

# NOVASearch at TREC 2017 Real-Time Summarization Track

Gustavo Gonçalves

NOVA LINCS

Department of Computer Science  
Faculty of Science and Technology  
Universidade NOVA de Lisboa  
Caparica, Portugal  
gs.goncalves@campus.fct.unl.pt

Flávio Martins

NOVA LINCS

Department of Computer Science  
Faculty of Science and Technology  
Universidade NOVA de Lisboa  
Caparica, Portugal  
flaviomartins@acm.org

João Magalhães

NOVA LINCS

Department of Computer Science  
Faculty of Science and Technology  
Universidade NOVA de Lisboa  
Caparica, Portugal  
jm.magalhaes@fct.unl.pt

## ABSTRACT

The rise of large data streams introduces new challenges regarding the delivery of relevant content towards an information need. This information need can be seen as a broad topic of information. One possible strategy to tackle the delivery of the most relevant documents regarding this broader topic is summarization. TREC 2017 Real-Time Summarization (RTS) provides a testbed for the development of stream based real-time summarization systems. Leveraging on the social media network, Twitter, the participants are challenged to deliver the most relevant and diverse information in two main scenarios.

The real-time push notifications scenario, or Scenario A, focuses on the identification and delivery of relevant information in near real-time. Whereas the daily-digest scenario, or scenario B, strives for the daily delivery of the most relevant and diverse documents.

This paper presents the participation of the NOVASearch group at TREC 2017 Real-Time Summarization (RTS). Our work was developed for tackling the scenario B, after an analysis of the proposed systems for the TREC RTS 2016. In our approach we explore document filtering methods; vocabulary expansions; and the identification of subtopics through the aggregation of documents based on their textual similarity.

## 1 INTRODUCTION

The continuous generation of content in social-media networks originates real-time data streams forming a deluge of data, that needs to be filtered and analyzed to identify information with significance towards end users. The TREC Real-Time Summarization (RTS) Track aims to tackle these issues through the delivery of the most relevant tweets in a near real-time push notifications scenario (A), or in a delayed digest in the daily summarization scenario (B). The two scenarios share the criteria of delivering relevant and diverse information. This track introduces new challenges, such as dealing with a dataset that is continuously expanding and prepare algorithms to face different relevance distributions over time, ultimately identifying the most relevant content and deliver it at just the right time. This ever-growing data volume might overwhelm users with duplicated or irrelevant information. This data volume introduces the need for algorithms that consider multiple features to estimate the relevance of a specific sub-branch of information and its change over time, to deliver relevant content at the right moment.

We developed a system aiming to participate in the daily summarization scenario, where significant occurrences towards a given topic are summarized at the end of the day. With our participation,

we have studied the impact of different retrieval models, vocabulary expansion methods, and summarization approaches. Our architecture is presented in Section 3 of this paper, as well as the summarization technique explored, and finally, the achieved results are presented in Section 4.

## 2 RELATED WORK

To tackle this summarization challenge, we started by studying the participation notes of the TREC RTS 2016 track. From the fifteen participating teams of the 2016 TREC RTS track Lin et al. [6], fourteen of them participated in the push notification scenario (A) and all of them in the daily summary scenario (B). As a starting point to develop our architecture we analyzed the approaches published in the participation papers that were submitted. The ten teams that submitted participation papers developed and evaluated systems for both scenarios A and B. We were able to identify the most common techniques that were used to tackle the different challenges posed by the track. Most works [1, 2, 4, 5, 11–16] applied filtering and text cleaning techniques to the incoming tweets before further analysis and indexing. The employed filters and cleaning criteria were based on heuristics that varied in strictness across teams. The most aggressive filtering modules focused on discarding tweets that did not contain any query term, such as the ones applied in Lee et al. [4], Li et al. [5], Tan et al. [14]. Several works opt for a more permissive approach applying a filtering criterion that considers the text length, presence of URLs Bei and Hu [2], Modha et al. [11]. In general, all works identified a need to strip special characters from the tweets, remove stop words and to stem the incoming text before further analysis and indexing.

The typical strategy used to achieve summarization was to define relevance and similarity thresholds as requirements for the submission of new information. Moreover, there are several approaches Moulahi et al. [12], Suwaileh et al. [13], Wang and Yang [15] that relied on query expansion to overcome the vocabulary mismatch problem brought forward by the characteristics of Twitter. Query expansions were based on external sources or the analysis of textual elements in the provided Interest Profiles. The external sources used ranged from Twitter itself to commercial search engines, such as Bing, as we can observe in the systems developed by Suwaileh et al. [13], and Wang and Yang [15], respectively. The identification and removal of duplicate information was a generalized concern tackled by all teams. Most teams defined a threshold based on a standard similarity measure, e.g., the Jaccard Similarity or designed their own as is the case of Moulahi et al. [12].

Some teams opted to use learning to rank methods in their systems, using features extracted from the available textual elements,

**Table 1: Summary of the approaches used in TREC RTS 2016.**

	Pre-processing		Summarization techniques				
	Filtering	Text cleaning	Threshold definition	Query or Doc. expansion	Duplicate removal	Learning to rank	Clustering
BJUT [15]	✓	✓		✓	✓	✓	✓
CCNU [2]	✓	✓	✓	✓	✓		
CLIP [1]	✓	✓	✓	✓	✓	✓	✓
DAICT [11]	✓	✓	✓	✓	✓		✓
HLJIT [5]	✓	✓	✓		✓		
IRIT [12]	✓	✓	✓		✓		
PKUICST [16]	✓	✓	✓	✓	✓		
PolyU [14]	✓	✓	✓	✓	✓	✓	✓
PRNA [4]	✓	✓	✓	✓	✓		✓
QU [13]	✓	✓	✓	✓	✓		

with the objective of automatically tuning these thresholds for relevance and similarity; ultimately better preparing their systems for the track. Bagdouri and Oard [1] is an example of leveraging on these methods and external datasets to train their system.

From this study we leveraged and adapted common techniques used across teams during the 2016 RTS track, emphasizing the work developed by the top three teams of the 2016 edition Moulahi et al. [12], Suwaileh et al. [13], Tan et al. [14].

### 3 TOPIC SUMMARIZATION

To build our approach to the daily summarization problem, we surveyed the techniques used by the systems on the 2016 edition [6]. Our architecture is inspired by the best participants systems and applies a similar pipeline. The system can be split roughly into three stages: preprocessing, vocabulary expansion, and summarization.

#### 3.1 Preprocessing

To discern between noise and useful documents a couple of heuristics were used. These heuristics were employed to avoid indexing irrelevant documents as early as possible in the system’s pipeline.

Firstly, incoming tweets are filtered according to the language of its textual content. We rely on the classification in the tweets’ *lang* metadata field, included by Twitter. This field exposes the internal language classification made by Twitter. Only tweets written in English were considered, as defined in the track’s guidelines.

To help with the removal of spam tweets, we filter tweets based on the URLs [7] linked in the text. The URLs are matched to a comprehensive URL blacklist<sup>1</sup> and are discarded immediately, if the URL domain is contained in the blacklist. The list contains web

<sup>1</sup><https://github.com/StevenBlack/hosts>

**Table 2: Tweets containing URLs are expanded with title of the pointing URL Web page, creating the so called virtual document [10].**

Tweet text	Expansion with URL page title
Russia’s information war might be thought of as the biggest trolling operation in history	Russian internet trolls were being hired to pose as pro-Trump Americans - Business Insider

pages classified as fake news, gambling, and pornography; thus avoiding the exploration of tweets that contain URLs that lead to irrelevant content.

The next step in our preprocessing pipeline consists in an adaptation of the work developed by the Qatar [13] and Hong-Kong [14] universities, in the 2016 edition of the track. We verify the tweets’ quality by enforcing two criteria. First, a tweet must have more than twenty characters, and second, it must not be entirely capitalized. We used these criteria to guarantee that there were sufficient textual elements to be analyzed and to avoid the indexing of small trivial documents.

Finally, the last step of the module, before proceeding with the tweet’s indexing, consists in cleaning the tweet’s text. We clean the tweet’s text by stripping it of special characters; splitting well-formed hashtags and adding the to the text; removing stopwords using Indri’s stoplist<sup>2</sup>; applying a lower case filter; and stemming using the Porter stemming algorithm.

#### 3.2 Vocabulary expansion

Twitter is a microblogging platform, and its documents are limited to 140 characters. Therefore, we apply query and document expansions to overcome the vocabulary limitations of the documents shared on this social media network.

**3.2.1 Query expansion.** For the query expansion, we experimented with two different models leveraging on Named Entity Recognition (NER) and Pseudo-Relevance Feedback (PRF) with external news sources. For the TREC RTS track, information needs were simulated through the creation of interest profiles that are composed by a title query, a description, and a narrative. The fields present in an interest profile differ regarding their extent, providing progressively more context and information about the user’s information need.

For our query expansion based on NER, we identified entities in the narrative of the interest profile and added them to the title query, provided that they were not already present. The named entity recognition was achieved resorting to the CoreNLP software [8] developed by the Stanford NLP lab. We focused on the identification of locations, persons, organizations and miscellaneous entities in the narrative of the topics. This method aims to obtain more vocabulary related to the information need adding words that were not in the title query. However, due to stemming it will increase the weight of some words that might be under the same lexical root.

<sup>2</sup><http://www.lemurproject.org/stopwords/stoplist.dft>

Moreover, this approach does not provide a generalized gain to all topics, since not all topics contain entities in their narratives.

For the PRF we leverage on the work of Martins et al. [9], which have developed a framework, named Jitter API, that delivers expansion terms based on monitored Twitter endpoints. Martins et al. [9] employ time-aware ranking models to deliver expansion terms based on the content published on the monitored endpoints. We focused on the *Pseudo-Relevance Vertical Feedback (PRVF)* endpoint, that monitors the content published in Twitter news channels. From this endpoint we retrieved 20 expansion terms. Where the original query terms gathered a total weight of 50%. With this balance we aimed to avoid topic drifts, still leveraging on the recent topic terms that were detected by the Jitter API.

The NER expansion is combined with the title query using a weighted interpolation, expressed by  $Q = IP_t + \alpha NERExp(IP_{\{n\}})$ . Where  $Q$  is the resulting query,  $IP_t$  is the interest profile's title,  $\alpha$  the determined interpolation factor, and  $NERExp(IP_{\{n\}})$  is the expansion of the interest profile's narrative. The PRF expansion terms provided by the Jitter API [9], already consider the original query terms weighted accordingly.

The NER expansions were performed at the start of the competition, since they only depend on the interest profiles. Whereas the PRF expansions were made at the end of each day, before generating the summaries.

**3.2.2 Document expansion.** The exploration of linked content via hyperlinks for microblog search was explored by McCreadie and Macdonald [10] by using the virtual document model. We leverage on this idea, to expand the tweet's vocabulary. The tweet's vocabulary is expanded by adding the words in the titles of the linked web pages, to the tweet's original text. If the URL's domain shared in the tweet is not present in the URL blacklist, it will be followed. We will clean the text as in the preprocessing phase and index it jointly with the original tweet's text, provided that we can successfully extract the title of the resolved web page.

Our premise is that trustworthy sources of information, such as news accounts, tend to use Twitter to share their news through short descriptions followed by a link for the full news. By extracting the web page title, we hope to expand the tweets vocabulary with words that were not initially present in the short news description.

### 3.3 Retrieval

We leveraged on several retrieval models to obtain the ranking of the documents, such as Vector Space Model with cosine distance (VSM), Okapi-BM25, and *query-likelihood* language models with Jelinek-Mercer and Dirichlet smoothing retrieval models.

We tuned the parameters of BM25 on the TREC RTS 2016 dataset and found the best parameters to be  $k = 1.0$ , and  $b = 0.2$ . These values indicate that BM25 performs better is stronger when focusing on term frequency, and less to document length normalization, in a dataset composed of small documents.

We use language modeling with Jelinek-Mercer (LMJM) smoothing with the parameter  $\lambda = 0.9$ . The LMJM smoothing settled on a relatively high-value when compared with previous studies, where  $\lambda = 0.1$  [17]. However, these experiments were done on larger documents. Thus we verified that when working with short documents

LMJM performs better when relying on the relative weighting of terms, aiming to match most query terms.

We use language modeling with Dirichlet (LMD) smoothing with the parameter  $\mu = 200$ . For LMD we found better results with a small value when compared with the usual value  $\mu = 2000$  [17]. Once again this is due to the short length of our documents. Using this parameterization makes LMD focus on term frequency for matching documents to queries.

The best parameters found by tuning on the TREC 2016 dataset are used for the experiments with the TREC 2017 dataset for better ranking results. We also leveraged on Reciprocal Rank Fusion (RRF) [3], with the proposed value of 60 for web datasets, to combine the results of these tuned models: BM25, LMD, and VSM, to diversify our ranking and better approximate the optimal ranking. The usage of RRF was also influenced by the limitation of the number of submissions. Since each team is limited to submit three runs, in each scenario, we use RRF to fuse different rankings aiming to gather more relevant documents; and using the runs to evaluate other features of our system.

### 3.4 Summarization

The final module of our system is the summarization module. This module strives to identify subtopics of interest that arise during the event, under the main topic. We aimed to summarize a topic by grouping documents and choosing the most representative ones to submit to the daily summary.

Our summarization strategy was developed to meet the evaluation procedure of the track. Where relevant documents were semantically clusters by NIST assessors, and only the first submitted tweet from each cluster would contribute for the final score. With this evaluation procedure in mind, we implemented a single-pass clustering and diversity algorithm, Algorithm 1.

Algorithm 1 iterates the document ranking, for every query, by the decrescent score of documents, as shown in lines 4 to 16. In lines 6 and 7 we can observe that the first document in the ranking will form the first cluster. Afterwards, every document will be compared with the first document of each cluster. Since the clusters are created following the ranking score, the comparison with the first document of each cluster guarantees that we are comparing each document with the most relevant document of each cluster. In lines 9 to 13, we add each document to the first cluster where the similarity comparison surpasses the pre-defined threshold. If a candidate document never surpasses the threshold, it will create a new cluster where the document will be the cluster representative, as shown in line 12. We leverage on the Jaccard similarity to assess the similarity among documents, where an experimental threshold of 0.22 was set for our submission. Finally, in lines 17 and 18 we iterate the set of all clusters, to retrieve the document with the highest score from each cluster, and return the synthesized ranking.

## 4 EVALUATION

Each participant was limited to submit at most three runs for evaluation, in each scenario. For this edition of the track, relevance was measured with resort to the nDCGp@10 and nDCG1@10 variations of the nDCG@10 metric. These variations differ from nDCG@10, since they account for the submission of documents for uneventful

**Algorithm 1:** Single-pass Jaccard diversity grouping

---

```

input : A similarity threshold, jaccThresh, and a document
         ranking, R.
output: An ordered list, L, of the representatives with the
         highest score.
1 Clusters  $\leftarrow$  {}
2 counter  $\leftarrow$  0
3 rankSize  $\leftarrow$  Length(R)
4 while counter < rankSize do
5   Doc  $\leftarrow$  R[counter]
6   if isEmpty(Clusters) then
7     | Clusters  $\leftarrow$  {[Doc]}
8   else
9     if  $\forall c \in$  Clusters :
10      | JaccardCoef(getFirstDoc(c), Doc) > jaccThresh
11      | then
12      | | c  $\leftarrow$  c + Doc
13      | else
14      | | Clusters  $\leftarrow$  Clusters  $\cup$  {[Doc]}
15      | end
16    end
17    counter  $\leftarrow$  counter + 1
18 end
19  $\forall c1 \in$  Clusters : L  $\leftarrow$  L + getFirstDoc(c1)
20 return sort(L)

```

---

**Table 3:** Features present in each of our submitted runs

Run ID	Features				RMTS
	NER	Jitter	URL Exp.	Clustering	
1					✓
2	✓	✓	✓	✓	
3	✓		✓	✓	

days, for a given query. The nDCG1@10 metric is the strictest of the metrics, where the submission of one tweet during a uneventful day will result in a complete loss of the available bonus for not submitting during a silent day. Whereas for nDCGp@10 will produce a score loss based on a decay, with a limit to up to ten tweets.

## 4.1 Results and discussion

**4.1.1 Submitted runs.** Table 3 summarizes the features used in each of the submitted runs. Our first run consisted on a temporal-aware learning to rank model named RMTS, developed by Martins et al. [9], without the vocabulary expansions and clustering algorithm. Secondly, we submitted a run leveraging on the RRF document ranking with both query expansion based on PRF with NER title queries, and document expansions; and clustering of the document ranking. The third run also leveraged on the RRF document ranking, with both query expansion based on NER and document expansions; and clustering of the document ranking.

Observing Table 4 we can verify that both nDCGp and nDCG1 present the same value for Run 1. This is an expected result since

**Table 4:** Submitted runs results

Run ID	nDCGp@10	nDCG1@10
1	0.1896	0.1896
2	0.1440	0.1333
3	0.2710	0.2587

**Table 5:** The Best Configuration (BC) is composed of the LMD retrieval model, query expansion with NER, document expansion with URLs and results clustering. The impact of each component is illustrated in the different rows.

Method	Cumulative index 2017	
	nDCGp	nDCG1
TREC	0.2710	0.2587
BC	0.2831	0.2697
BC <sub>-URL</sub>	0.2675	0.2554
BC <sub>-NER</sub>	0.2746	0.2601
BC <sub>-Clustering</sub>	0.2812	0.2682

the Run 1 system does not have a strategy for grouping or non-submitting documents; hence being fully penalized by always submitting tweets on silent days. Run 2 is our run with most features, combining all the modules that we have developed. Whereas Run 3 only differs from Run 2, by not making use of the expansions provided by the Jitter API, as we can observe in Table 3. However, the results obtained by these two runs differ hugely, possibly indicating that the queries expansions provided by the Jitter API [9] were not properly tuned for this track, taking full advantage of the temporal ranking provided by the API.

The overall performance of our runs was positive, staying above the nDCGp and nDCG1 median, that were 0.1868 and 0.1155 respectively, with our best run being Run 3.

**4.1.2 Post-mortem experiments.** With the conclusion of the track and the publication of the relevance judgments, we made final experiments with our system, that assessed the impact of each of the components of our best run, and are presented in Table 5. The first line of the table shows our best submitted run as a comparison reference towards the other lines.

Through this experiment we have identified that the best configuration (BC) of our system was obtained through the utilization of the LMD retrieval model, with the tuned parameters; document and query expansion based on NER; and clustering the document ranking. As we can observe in the third line of Table 5, the document expansion with the title of webpages was our best investment, providing the greatest relevance loss when removed from the BC. The query expansion based on NER did provided a slight less loss, when removed from the BC. Indicating that it deserves further a more careful study on how can he be employed to enhance our results. Finally, and surprisingly, the clustering module was the one that had a trivial impact on the overall relevance. With this analysis we understood that there is still a lot of room for improvement in our subtopic detection strategy.

## 5 CONCLUSION

Our approach to solve the daily-digest summarization problem was composed of many features. It ended up being an exploratory work of these features, that delivered satisfactory results being above the median. From the vocabulary expansions, the document expansion with the webpage titles was the approach with the biggest impact. The query expansions both with NER and PRF, need further development; The NER approach can be better explored with the identification of named entities also in the tweet text. For the PRF expansion, leveraging on the Jitter API, careful tuning is needed to extract better results. The document retrieval using RRF did not generate the expected results, what indicates that a more careful review and tuning is needed.

Ultimately, through our participation in this track we have obtained an excellent insight about the dynamics of subtopics across time. The detection of subtle changes in documents sharing the same topic, is a difficult task that needs further research. Moreover the detection of such changes in real-time can be crucial to deliver immediately the most relevant content towards end users; Thus showing the significance of this task.

## REFERENCES

- [1] Mossaab Bagdouri and Douglas W. Oard. 2016. CLIP at TREC 2016: LiveQA and RTS. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [2] Chao Bei and Po Hu. 2016. CCNU at TREC 2016 Real-Time Summarization Track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [3] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. ACM, New York, NY, USA, 758–759. <https://doi.org/10.1145/1571941.1572114>
- [4] Kathy Lee, Ashequl Qadir, Vivek V. Datla, Sadid A. Hasan, Joey Liu, Aaditya Prakash, and Oladimeji Farri. 2016. Assorted Textual Features and Dynamic Push Strategies for Real-Time Tweet Notification. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [5] Song Li, Zhenyuan Hao, Zhongyuan Han, Leilei Kong, and Haoliang Qi. 2016. HLJIT at TREC 2016: The Approaches Based on Document Language Model for Real-Time Summarization Track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [6] Jimmy J. Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen M. Voorhees, and Fernando Diaz. 2016. Overview of the TREC-2016 Real-Time Summarization Track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees (Ed.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [7] Cristian Lumezanu and Nick Feamster. 2012. Observing Common Spam in Twitter and Email. In *Proceedings of the 12th ACM SIGCOMM Internet Measurement Conference, IMC '12, Boston, MA, USA, November 14-16, 2012*, John W. Byers, Jim Kurose, Ratul Mahajan, and Alex C. Snoeren (Eds.). ACM, 461–466. <https://doi.org/10.1145/2398776.2398824>
- [8] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*. The Association for Computer Linguistics, 55–60.
- [9] Flávio Martins, João Magalhães, and Jamie Callan. 2016. Barbara Made the News: Mining the Behavior of Crowds for Time-Aware Learning to Rank. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, Paul N. Bennett, Vanja Josifovski, Jennifer Neville, and Filip Radlinski (Eds.). ACM, 667–676. <https://doi.org/10.1145/2835776.2835825>
- [10] Richard McCreadie and Craig Macdonald. 2013. Relevance in Microblogs: Enhancing Tweet Retrieval Using Hyperlinked Documents. In *Open Research Areas in Information Retrieval, OAIR '13, Lisbon, Portugal, May 15-17, 2013*, João Ferreira, João Magalhães, and Pável Calado (Eds.). ACM, 189–196.
- [11] Sandip Modha, Krati Agrawal, Deepali Verma, Prasenjit Majumder, and Chintak Mandalia. 2016. DAICT at TREC RTS 2016: Live Push Notification and Email Digest. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [12] Bilel Moulahi, Lamjed Ben Jabeur, Abdelhamid Chellal, Thomas Palmer, Tamine Lynda, Mohand Boughanem, Karen Pinel-Sauvagnat, and Gilles Hubert. 2016. IRIT at TREC Real-Time Summarization 2016. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [13] Reem Suwaileh, Maram Hasanain, and Tamer Elsayed. 2016. Light-Weight, Conservative, yet Effective: Scalable Real-Time Tweet Summarization. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [14] Haihui Tan, Dajun Luo, and Wenjie Li. 2016. PolyU at TREC 2016 Real-Time Summarization. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [15] Kai Wang and Zhen Yang. 2016. BJUT at TREC 2016: Real-Time Summarization Track. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [16] Lili Yao, Chao Lv, Feifan Fan, Jianwu Yang, and Dongyan Zhao. 2016. PKUICST at TREC 2016 Real-Time Summarization Track: Push Notifications and Email Digest. In *Proceedings of The Twenty-Fifth Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, Ellen M. Voorhees and Angela Ellis (Eds.), Vol. Special Publication 500-321. National Institute of Standards and Technology (NIST).
- [17] ChengXiang Zhai and John D. Lafferty. 2004. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Trans. Inf. Syst.* 22, 2 (2004), 179–214. <https://doi.org/10.1145/984321.984322>