# DAIICT at TREC RTS 2016: Live Push Notification and Email Digest

Sandip Modha *sjmodha@gmail.com*[1],
Chintak Mandalia *chintak.soni*75*@gmail.com*[2],
Krati Agrawal *kratiagrawal*1410*@gmail.com*[1],
Deepali Verma *deepaliverma*394*@gmail.com*[1], and
Prasenjit Majumder *prasenjit.majumder@gmail.com*[1]

[1]DAIICT Gandhinagar Gujarat-382007
[2]LDRP Gandhinagar Gujarat-382015

January 30, 2017

## ABSTRACT

This paper describes the participation of Information Retrieval Lab(IRLAB) at DA-IICT Gandhinagar,India in Real-Time Summarization track TREC 2016. This year TREC RTS offered two tasks. In the first task, that is scenario A, our system will be monitoring continuous posts from Twitter public stream and push the relevant tweet for each interest profile to RTS evaluation broker. For the same, we have expanded interest profile using Word2vec training model with past 30 days tweets. We have calculated relevance score between tweets and expanded interest profile using Okapi BM25 model. For Scenario B, Email digest, we anticipated summarization problem as a clustering problem. In scenario A, we reported result in terms of Expected Gain EG-1(primary metric)=0.1708 and in scenario B we have achieved primary metric nDCG-1 = 0.1972.
**Keywords:** Social media, BM25, clustering, jaccard similarity, cosine similarity, Word2vec.

## 1 INTRODUCTION

Social media, like Twitter, is one of the noisiest sources of real-time information. Twitter, a popular microblogging website, which has massive user-generated content due to its large number of registered users. Due to its real-time nature,this year TREC 2016 has offered Real-Time Summarization (RTS) track with following objectives: (i) how fast the participating system can deliver relevant and novel tweets based upon the interest profile to mobile assessor via

RTS broker through push notification. (ii) How we can generate the day-wise summary of tweet for each interest profile.
Twitter allows its registered user to post up to 140 character short message called tweet. Due to this limitation, it is challenging for participating system to calculate its relevancy against the interest profile. In the rest of paper we will use tweet and post interchangeably. The organizer of TREC 2016 RTS[1] gave 203 interest profiles having topic-id, title, narrative and description. They were a combination of interest profiles which were assessed from TREC 2015, culled from 2015 and additional for 2016. However all the interest profiles were provided before the evaluation period. TREC RTS 2016 had two scenarios.

The two scenarios were: Scenario A[1]: Push notifications: Participating system continuously listens to the Twitter sample stream using Twitter Streaming API[7]. The Twitter streaming API offers an approximately 1% sample of all tweets (sometimes called the "spritzer") and is freely available to all registered users. As soon as the system identifies a relevant post against the required profile, it is immediately pushed to the user's mobile phone via a push notification. Push notifications should be relevant, timely and novel. Scenario B[1]: Email digest. Alternatively, a user might want to receive a daily email digest that summarizes what happened that day with respect to the interest profiles. At a high level, these results should be relevant and novel; timeliness is not particularly important, provided that the tweets were all posted on the previous day.

For Scenario A, we pre-processed interest profile, removed all the stop words, and considered only noun, proper noun, and verb using Stanford POS tagger. We then expanded interest profile using Word2vec[6]. We trained Word2vec model for each profile with past 1-month profile specific tweet corpus. In the next step, live tweets were collected, pre-processed and cleaned. Then we applied Okapi BM25 ranking function to get relevance score between tweets and each expanded interest profile. The threshold for interest profile was experimentally determined by studying our system results for 15 days prior to the competition. If the relevance score of a tweet is more than the threshold, it will push to RTS broker. After that, to assure novelty we applied Jaccard similarity between the tweets which were already pushed and current tweet.

Scenario B, e-mail digest, the submission was after scenario A in which we had to send a maximum of 100 tweets for each profile each day. At the end of a day, tweets were sorted in descending order of BM25 score for each interest profile. All the tweets above the threshold were selected and saved according to the template given by TREC 2016. Again threshold was determined experimentally by studying our system results 15 days prior to the competition.

The rest of paper is organized as follows: In section2 we discuss related work, In section 3 we define the problem statement for scenario A and B. In Section 4 we discussed the methodology for scenario A. In section 5 we describe the methodology for scenario B. In section 6 we discussed the result and performance evaluation metric. In section 7 we conclude the discussion.

# 2 RELATED WORK

We started our work by referring TREC MICROBLOG 2015 papers.

CLIP[2] has trained their Word2vec model using 4 years tweet corpus. They used Okapi BM25 relevance model to calculate the score. To refine the scores of the relevant tweets, tweets were rescored using the SVM rank package using the relevance score of the previous stage. Then Novelty Detection is done, where the tweets which are not useful are discarded, this is done using Jaccard similarity.

University of waterloo[4] implemented the filtering tasks, by building a term vector for each user profile and assigning different weights to different types of terms. To discover the most significant tokens in each user profile, they calculated pointwise KL divergence and ranked the scores for each token in the profile.

# 3 PROBLEM STATEMENT

Scenario A was mainly real time filtering task.
Given an interest profile Q=$\{Q_1, Q_2, ..Q_n\}$, and stream of tweets T=$\{t_1, t_2, ..t_n\}$ from public sample stream we need to compute the relevance score between tweets and profile R_score=f(Q,T). Tweets having R_score greater than threshold with respect to profile moved in the set PT= $\{pt_1, pt_2, ...pt_n\}$. At most, 10 novel tweets can be pushed to RTS broker per profile per day.

In the Scenario B, summary s=$\{s_1, s_2, ..., s_n\}$ has to be formed from relevant tweet RT=$\{rt_1, rt_2...rt_n\}$ where rt_$i$ represents relevant tweet for a particular profile. A batch of top 100 ranked tweets per day per interest profile with any two tweets having a similarity of less than threshold $\text{sim}(t_1, t_2) < Ts$ is used for Email Digest. All tweets from 00:00:00 to 23:59:59 were eligible candidates for a particular day.

# 4 METHODOLOGY FOR SCENARIO A:

## 4.1 Interest Profile Pre-processing

TREC RTS 2016 has given 203 interest profiles. We converted these profiles into query by removing stop words and considering nouns, proper nouns and verbs using Stanford POS tagger.

## 4.2 Profile (Query) expansion

We downloaded 1-month profile specific tweet corpus and trained Word2vec model for finding top 5 similar words and hashtag. We have set different weights for original terms and expanded terms.

### 4.3 Profile Normalization

All interest profiles were gathered from above three mentioned sources. Title and description were merged so as to make interest profiles more informative. To increase the relevance, interest profiles were also pre-processed by converting all alphabets to small case and expanding the abbreviations. Example: NYC- New York City. Also, interest profiles were stemmed. Eg: behaving was converted to behave.

### 4.4 Tweet Pre-processing

After gathering tweets, non-English tweets were filtered out. Tweet includes smileys, hashtags, and many special characters. We did not consider retweets and tweet with only hashtag or emoticon or special characters. We also ignored the tweet with less than 5 words and removed all the stopwords from the tweet.

### 4.5 Relevance Score

To calculate relevance score between tweets and interest profiles, we set weight as 2 for the original term in the interest profile and 1 for the terms added after training the profile. We have used BM25 model for calculating relevance score between expanded interest profile and query.Score is defined as:

$$R\_score = BM25\_Sim(Q\_exp, T)]$$

### 4.6 Novelty detection

For novelty detection, Jaccard Similarity algorithm was used.

$$J(A, B) = (A \cap B)/(A \cup B)$$

where A and B are pushed and current tweets respectively. The highest ranked tweet for each profile was sent to TREC for assessment. Now for next eligible tweet, we calculated it's similarity with already sent tweet(s) so as to ensure novelty between them. Again a Jaccard threshold was decided and tweets below it were sent. Lower the similarity score, greater is the dissimilarity ensuring more novelty.
Also, there were interest profiles where no relevant tweets for some day were found, which were known as SILENT DAYS. There were marks for proper treatment of silent days.

## 5 METHODOLOGY FOR SCENARIO B:

In this scenario, we had to make a summary up to top 100 relevant tweets for each interest profile. We applied Okapi BM25 ranking function to calculate

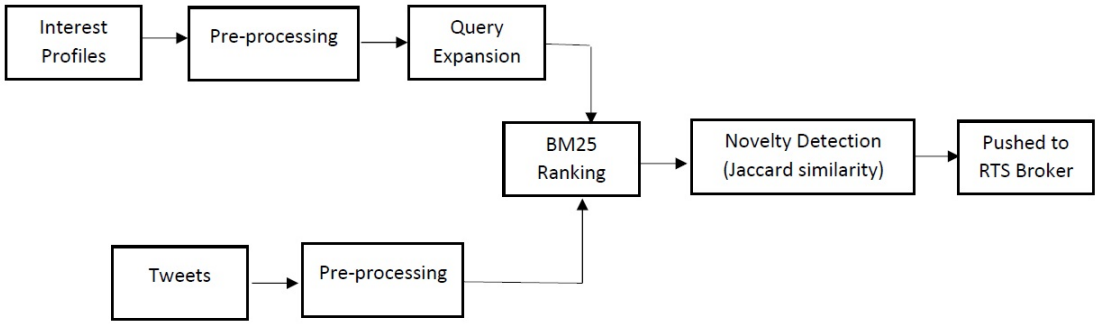


Figure 1: Scenario-A

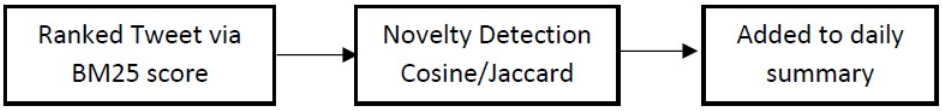


Figure 2: Scenario-B

the rank. So, for each day we had relevant tweets corresponding to each interest profile. A particular format of the plain text file was given by TREC, that is:

YYYYMMDD topic_id Q0 tweet_id rank score run tag

Again, for top 100 novel tweets two similarities technique were applied: Jaccard Similarity and Cosine Similarity. Jaccard was same as in scenario A . However, Jaccard similarity proves out to give more robust results.

$$CosineSimilarity(A, B) = \vec{A} * \vec{B}/||\vec{A}|| * ||\vec{B}||$$

where A and B are pushed and current tweets respectively.

# 6 RESULTS

The evaluation of TREC 2016 Microblog track lasted 10 days, from Monday, August 2, 2016, 00:00:00 UTC to August 12, 2016, 23:59:59 UTC. It consisted of 203 interest profiles. During the evaluation time, participants listened to the tweet stream continuously and analyzed with every tweet.

## 6.1 Scenario A: User in the loop assessments

This approach is new for TREC 2016 and promises a number of significant advantages over traditional post hoc batch evaluations because it is able to capture live user assessments. In this method, tweets submitted by participating systems to the RTS evaluation broker are immediately routed to the mobile phone of an assessor, where it is rendered as a push notification containing the text of the tweet and the corresponding interest profile. The assessor may choose to judge the tweet immediately, or if it arrives at an inappropriate time, to ignore it. Either way, the tweet is added to a judging queue in a custom app on the assessor's mobile phone, which the assessor can access at any time to judge the queue of accumulated tweets. As the assessor judges tweets, the results are relayed back to the evaluation broker and recorded.

| Relevant | Redundant | Non_Relevant | Unjudged | Total_length |
|---|---|---|---|---|
| 105 | 10 | 259 | 1721 | 2083 |

Table 1: Result of Live User Assessment

## 6.2 Scenario A: Post Hoc Batch Evaluations

The evaluation methodology was based on pooling. A common pool has been constructed based on scenario A and scenario B submissions. The pool depth was be determined after the evaluation period ended by NIST based on the number of submissions and available resources. The assessment workflow is as follows: First, tweets returned by the systems were assessed for relevance. Tweets were judged as not relevant, relevant, or highly relevant.
Also, two main differences between the metrics this year and the metrics from TREC 2015 were treatment of "silent days" and treatment of the latency penalty

Expected Gain (EG)[7] is defined as:

$$EG(t) = (1/N) \sum G(t)$$

where N is the number of tweets returned and G(t) is the gain of each tweet:

- Not relevant tweets receive a gain of 0.
- Relevant tweets receive a gain of 0.5.
- Highly-relevant tweets receive a gain of 1.0.

Once a tweet from a cluster is retrieved, all other tweets from the same cluster automatically become irrelevant. This penalizes systems for returning redundant information. Note that unlike last year, there is no latency penalty applied to the gain; the latency is computed separately (see below). Normalized Cumulative Gain (nCG)[7] (for an interest profile on a particular day) is defined as follows:

$$nCG(t) = (1/Z) \sum G(t)$$

where Z is the maximum possible gain.

Gain Minus Pain (GMP)[7] is defined as follows:

$$GMP = \alpha * G - (1 - \alpha) * P$$

The G (gain) is computed in the same manner as above; P (pain) is the number of non-relevant tweets that are pushed and  controls the balance between the two.

| EG1 | EG0 | nCG1 | nCG0 |
|---|---|---|---|
| 0.1708 | 0.0440 | 0.1546 | 0.0278 |

Table 2a: Result of Post Hoc Batch Evaluation

| GMP.33 | GMP.5 | GMP.66 | mean latency | median latency |
|---|---|---|---|---|
| -0.7448 | -0.5397 | -0.3467 | 176709.4 | 36152.0 |

Table 2b: Result of Post Hoc Batch Evaluation

### 6.3 Scenario B

In scenario B, nDCG score computed of each day for each interest profile and then average across them. In nDCG1, on a "silent day", the system receives a perfect score if it does not return any tweets and zero otherwise. In nDCG0, for a silent day, all systems receive a gain of zero no matter what they do.We have submitted 2 run for scenario B namely IRLAB and IRLAB2. In the first run,IRLAB, we have applied cosine similarity to ensure novelty between tweets and we have achieved nDCG1= 0.1532. In the second, IRLAB2 we have applied Jaccard similarity between relevant tweets of each interest profile for novelty detection and achieve nDCG1=0.1972.

| nDCG1 | nDCG0 |
|---|---|
| 0.1972 | 0.0169 |

Table 3: Run tag A : IRLAB2

| nDCG1 | nDCG0 |
|---|---|
| 0.1532 | 0.0711 |

Table 4: Run tab B : IRLAB

# 7 CONCLUSION

In this paper, we have calculated relevance score between a tweet and expanded interest profile using BM25 Ranking function. Novelty detection has been done using Jaccard similarity between previous pushed tweet and tweet to be pushed. We achieved average EG-1 = 0.1708. After careful analysis of result, we conclude that our system performance is better for TREC 2015 interest profiles compared to new interest profile created for TREC 2016. We also conclude that Jaccard similarity outperforms cosine similarity for novelty detection between tweets after analyzing the result of scenario B.

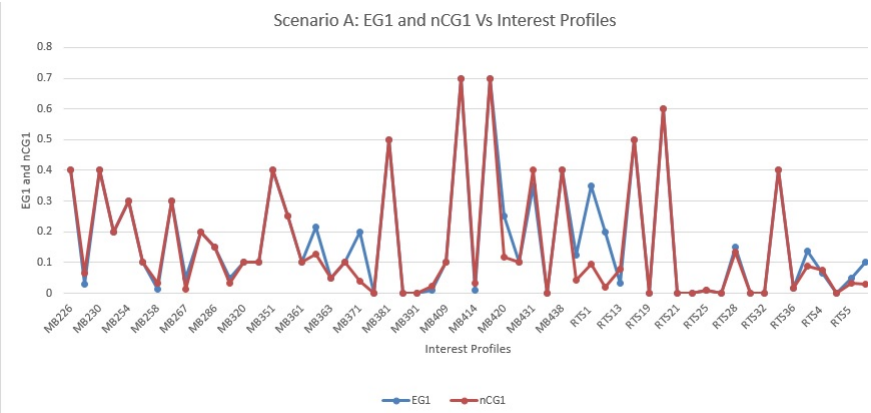


Figure 3: Scenario-A

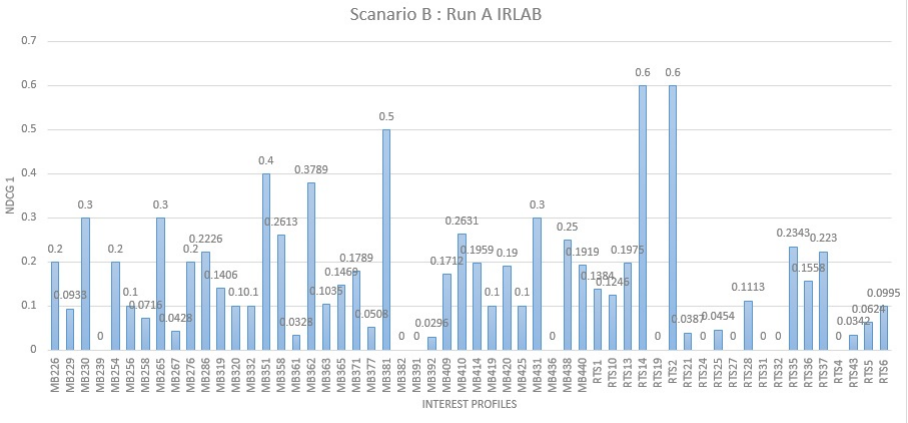


Figure 4: Scenario-B IRLAB

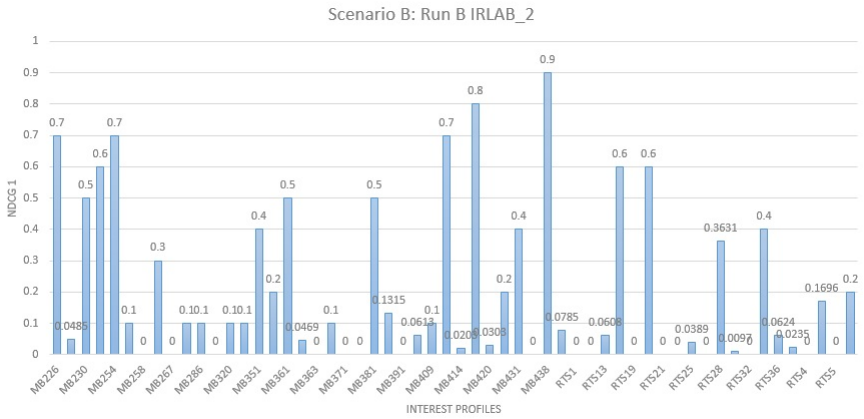


Figure 5: Scenario-B IRLAB2

# References


[1] *TREC 2016 Official Guidelines*

[2] Mossaab Bagdouri, Douglas W.Oard.
*CLIP at TREC 2015: Microblog and LiveQA.*

[3] Xiang Zhu, Jiuming Huang, Sheng Zhu, Ming Chen, Chenlu Zhang, Li Zhenzhen, Huang Dongchuan, Zhao Chengliang, Aiping Li, Yan Jia
*NUDTSNA at TREC 2015 Microblog Track.*

[4] Luchen Tan, Adam Roegiest, Charles L.A. Clarke
*University of Waterloo at TREC 2015 Microblog Track*

[5] Luchen Tan, Adam Roegiest, Charles L.A. Clarke, Jimmy Lin
*Simple Dynamic Emission Strategies for Microblog Filetering*

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean.
*Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013*

[7] Luchen Tan, Adam Roegiest, Jimmy Lin, and Charles L. A. Clarke *An Exploration of Evaluation Metrics for Mobile Push Notifications*

[8] Shamanth Kumar, Fred Morstatter, Huan Liu
*Twitter Data Analytics*

[9] Lichan Hong, Gregorio Convertino, Ed H. Chi
*Language Matters in Twitter: A Large Scale Study*