receive tweets after it posted more than a certain time cause their interest towards tweets will decay with time even though the tweets are very relevant to their interest topics. Another challenge is relevance calculation. As tweets are short documents with no more than 140 words. Simply applying language models can cause huge error because the short-length text may contain hidden concepts or multiple semantic meanings. For example, if a tweet contains the content "Wow Apple, niceTech", it may have a positive attitude towards Apple Inc instead of the apple fruit. Therefore, how to tackle the short-length text challenge is also worthy to think about. The third challenge is novelty. As users can only receive limited amount of notifications each day, they are not willing to be bothered by notifications with similar content.

To solve all the challenges above, we come up with a fast-NLP based filtering model which can recognize tweet content pattern in real time and compute the similarity closeness

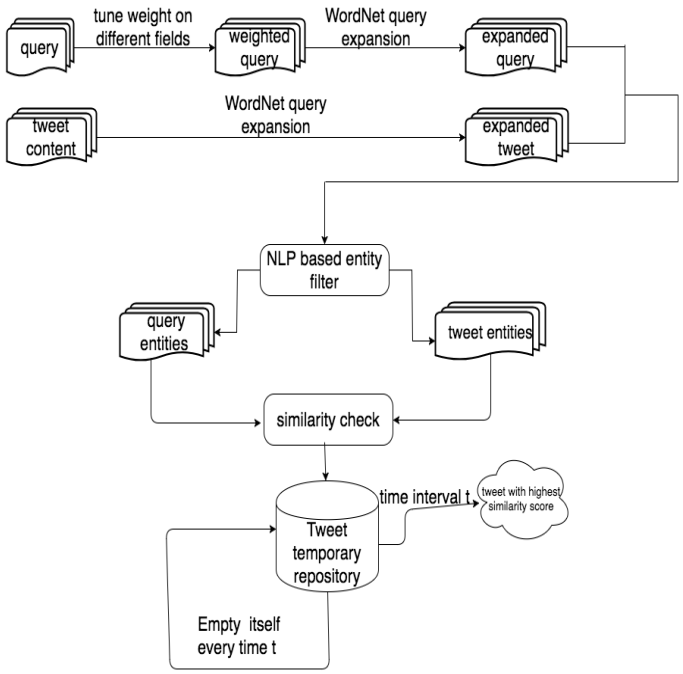

Fig. 1. Tweet real time recommendation in Scenario A

with user interest profiles. We conclude our model into the following 4 steps. The details of the work flow is showed in Figure 1.

First step is query reconstruction. Based on the task, a topic (query) is formed with three fields including 'narrative', 'title', 'description'. Title is the main keywords of a topic, which only contains several words only such as 'HPV vaccine side effects'. Description extends the title a little bit to briefly introduce what a user is looking for towards the topic. For example, to the title above, its related description is 'Information concerning possible side effects of the HPV vaccine'. Narrative is a short paragraph to describe the purpose of the topic and why a user is interested in the topic. It contains one or several sentences to describe and its content length a little bit longer than the rest two fields.

Each field refers to different information, to best capture users' interest, we use a linear combination of these three fields to finalize the query for a given topic. We can tune all the weights or arbitrarily assign weights based on empirical experience.

$$Q = \alpha Q(title) + \beta Q(description) + \gamma Q(narrative)$$

Second, as both tweets and topics are short documents, it is necessary to expand their content to retrieve more semantic meaning and content information. There are multiple ways for content expansion. By comparing different techniques such as Google Search Result Page or using Word2vec on large existing corpus to retrieve synonyms, we decide to use WordNet as an external resource to extend our both tweet content and topic content.

Third step is NLP based entity filtering. As both tweets and topics are unstructured text information, in order to extract quantitative information out of them, NLP related methods are applied. Using POS tagger, given a piece of text, we can extract all nouns, adjectives and verbs from it which contain most of meaning of the text. Moreover, by leveraging bag-of-word model, besides 1-gram words, we also extract 2-gram words as these words contain structural relationship as well as more semantic meanings of the sentence. In the end, after NLP based entity filtering step, both tweets and topics are represented as a batch of 1-gram and 2-gram entities. In the end, we can project both topics and tweets into a high dimensional space where each dimension refers to an entity.

Therefore, the fourth step is to build up a similarity calculation function to quantify the timeliness, relevance, and novelty of each coming tweet. We define the function f as:

$$f(tweet, topic) = (cos(tweet, topic) - \frac{\sum_{i=1}^{N} cos(tweet, C)}{N}) D(t)$$

$$D(t) = 1 - e^{\frac{-(t-t_0)}{\gamma}}$$

where D(t) is a time decay function, cos() refers to the cosine similarity between two pieces of text and C is the existing tweet collection. By leveraging similarity function f, we can quantitatively evaluate each tweet and push it into a tweet temporary repository C.

In every time interval T, we push the tweets with highest similarity score of a topic to users and empty the tweet collection repository.

B. Scenario B

In Scenario B, everyday we need to generate a tweet ranking list given a particular topic for users as an email digest. As users are not willing to read too many tweets at a time, the upper bound of tweet number is 100. Therefore, it is a pure information retrieval problem. We don't need to consider timeliness and novelty anymore, the only criteria is tweet relevance towards a topic.

Everyday during the 'Twitter Streaming listening' period, we stored the sampled tweets into local disk. In the end of each day, we generate a ranking list for each topic based on the stored daily tweet collection.

In Scenario B, the first several steps are the same as Scenario A, data pre-processing is also necessary in the begining. We The overall flow is showed in Figure 2.

In the beginning steps, we also need to calculate weighted queries, and extract entities from both tweets and topics (queries) and therefore project tweets and topics into the same high dimensional space.

After that, we create a language model ensemble in which there are five classic language models including TF-IDF model, Vector space model, BM25 model, language model with Dirichlet Smoothing and language model with Jelinek Mercer Smoothing. For each topic-tweet pair, we calculate the similarity score using all five models. We standardize each model similarity score so that similarity scores in each model

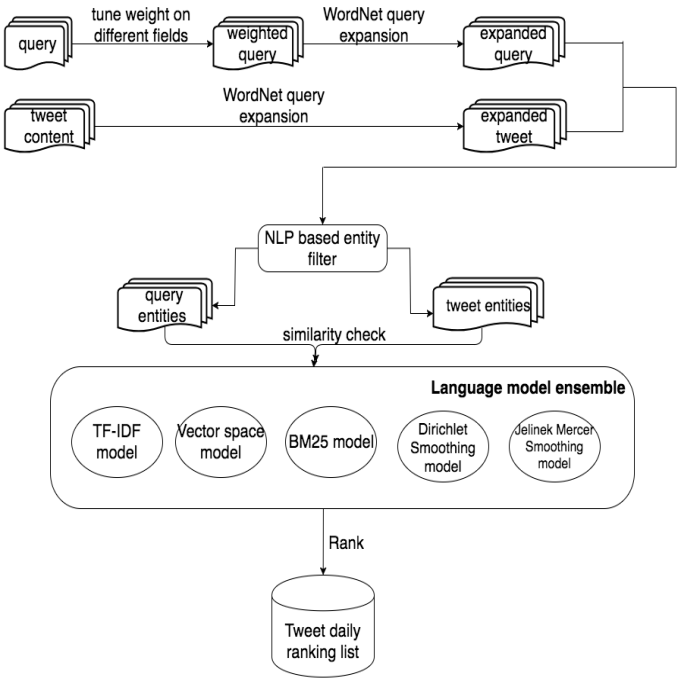

Fig. 2. Tweet email digest in Scenario B

are in the same range and comparable. Afterwards, for each tweet-topic pair, we sum up all five similarity scores together as a representation of final similarity score. In the end, for each topic, we rank the top 100 tweets with highest similarity scores and push the ranking list to users as a daily email digest.

IV. RESULT

In total, the RTS track offers 188 topics. Each day, all participants need to submit up to 10 tweets for each topic for Scenario A and a tweet ranking list up to 100 tweets for each topic to fulfill the requirement of Scenario B. We received the evaluation result based on both human assessors and NIST official pooling evaluation.

A. Scenario A

TABLE I. EVALUATION RESULT FOR SCENARIO A

Assessment	NIST	Assessment	Mobile
strict precision	0.3403	EG-p	0.2194
lenient precision	0.4174	EG-1	0.1951
strict utility	-805	nCG-p	0.2095
lenient utility	-456	nCG-1	0.1826

In Scenario A, the result are judged from 2 independent resources. One is from NIST assessment and the other is from mobile assessment.

NIST assessors judged 96 topics this year. Per official track guidelines, the metrics used include EG-p, EG-1, nCG-p and nCG-1.

Mobile assessors judged 188 topics (with uneven effort), and some tweets were judged by multiple assessors.

The whole result is showed in Table 1.

B. Scenario B

TABLE II. EVALUATION RESULT FOR SCENARIO B

nDCG@10-p	0.2194
nDCG@10-1	0.1865

For Scenario B, the result is evaluated by NIST assessors. It judged 96 topics this year. Per official track guidelines, the official metrics are nDCG@10 and nDCG@10-p. And the detail is showed in Table 2.

V. CONCLUSION

Even though this is the second year of the Real Time Summarization track in TREC, as it merges two former tracks, its tasks reflect real user need in real society. Due to this, we believe solving these tasks is a great help to explore more innovations in both information retrieval and recommendation domains. In our work, we develop a Fast NLP-based Pattern recognition model to calculate tweet similarity towards a given topic in relevance, timeliness and novelty aspects. The result shows we can retrieve considerably good results for users. In the future, we are planning to focus more on pattern analysis on text content by involving more NLP based models such as constituency-based parse tree to generate better features containing more semantic meaning of original contents.

REFERENCES

[1] A. Bandyopadhyay, K. Ghosh, P. Majumder, and M. Mitra. Query expansion for microblog retrieval. *International Journal of Web Science*, 1(4):368–380, 2012.

[2] C. Bei and P. Hu. Ccnu at trec 2016 real-time summarization track. In *TREC*, 2016.

[3] Z. Gao and R. Bi. University of pittsburgh at trec 2014 microblog track.

[4] Z. Gao and X. Liu. Personalized community detection in scholarly network.

[5] K. Lee, A. Qadir, V. V. Datla, S. A. Hasan, J. Liu, A. Prakash, and O. Farri. Assorted textual features and dynamic push strategies for real-time tweet notification. In *TREC*, 2016.

[6] H. T. D. L. W. Li. Polyu at trec 2016 real-time summarization.

[7] J. Lin, A. Roegiest, L. Tan, R. McCreadie, E. Voorhees, and F. Diaz. Overview of the trec 2016 real-time summarization track. In *Proceedings of the 25th Text REtrieval Conference, TREC*, volume 16, 2016.

[8] X. Liu, X. Yu, Z. Gao, T. Xia, and J. Bollen. Comparing community-based information adoption and diffusion across different microblogging sites. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, pages 103–112. ACM, 2016.

[9] B. Moulahi, L. B. Jabeur, A. Chellal, T. Palmer, L. Tamine, M. Boughanem, K. Pinel-Sauvagnat, and G. Hubert. Irit at trec real time summarization 2016.

[10] R. Suwaileh, M. Hasanain, and T. Elsayed. Lightweight, conservative, yet effective: Scalable real-time tweet summarization. In *TREC*, 2016.

[11] K. Wang and Z. Yang. Bjut at trec 2016: Real-time summarization track. In *TREC*, 2016.

[12] L. Yao, C. Lv, F. Fan, J. Yang, and D. Zhao. Pkuicst at trec 2016 real-time summarization track: Push notifications and email digest. In *TREC*, 2016.

[13] C. Zhang, Z. Gao, and X. Liu. How others affect your twitter# hashtag adoption? examination of community-based and context-based information diffusion in twitter. *IConference 2016 Proceedings*, 2016.