Overview of the TREC 2018 Real-Time Summarization Track

Royal Sequiera, Luchen Tan, and Jimmy Lin
David R. Cheriton School of Computer Science, University of Waterloo
jimmylin@uwaterloo.ca

1 INTRODUCTION

The TREC 2018 Real-Time Summarization (RTS) Track is the third iteration of a community effort to explore techniques, algorithms, and systems that automatically monitor streams of social media posts such as tweets on Twitter to address users' prospective information needs. These needs are articulated as "interest profiles", akin to topics in *ad hoc* retrieval. In our formulation of real-time summarization, the goal is for a system to deliver relevant and novel content to users in a timely fashion. We refer to these messages generically as "updates".

As with previous iterations of the evaluation, the task setup required participating systems to monitor the live Twitter sample stream during a pre-defined evaluation period, this year beginning Monday July 23, 2018 00:00:00 UTC and ending Friday August 3, 2018 23:59:59 UTC. The interest profiles were distributed to participants ahead of time. The RTS evaluation considered two methods for disseminating updates:

- **Scenario A: Real-time updates.** As soon as the system identifies a relevant tweet, it is immediately delivered to the user's mobile device (see Figure 1). These updates should be relevant (on topic), novel (users should not be delivered multiple notifications that say the same thing), and timely (updates should be provided as soon after the actual event occurrence as possible).
- **Scenario B: Email digests.** Alternatively, a user might wish to receive a daily email digest that summarizes "what happened" on that day with respect to the interest profiles (see Figure 2). One might think of these emails as supplying "personalized headlines". These results should be relevant and novel, but timeliness is not particularly important provided that the posts were all written on the day for which the digest was produced.

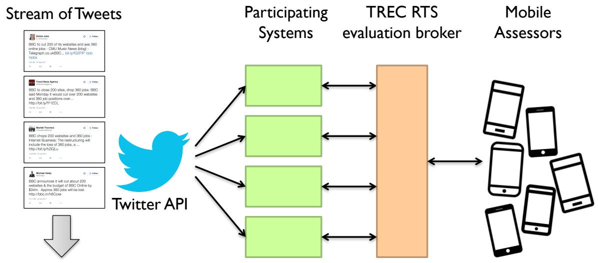


Figure 1: Evaluation setup for scenario A. Systems processed the Twitter sample stream in real time and submitted relevant tweets to the RTS evaluation broker, which immediately delivered the tweets to the mobile devices of assessors who had subscribed to those interest profiles.

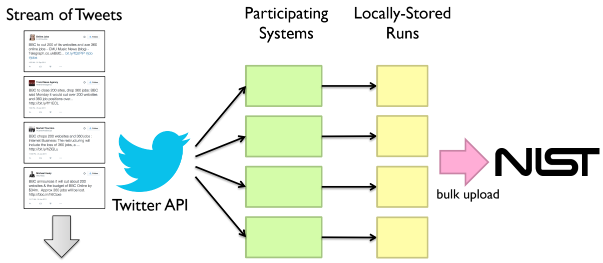


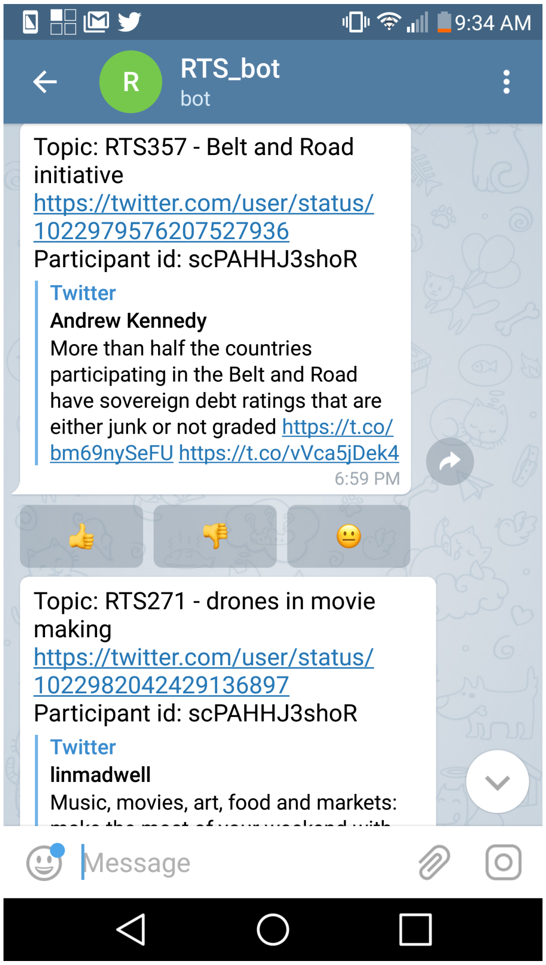
Figure 2: Evaluation setup for scenario B. Systems processed the Twitter sample stream in real time and stored their results locally during the evaluation period. After the evaluation ended, the runs were uploaded to NIST in batch.

2 EVALUATION DESIGN

The RTS evaluation at TREC 2018 followed the same methodology as the evaluation in TREC 2017 [2], except with two substantive changes, described below. Anything that is not explicitly discussed in this track overview can be assumed to have remained unchanged from last year.

Push vs. pull in mobile update delivery. For scenario A, participating systems subscribed to the live Twitter sample stream during the evaluation period to identify tweets relevant to interest profiles in real time. Based on each system's algorithm, these updates were submitted to the RTS evaluation broker (see Figure 1), which then immediately delivered the updates to a group of mobile assessors who were specifically recruited for the evaluation (framed as a user study). This *in situ* evaluation setup accurately mimicked the deployment of update delivery systems, since the assessors were simply going about their daily lives and could choose to either ignore or engage with systems' updates.

In 2016, updates were delivered to the assessors' mobile devices (via a custom app developed by the track organizers) and each update was accompanied by a push notification, an explicit alert that drew their attention to the update [3]. We refer to this as the "push" interface condition. In 2017, updates were delivered to the assessors' mobile devices (via a completely redesigned mobile web app), but each update was *not* accompanied by a push notification [2]. In other words, delivery was akin to depositing updates into an email inbox, and the assessor had to proactively (i.e., on the assessor's own initiative) visit a mobile interface to examine system updates. We refer to this as the "pull" interface condition. The behavior of mobile assessors under these two interface conditions was the subject of a SIGIR 2018 paper [1], which compared how assessors rendered their judgments in 2016 and 2017. However, an explicit weakness of the study was that it did not control for a number of factors such as the assessment interface and the interest profiles, since the user behavior data drew from two different iterations of the RTS Track.



Topic: RTS357 - Belt and Road initiative
https://twitter.com/user/status/1022979576207527936
Participant id: scPAHHJ3shoR

Twitter
Andrew Kennedy
More than half the countries participating in the Belt and Road have sovereign debt ratings that are either junk or not graded https://t.co/bm69nySeFU https://t.co/vVca5jDek4
6:59 PM

Topic: RTS271 - drones in movie making
https://twitter.com/user/status/1022982042429136897
Participant id: scPAHHJ3shoR

Twitter
linmadwell
Music, movies, art, food and markets:

Figure 3: Sample screenshot of the evaluation interface using the Telegram messaging platform.

This year, we wished to more closely examine behavioral differences between "push" and "pull" mechanisms for notification delivery. This required rebuilding the update delivery infrastructure so that push notifications could be more carefully controlled and isolated as the experimental variable. After careful initial study, we decided to use the Telegram messaging platform.[1] Specifically, mobile assessors installed the Telegram messaging app on their mobile devices and updates were delivered via a Telegram bot we built called the RTS_bot. The assessors were asked to subscribe to the bot. Updates from participating systems, along with the associated interest profile (i.e., the information need), were delivered as messages in the Telegram app from our bot. Assessors provided judgments by clicking on buttons indicating that the tweet was relevant, not relevant, or redundant, which triggered corresponding API invocations that recorded the decision in a backend database. A screenshot from the interface is shown in Figure 3.

We randomly divided the mobile assessors into the "push" and "pull" interface conditions. Prior to the evaluation period, as part of the assessor onboarding process, we ensured that all participants in the pull condition switched their push notifications off in the app settings. That is, the delivery of a system update did not trigger a notification on their mobile devices. Instead, the assessors had to visit the messaging app on their own initiative to examine the delivered updates. For users in the push condition, we verified that push notifications did indeed accompany update delivery. In this way, we carefully manipulated the mobile interface such that the presence or absence of push notifications was the only difference, corresponding to our variable of interest.

Disjoint interest profiles between mobile assessors and NIST assessors. In both the 2016 and 2017 iterations of the RTS Tracks, scenario A systems were evaluated by *both* mobile assessors (i.e., using the *in situ* evaluation methodology described in Figure 1) as well as NIST assessors (i.e., using a traditional pool-based methodology). Scenario B (which modeled daily email digests) was only evaluated with judgments from NIST assessors.

However, a relevance feedback mechanism introduced in the 2017 evaluation presented a complication: In 2017, as well as in this year's evaluation, participating systems in scenario A were able to obtain the mobile assessors' relevance judgments as they were being generated during the evaluation period. This allowed systems to experiment with relevance feedback and techniques based on active learning, or otherwise alter the system's behavior. As a result, consistency between the mobile assessors and NIST assessors became an issue since participating systems could have been incorporating live feedback. To avoid divergence of relevance criteria, in the batch assessments, NIST assessors were asked to consider the mobile judgments and to maintain consistency if possible; see the track overview from last year [2] for the exact instructions given to the NIST assessors. Thus, potentially noisy judgments from the mobile assessors unavoidably affected the NIST judgments, but it is unclear to what extent conflicting relevance judgments affected their overall quality.

To avoid the same issue this year, the mobile assessors and NIST assessors examined system updates from completely disjoint interest profiles. The effect is that scenario A systems were only assessed by mobile assessors and scenario B systems were only assessed by NIST assessors.

3 RESULTS

In total, we had 142 interest profiles created by the mobile assessors (scenario A) and 156 profiles created by the NIST assessors (scenario B) this year.

For scenario A, we received a total of 14 runs from 6 groups. These runs submitted a total of 23,021 tweets, or 13,756 unique tweets after de-duplicating within each interest profile (but not across interest profiles). We recruited a total of 53 mobile assessors for the *in situ* evaluation of the scenario A systems: 27 were assigned to the "pull" condition and 26 were assigned to the "push" condition. Two assessors did not judge any tweets. Not including these two, the remaining assessors judged a total of 26,150 tweets; mean 513, median 469, max 1317. The distribution of judgments by assessor is shown in Table 1. The columns list: assessor id, the number of judgments provided, the number of profiles subscribed to, and the number of tweets delivered to that assessor. The final column shows

[1] https://telegram.org/

the response rate, computed as the ratio between the second and fourth columns.

Evaluation results for scenario A systems by the mobile assessors are shown in Table 2. The metrics and presentation is exactly the same as last year [2]. The first two columns show the participating team and run. The next columns show the number of tweets that were judged relevant (R), redundant (D), and not relevant (N); the number of unjudged tweets (U); the length of each run (L), defined as the total number of messages delivered by the system. The next column shows coverage (C), defined as the fraction of *unique* tweets that were judged. Following that, the columns report the mean ($\bar{t}$) and median ($\tilde{t}$) latency of submitted tweets in seconds, measured with respect to the time the original tweet was posted. The next sets of columns provide metrics of quality: strict and lenient precision, strict and lenient utility. The rows in the results table are sorted by strict precision.

For scenario B, we received a total of 11 runs from 4 groups. Evaluation results based on NIST assessors are shown in Table 3. The metrics and presentation is exactly the same as last year [2], with runs sorted by nDCG-p. For reference, the empty run would have received nDCG-p and nDCG-1 scores of 0.7557.

4 CONCLUSIONS

This year represents the final year of the Real-Time Summarization Track, the culmination of a journey that began with the Microblog Track at TREC 2011. During these eight years, the tracks have explored numerous aspects of users' information needs as they pertain to social media posts, with tweets on Twitter as representatives of such texts. Over the years, the tracks have brought together researchers with common interests, built test collections, and contributed to evaluation methodologies. However, with diminishing marginal benefits to the broader community, we believe it is time to make way for different avenues of exploration at TREC.

5 ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada. We'd like to thank Yinan Zhang for help in data analysis.

REFERENCES

[1] Jimmy Lin, Salman Mohammed, Royal Sequiera, and Luchen Tan. 2018. Update Delivery Mechanisms for Prospective Information Needs: An Analysis of Attention in Mobile Users. In *Proceedings of the 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018)*. Ann Arbor, Michigan, 785–794.

[2] Jimmy Lin, Salman Mohammed, Royal Sequiera, Luchen Tan, Nimesh Ghelani, Mustafa Abualsaud, Richard McCreadie, Dmitrijs Milajevs, and Ellen Voorhees. 2017. Overview of the TREC 2017 Real-Time Summarization Track. In *Proceedings of the Twenty-Sixth Text REtrieval Conference (TREC 2017)*. Gaithersburg, Maryland.

[3] Jimmy Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen 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.

Assessor	Condition	Profiles	Judgments	Messages	Response
RTS18_A001	pull	9	85	221	38.46%
RTS18_A002	push	12	469	863	54.35%
RTS18_A003	pull	9	221	703	31.44%
RTS18_A004	pull	10	180	279	64.52%
RTS18_A005	push	11	1248	2352	53.06%
RTS18_A006	pull	11	338	470	71.91%
RTS18_A007	pull	11	85	982	8.66%
RTS18_A008	pull	9	1305	2066	63.17%
RTS18_A009	push	11	671	1258	53.34%
RTS18_A010	push	11	1317	2251	58.51%
RTS18_A011	pull	10	1212	2013	60.21%
RTS18_A012	pull	10	518	990	52.32%
RTS18_A013	push	11	511	924	55.3%
RTS18_A014	push	9	742	1283	57.83%
RTS18_A015	push	10	1121	1867	60.04%
RTS18_A016	pull	11	530	957	55.38%
RTS18_A017	pull	11	602	2369	25.41%
RTS18_A018	pull	11	844	1441	58.57%
RTS18_A019	push	11	321	634	50.63%
RTS18_A020	pull	11	351	600	58.5%
RTS18_A021	pull	11	350	617	56.73%
RTS18_A022	push	11	26	399	6.52%
RTS18_A023	pull	12	58	645	8.99%
RTS18_A024	pull	10	157	821	19.12%
RTS18_A025	push	11	1	781	0.13%
RTS18_A026	push	11	506	1371	36.91%
RTS18_A027	pull	10	804	1639	49.05%
RTS18_A028	pull	13	315	589	53.48%
RTS18_A029	push	10	1310	2093	62.59%
RTS18_A030	pull	13	152	466	32.62%
RTS18_A031	push	11	685	1167	58.7%
RTS18_A032	push	11	19	402	4.73%
RTS18_A033	push	12	1115	1920	58.07%
RTS18_A034	push	10	416	657	63.32%
RTS18_A035	pull	11	20	818	2.44%
RTS18_A036	pull	9	0	890	0.0%
RTS18_A037	push	11	495	1287	38.46%
RTS18_A038	push	10	376	726	51.79%
RTS18_A039	pull	11	1088	1834	59.32%
RTS18_A040	push	11	1122	2007	55.9%
RTS18_A041	pull	10	3	675	0.44%
RTS18_A042	push	10	381	719	52.99%
RTS18_A043	pull	9	0	508	0.0%
RTS18_A044	pull	15	885	1606	55.11%
RTS18_A045	push	11	604	1219	49.55%
RTS18_A046	pull	11	168	917	18.32%
RTS18_A047	push	13	597	1030	57.96%
RTS18_A048	pull	10	791	1397	56.62%
RTS18_A049	push	10	4	337	1.19%
RTS18_A050	pull	10	152	771	19.71%
RTS18_A051	push	10	575	950	60.53%
RTS18_A052	push	11	17	489	3.48%
RTS18_A053	push	10	287	410	70.0%

Table 1: Summary of assessor statistics. For each assessor, columns show the interface condition, the number of interest profiles the assessor subscribed to, the number of judgments provided, the number of tweets delivered to that assessor, and the response rate.

team	run	R	D	N	U	L	C	$\bar{\tau}$	$\tilde{\tau}$	P_s	P_l	Util_s	Util_l
umd_hcil	primary_run-16	21	0	2	0	8	1.0	4716.9	1065.0	0.913	0.913	19	19
IRIT	IRIT-Run3-08	3115	84	2385	62	1836	0.966	328.8	34.0	0.5578	0.5729	646	814
UA_GPLSI	GPLSI-runA1-13	4011	50	3171	74	2380	0.969	6.1	1.0	0.5546	0.5615	790	890
UA_GPLSI	GPLSI-runA3-15	2844	29	2465	56	1730	0.968	7.9	1.0	0.5328	0.5382	350	408
IRIT	IRIT-Run1-06	3226	93	3112	90	2182	0.959	316.6	33.0	0.5016	0.5161	21	207
UA_GPLSI	GPLSI-runA2-14	2416	30	2525	54	1610	0.966	9.4	1.0	0.486	0.4921	-139	-79
BJUT	BJUT_run2-A-04	673	52	664	33	532	0.938	272490.4	368638.0	0.4845	0.522	-43	61
ldrpitr	ldrpitr_Run2-12	180	0	200	0	140	1.0	511.8	367.5	0.4737	0.4737	-20	-20
IRIT	IRIT-Run2-07	3507	71	4337	61	2579	0.976	270.0	32.0	0.4431	0.4521	-901	-759
BJUT	BJUT_run1-A-03	431	11	589	12	367	0.967	11159.7	63.0	0.418	0.4287	-169	-147
LDRP	ldrpTest-09	88	2	121	3	80	0.963	12.3	1.0	0.4171	0.4265	-35	-31
ldrpitr	ldrpitrTest-11	4	0	8	0	4	1.0	447.8	447.5	0.3333	0.3333	-4	-4
LDRP	ldrpTest-10	53	0	186	8	117	0.932	2.0	1.0	0.2218	0.2218	-133	-133
BJUT	BJUT_run3-A-05	201	7	721	4	331	0.988	207993.7	323364.0	0.2164	0.2239	-527	-513

Table 2: Evaluation of scenario A runs by the mobile assessors. The first two columns show the participating team and run. The next columns show the number of tweets that were judged relevant (R), redundant (D), and not relevant (N); the number of unjudged tweets (U); the length of each run (L), defined as the total number of messages delivered by the system. The next columns show coverage (C), defined the fraction of *unique* tweets that were judged; the mean ($\bar{t}$) and median ($\tilde{t}$) latency of submitted tweets in seconds, measured with respect to the time the original tweet was posted; strict and lenient precision; strict and lenient utility. Rows are sorted by strict precision.

team	run	nDCG-p	nDCG-1
IRIT	IRIT-RunB3	0.8449	0.7411
GPLSI	IR-NR2	0.8387	0.7554
GPLSI	IR-NR1	0.8379	0.7561
GPLSI	IR-N	0.8378	0.7561
IRIT	IRIT-Run2	0.8294	0.6613
IRIT	IRIT-Run1	0.8287	0.6614
BJUT	bjut-run1	0.7793	0.7536
BJUT	bjut-run2	0.7792	0.7536
BJUT	bjut-run3	0.7788	0.7531
IRLAB	IRLAB-LDRP2-Run2	0.3219	0.2902
IRLAB	IRLAB-LDRP2-Run1	0.3062	0.2816

Table 3: Evaluation of scenario B runs by NIST assessors. Rows are sorted by nDCG-p.