Overview of the TREC-2015 Microblog Track

Jimmy Lin,[1] Miles Efron,[2] Yulu Wang,[3] Garrick Sherman,[2] and Ellen Voorhees[4]

[1] University of Waterloo [2] University of Illinois, Urbana-Champaign
[3] University of Maryland, College Park [4] NIST

jimmylin@uwaterloo.ca, {mefron, gsherma2}@illinois.edu, ylwang@cs.umd.edu, Ellen.Voorhees@nist.gov

1. INTRODUCTION

The TREC 2015 Microblog track introduced a single real-time filtering task broken down into two scenarios. Our goal is to explore techniques for monitoring streams of social media posts with respect to users' interest profiles. An interest profile describes a topic about which the user wishes to receive information updates in real time, and is different from a typical *ad hoc* topic in that the profile represents a prospective (as opposed to a retrospective) information need. Thus, the nature of the desired information is qualitatively different. In real-time filtering, the goal is for a system to "push" (i.e., recommend, suggest) interesting and novel content to users in a timely fashion.

We operationalized this task in terms of two scenarios:

- **Scenario A (push notification):** Content that is identified as interesting and novel by a system based on the user's interest profile might be shown to the user as a notification on his or her mobile phone. The expectation is that such notifications are triggered a relatively short time after the content is generated.
- **Scenario B (email digest):** Content that is identified as interesting and novel by a system based on the user's interest profile might be aggregated into an email digest that is periodically sent to a user (e.g., nightly). It is assumed that each item of content is relatively short; one might think of these as "personalized headlines".

In both scenarios, it is assumed that the content items delivered to the users are relatively short. For expository convenience and to adopt standard information retrieval parlance, we write of users desiring *relevant* content, even though "relevant" in our context might be better operationalized as interesting, novel, and timely.

2. EVALUATION DESIGN

2.1 General Setup

Although we are interested in exploring filtering techniques over streams of social media posts in general, the Microblog track restricted the content under consideration to tweets. In particular, Twitter provides a streaming API through which clients can obtain a sample (approximately 1%) of public tweets—this level of access is available to anyone who signs up for an account.

During the official evaluation period, which began Monday, July 20, 2015, 00:00:00 UTC and lasted until Wednesday, July 29, 2015, 23:59:59 UTC, participants' systems "listened" to Twitter's live tweet sample stream to identify relevant tweets with respect to users' interest profiles. The identified tweets were recorded by the participants' systems and submitted to NIST shortly after the conclusion of the evaluation period. Although systems were expected to conform to the temporal constraints imposed by the task scenarios (more details below), there was no enforcement mechanism because of the batch submission setup.

An important consequence of the evaluation setup is that, unlike in most TREC evaluations, no collection was distributed ahead of time. Since each participant "listened" to tweets from Twitter's streaming API, the collection was generated in real-time and delivered to each participant independently. In a separate pilot study [5], we verified that multiple listeners to the public Twitter sample stream receive effectively the same tweets (Jaccard overlap of 0.999 across six independent crawls over a three day sample in March 2015). For evaluation purposes (i.e., pool formation for judgments) the stream collected by the organizers was arbitrarily designated as the "official" collection.

Another substantial departure from most previous TREC evaluations is the requirement that participants maintain a running system that continuously monitors the tweet sample stream during the evaluation period. The track organizers provided boilerplate code and reference implementations, but it was the responsibility of each individual team to run its own system and cope with crashes, network glitches, power outages, etc.

Additional details for each of the task scenarios are provided below:

- **Scenario A (push notification):** A system for this scenario (a "type A" system) is allowed to return a maximum of ten tweets per day per interest profile (and of course may choose to return fewer than ten tweets). Each day of the evaluation is defined in terms of the time range 00:00:00 to 23:59:59 UTC. Additional tweets beyond ten per day are simply ignored in computing evaluation metrics. Given the real-time nature of the task, each system was asked to record the time at which a tweet was putatively delivered as a push notification; this information is used to compute a latency penalty (more details later).

  The per-day tweet delivery limit represents a crude attempt to model user fatigue in mobile push notifications. Note, however, that in this design we are not modeling real-world constraints such as "don't send users notifications in the middle of the night". This simplification is intentional.

- **Scenario B (email digest).** A system for this scenario (a "type B" system) is tasked with identifying a batch of up to 100 ranked tweets per day (per interest profile). In our task model, these tweets are delivered to the user daily. For simplicity, all tweets from 00:00:00 to 23:59:59 UTC are valid candidates for a particular day. It is expected that systems will compute the results in a relatively short amount of time after the day ends (e.g., at most a few hours), but this constraint was not enforced.

It is worth mentioning that despite superficial similarities, our task is very different from document filtering in the context of earlier TREC Filtering tracks, which ran from 1995 [3] to 2002 [6], and the general research area of topic detection and tracking (TDT) [1]. The TREC Filtering tracks are best understood as binary classification on *every* document in the collection with respect to standing queries, and TDT is similarly concerned with identifying *all* documents related to a particular event—with an intelligence analyst in mind. In contrast, we are focused on identifying a small set of the most relevant updates to deliver to users. Furthermore, in both TREC Filtering and TDT, systems must make online decisions as soon as documents arrive. In our case, systems can choose to push older content (subjected to the latency penalty), thus giving rise to the possibility of algorithms operating on bounded buffers. Finally, previous evaluations merely *simulated* the streaming nature of the document collection, whereas participants in our evaluation actually operated on tweets posted in real time.

2.2 Run Submission

In both scenarios, systems were asked to only consider tweets in English. Each team was allowed to submit up to three runs for scenario A and three runs for scenario B. Systems for either scenario were categorized into three different types based on the amount of human involvement:

- **Automatic Runs:** In this condition, system development (including all training, system tuning, etc.) must conclude prior to downloading the interest profiles from NIST (which were made available before the evaluation period). The system must operate without human input before and during the evaluation period. Note that it is acceptable for a system to perform processing on the profiles (for example, query expansion) before the evaluation period, but such processing cannot involve human input.
- **Manual Preparation:** In this condition, the system must operate without human input during the evaluation period, but human involvement is acceptable before the evaluation period (i.e., after downloading the interest profile). Examples of manual preparation include human examination of the interest profiles to add query expansion terms or manual relevance assessment on a related collection to train a classifier. However, once the evaluation period begins, no further human involvement is permissible.
- **Manual Intervention:** In this condition, there are no limitations on human involvement before or during the evaluation period. Crowd-sourcing judgments, human-in-the-loop search, etc. are all acceptable.

Participants were asked to designate the run type at submission time. All types of systems were welcomed; in particular, manual preparation and manual intervention runs are helpful in understanding human performance and enriching the judgment pool.

2.3 Interest Profiles

Our initial idea was to develop interest profiles that consisted of short English statements of information needs and a few example tweets. Upon further reflection, the history of previous TREC Filtering tracks suggested that this approach would have been problematic. The initial effectiveness of filtering systems is quite poor as the systems need to retrieve at least some non-relevant documents to learn to distinguish between relevant and non-relevant cases. Since there was no possibility of providing assessor feedback in the track setup this year and the few example tweets were guaranteed to be an incomplete, highly-biased sample of relevant tweets, systems would get caught in this initial effectiveness "trough" since they lack guidance in the form of incremental relevance judgments.

Instead, we adopted the "standard" TREC topic format of "title", "description", and "narrative" for the interest profiles. The so-called title consists of two to three keywords that provide the gist of the information need, akin to something a user might type into the query box of a search engine. The description is a one-sentence statement of the information need, and the narrative is a paragraph-length chunk of prose that sets the context of the need and expands on what makes a tweet relevant. By necessity, these interest profiles are more generic than the needs expressed in typical retrospective topics because the topic developer does not know what future events will occur. Thus, despite superficial similarities in format, we believe that interest profiles are qualitatively different from *ad hoc* topics.

Three sample profiles were released to participants prior to the evaluation period and were also provided to the NIST assessors as templates for profile development. Assessors were not required to use the samples as templates; they could (and did) use their own ideas as well. Assessors also performed web searches to find events that would happen during the evaluation period and constructed profiles targeting those events. Since profile development happened long before the evaluation period, we could not know which interest profiles would ultimately be appropriate for evaluation—we desired profiles that had neither too many nor too few relevant tweets (since the former would cause excessive evaluation burden and the latter would complicate system development). Thus, we created many (225) profiles as the test set, with the intention of culling a smaller set of around 50 profiles as the evaluation set.

Contrary to expectations, pooling statistics (how many documents retrieved per run, how much overlap in the retrieved sets across runs, etc.) did not provide any signal as to which profiles should be included in the evaluation set. We examined the statistics for a profile that could not possibly have any relevant tweets (terrorist activity on Bastille day; there wasn't any such activity) and could not distinguish that profile from the statistics for other profiles. During the evaluation, NIST (manually) monitored the news looking for stories that matched the interest profiles and was able to identify 12 profiles that definitely did have activity during the evaluation period (but not necessarily in the tweet stream); these profiles were added to the evaluation set. Beyond that, assessors were simply asked to pick profiles to judge from the set they had created based on their own interests. In the end, 51 interest profiles were judged. Of those, four have zero relevant tweets, and 47 have three or more relevant tweets. The largest number of relevant tweets is 1543, though the next largest is 583; a total of six profiles have more than 300 relevant tweets. We specifically retained profiles with no relevant tweets in the final evaluation set to test systems' ability to recognize and handle information needs with no relevant content.

2.4 Judgments and Metrics

The assessment workflow was modeled after the TREC 2014 Microblog track [4] and proceeded in two major stages: relevance assessment and semantic clustering.

Relevance assessments were performed using pooling with a single pool across both scenario A and scenario B runs. Scenario A runs contributed a maximum of ten tweets per day to the pools; while this should have been their 'entirety', some systems returned (many) more than this limit (contrary to track guidelines). Scenario B runs contributed no more than 85 documents for each interest profile; tweets were added to the judgment pool in a round-robin fashion across days. That is, the top-ranked tweet from each day was first added to the pool, then the second-ranked tweet from each day, and so on. If we ran out of tweets from a particular day before the 85 limit had been reached, tweets were selected from the remaining days until the limit.

This year, NIST assessors judged tweets in chronological order in the pools, not clustered by textual similarity as in previous years.[1] Each tweet was independently assessed on a three-way scale of "not relevant", "relevant", and "highly relevant". Non-English tweets were marked as not relevant by fiat. If a tweet contained a mixture of English and non-English content, discretion was left to the assessor. As with previous TREC Microblog evaluations, assessors examined links embedded in tweets, but did not explore any additional external content beyond those.

After the NIST assessors conducted relevance judgments using the pooling procedure described above, six assessors at the University of Illinois, Urbana-Champaign (graduate students in Library and Information Science) proceeded to perform semantic clustering on the relevant tweets using the protocol from the tweet timeline generation (TTG) task from the TREC 2014 Microblog track [4, 7]. While it is not ideal that the tweets were clustered by a different group of assessors than those who performed the relevance assessments (just like last year), this was a necessary compromise due to NIST resource limitations.

The TTG protocol was specifically designed to reward novelty (or equivalently, to penalize redundancy) in system output. In both scenario A and scenario B, we assume that users would not want to see multiple tweets that "say the same thing", and thus the evaluation methodology should reward systems that eliminate redundant output. Following the TREC 2014 Microblog track, we operationalized redundancy as follows: for every pair of tweets, if the chronologically later tweet contains substantive information that is not present in the earlier tweet, the later tweet is considered novel; otherwise, the later tweet is redundant with respect to the earlier one. In our definition, redundancy and novelty are antonyms, so we use them interchangeably but in opposite contexts.

Grade	Description
0	not relevant
1	relevant
2	highly relevant
-1	not relevant (propagated)
3	relevant (propagated)
4	highly relevant (propagated)

Table 1: Interpretation of qrels.

Due to the temporal constraint, redundancy is *not* symmetric. If tweet A precedes tweet B and tweet B contains substantively similar information found in tweet A, then B is redundant with respect to A, but not the other way around. We also assume transitivity. Suppose A precedes B and B precedes C: if B is redundant with respect to A and C is redundant with respect to B, then by definition C is redundant with respect to A.

In the TTG protocol, relevant tweets (from the judgment pool) for an interest profile were presented to a human assessor (from UIUC) in chronological order inside a JavaScript annotation interface. For each tweet, the assessor can add it to an existing cluster of semantically equivalent tweets or create a new cluster. Thus, the output of the assessment process (for each interest profile) is a list of clusters where tweets in each cluster represent a semantic equivalence class. Within each cluster, the earliest tweet is novel; all subsequent tweets are redundant with respect to all earlier tweets.

2.4.1 *Treatment of Retweets*

In previous TREC Microblog tracks, retweets were treated as not relevant by fiat. A consequence of this decision is that systems are not able to effectively take advantage of the retweet signal (i.e., number of retweets). If retweets are considered not relevant and a system observes a highly-retweeted tweet, the original underlying tweet must be part of the sample stream to be a valid result. This is unlikely due to sampling. Thus, in this year's evaluation, retweets were treated the same as any other tweet. Note that for retweets, the temporal penalty (detailed later) is computed with respect to the tweet itself, not the underlying tweet that was retweeted. To be clear, if A is a retweet of B, and the system returns tweet A, the temporal penalty is computed based on the timestamp of A, not B.

To reduce the sparsity of relevance judgments, we performed label propagation on the retweets as follows: First, the judgment of the retweet was propagated to the underlying tweet that was retweeted. Thus, if an assessor judged tweet t_1 and t_1 is a retweet of tweet s_1 (with or without additional commentary), then s_1 received the judgment of t_1. This label was then propagated to all other retweets of that tweet; that is, other tweets $t_2, t_3, t_4, \ldots$ that were also retweets of s_1 received its label. In cases where this label propagation yielded conflicts, we took the most common label, and in the case of ties, we broke them in favor of the higher relevance grade. The interpretation of the "relevance grade" in the official qrels file (i.e., the third column) is shown in Figure 1. In total, propagation of retweets led to an additional 2042 judgments across 25 interest profiles.

[1] However, the assessors indicated that they would have preferred textual similarity clustering after the fact.

2.4.2 Scenario A Metrics

To assess scenario A runs, we compute two temporally-discounted gain measures (explained in detail below) for each interest profile for each day in the evaluation period. The score for an interest profile is the average of the daily scores in the evaluation period. The score of a run is the average of the scores across all interest profiles.

The first (and primary) metric is expected latency-discounted gain (ELG) adapted from the TREC temporal summarization track [2]:

$$\frac{1}{N}\sum \mathrm{G}(t) \tag{1}$$

where N is the number of tweets returned and $\mathrm{G}(t)$ 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.

As with the TTG task, once a tweet from a cluster is retrieved, all other tweets from the same cluster automatically become not relevant. This penalizes systems for returning redundant information.

Furthermore, a latency penalty is applied to all tweets, computed as $\mathrm{MAX}(0, (100-d)/100)$, where the delay d is the time elapsed (in minutes, rounded down) between the tweet creation time and the putative time the tweet was delivered. That is, if the system delivers a relevant tweet within a minute of the tweet being posted, the system receives full credit. Credit decays linearly such that after 100 minutes, the system receives no credit even if the tweet is relevant. Lacking any empirical guidance, the linear decay and the 100 minute threshold represented arbitrary decisions made by the organizers.

The second metric is normalized cumulative gain (nCG):

$$\frac{1}{\mathcal{Z}}\sum \mathrm{G}(t) \tag{2}$$

where $\mathcal{Z}$ is the maximum possible gain (given the ten tweet per day limit). The gain of each individual tweet is computed as above. Note that gain is not discounted (as in nDCG) because the notion of document ranks is not meaningful in this context.

Finally, for both ELG and nCG, we handle days in which there are no relevant tweets as a special case, which we detail in Section 2.4.4.

2.4.3 Scenario B Metric

Scenario B runs were evaluated in terms of nDCG as follows: for each interest profile, the list of tweets returned per day is treated as a ranked list and from this nDCG@10 is computed. Note that in this scenario, the evaluation metric *does* include gain discounting because the email digests can be interpreted as ranked lists of tweets. The score of an interest profile is the average of the nDCG@10 scores across all days in the evaluation period, and the score of the run is the average over all profiles. See additional discussion below for scoring days without any relevant tweets.

2.4.4 Additional Details and Corner Cases

For simplicity, the TTG clustering protocol was applied to all tweets for a particular interest profile across the entire evaluation period (as opposed to each day separately). Thus, it is possible that a cluster spans multiple days. However, if tweet t_1 and tweet t_2 are in the same cluster, but different days, t_2 is still considered redundant if a run has already returned t_1.

Due to the setup of the task and the nature of interest profiles, it is possible (and indeed observed empirically) that for some days, no relevant tweets appear in the judgment pool. In terms of evaluation metrics, a system should be rewarded for correctly identifying this case and not generating any output. We can break down scoring for scenario A (both ELG and nCG) and scenario B as follows.

If there are relevant tweets for a particular day:

> . . . and the system returns zero tweets: the system receives a score of zero.
>
> . . . and the system returns any number of tweets: the score is computed normally per above.

If there are *no* relevant tweets for that day:

> . . . and the system returns zero tweets: the system receives a score of one (i.e., perfect score).
>
> . . . and the system returns any number of tweets: the system receives a score of zero.

This means that an empty run that never returns anything may have a non-zero score.

It is not clear if this treatment appropriately captures the task model for the purpose of guiding system development. On the one hand, push notifications are associated with high cognitive effort since they may interrupt the user. From that perspective, we should reward systems for not generating spurious output. On the other hand, on a day for which there are no relevant tweets, the only possible scores are one or zero. Having such highly binarized scores creates many issues for system tuning since it creates discontinuities in the objective. We have no guidance on how to properly balance these two competing concerns.

3. RESULTS

In total, we received 37 runs from 14 groups for scenario A and 42 runs from 16 groups for scenario B. Scenario A results are shown in Table 2 and scenario B results are shown in Table 3. For reference, an empty run that does not return anything receives an ELG score of 0.2471 and an NCG score of 0.2471.

Figure 1 shows a heatmap of the distribution of all relevant and highly-relevant tweets: each column corresponds to an interest profile and each row corresponds to a day in the evaluation period. For this visualization, we left out propagated retweets. Figure 2 is organized in the same manner, but we only show the first tweet in each cluster.

4. CONCLUSIONS

To our knowledge, the TREC Microblog track in 2015 represented the first attempt by the information retrieval community to operationalize push notification and email digest into a concrete task to facilitate multi-system comparisons. Overall, this has been a good learning experience for us and the participants.

Next year, the Microblog track will merge with the Temporal Summarization track to form the Real-Time Summarization (RTS) track at TREC 2016. The creation of

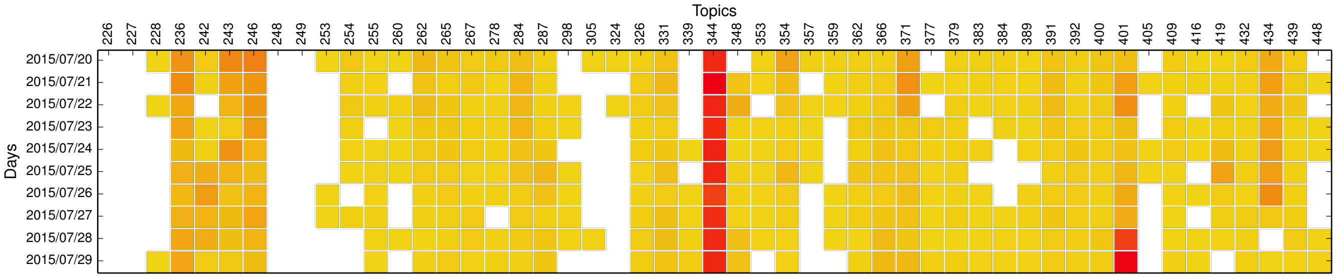

Figure 1: Heatmap of the distribution of all relevant and highly-relevant tweets (leaving out retweet propagation): interest profiles in columns, days of the evaluation in rows.

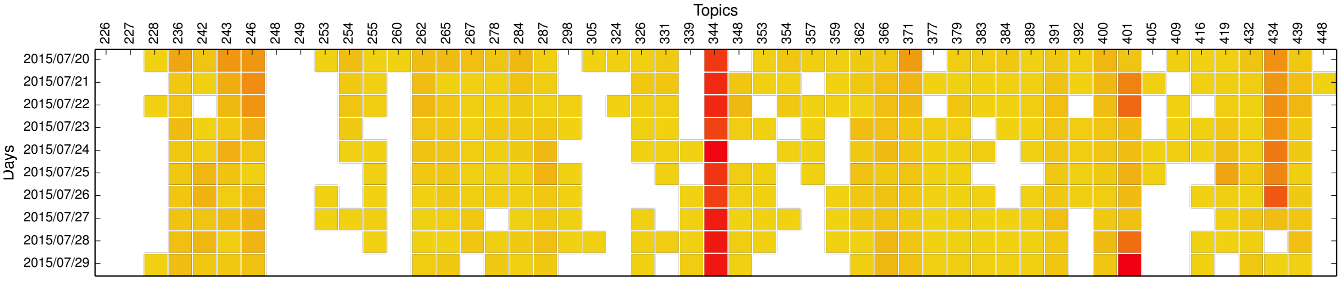

Figure 2: Heatmap of the distribution of the first tweet in each semantic cluster: interest profiles in columns, days of the evaluation in rows.

RTS was designed to leverage synergies between the two tracks in exploring prospective information needs over document streams containing novel and evolving information. By "joining forces", we plan to develop more refined push notification and email digest tasks, along with associated evaluation infrastructure to support online user-in-the-loop evaluations. Our hope is that these activities will grow the research community and push forward the state of the art.

5. ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation under IIS-1217279, IIS-1218043, and CNS-1405688. Additional support was provided by the Natural Sciences and Engineering Research Council of Canada. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the sponsor.

6. REFERENCES

[1] J. Allan. *Topic Detection and Tracking: Event-Based Information Organization.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[2] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, R. McCreadie, and T. Sakai. TREC 2014 Temporal Summarization Track overview. In *Proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014)*, Gaithersburg, Maryland, 2014.

[3] D. D. Lewis. The TREC-4 Filtering Track. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 165–180, Gaithersburg, Maryland, 1995.

[4] J. Lin, M. Efron, Y. Wang, and G. Sherman. Overview of the TREC-2014 Microblog Track. In *Proceedings of the Twenty-Third Text REtrieval Conference (TREC 2014)*, Gaithersburg, Maryland, 2014.

[5] J. H. Paik and J. Lin. Do multiple listeners to the public twitter sample stream receive the same tweets? In *Proceedings of the SIGIR 2015 Workshop on Temporal, Social and Spatially-Aware Information Access*, Santiago, Chile, 2015.

[6] S. Robertson and I. Soboroff. The TREC 2002 Filtering Track report. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, Maryland, 2002.

[7] Y. Wang, G. Sherman, J. Lin, and M. Efron. Assessor differences and user preferences in tweet timeline generation. In *Proceedings of the 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015)*, pages 615–624, Santiago, Chile, 2015.

Run	Group	ELG	nCG	Type
PKUICSTRunA2	PKUICST	0.3175	0.3127	manual intervention
UWaterlooATDK	UWaterlooMDS	0.3150	0.2679	automatic
SNACSA	NUDTSNA	0.3086	0.3349	automatic
SNACS_LA	NUDTSNA	0.2863	0.2974	automatic
QUBaseline	QU	0.2750	0.2347	automatic
udelRun2A	udel	0.2670	0.2064	automatic
UWaterlooATEK	UWaterlooMDS	0.2654	0.2365	automatic
IRIT-KLTFIDF	IRIT	0.2652	0.2600	manual
prnaTaskA2	prna	0.2603	0.2296	automatic
prnaTaskA1	prna	0.2597	0.2348	automatic
prnaTaskA3	prna	0.2566	0.2289	automatic
udelRun1A	udel	0.2505	0.2070	automatic
hpclab_pi_algA	HPCLAB_PI	0.2477	0.2472	manual
umd_hcil_run01	umd_hcil	0.2471	0.2471	automatic
UWaterlooATNDEK	UWaterlooMDS	0.2470	0.2170	automatic
UWCMBP1	WaterlooClarke	0.2450	0.2035	automatic
ECNURUNA1	ECNU	0.2314	0.2314	automatic
ECNURUNA2	ECNU	0.2314	0.2314	automatic
ECNURUNA3	ECNU	0.2314	0.2314	automatic
udelRun3A	udel	0.2259	0.1910	automatic
IritSigSDA	IRIT	0.2122	0.2043	automatic
umd_hcil_run02	umd_hcil	0.2020	0.2020	automatic
IRIT-RTNotif.33	IRIT	0.1950	0.1834	automatic
QUDyn	QU	0.1850	0.1762	automatic
QUDynExp	QU	0.1848	0.1763	automatic
DALTRECMA2	DalTREC	0.1822	0.1814	manual
CLIP-A-DYN-0.5	CLIP	0.1753	0.2426	automatic
DALTRECMA1	DalTREC	0.1620	0.1614	manual
CLIP-A-5.0-0.5	CLIP	0.1552	0.2193	automatic
CLIP-A-5.0-0.6	CLIP	0.1543	0.2221	automatic
DALTRECAA1	DalTREC	0.1447	0.1473	automatic
PKUICSTRunA1	PKUICST	0.1415	0.1566	automatic
PKUICSTRunA3	PKUICST	0.1382	0.1711	automatic
UWCMBP2	WaterlooClarke	0.1296	0.1275	automatic
MPII_HYBRID_PW	MPII	0.1025	0.0777	automatic
MPII_LUC_SORT	MPII	0.0841	0.1700	automatic
MPII_COMB_SORT	MPII	0.0575	0.1104	automatic

Table 2: Results of the real-time filtering task for scenario A, showing each run with ELG and nCG scores. Rows are sorted by ELG.

Run	Group	nDCG@10	Type
SNACS_LB	NUDTSNA	0.3670	automatic
SNACS	NUDTSNA	0.3345	automatic
CLIP-B-0.6	CLIP	0.2491	automatic
umd_hcil_run03	umd_hcil	0.2471	automatic
CLIP-B-0.5	CLIP	0.2420	automatic
PKUICSTRunB3	PKUICST	0.2343	automatic
PKUICSTRunB2	PKUICST	0.2228	manual intervention
PKUICSTRunB1	PKUICST	0.2226	automatic
DALTREC_B_PREP	DalTREC	0.2210	manual
UWaterlooBT	UWaterlooMDS	0.2200	automatic
UWaterlooBTND	UWaterlooMDS	0.2196	automatic
CLIP-B-0.4	CLIP	0.2117	automatic
MPII_COM_MAXREP	MPII	0.2093	automatic
hpclabpibm25mod	HPCLAB_PI	0.2046	manual
UNCSILS_WRM	UNCSILS	0.2045	automatic
udelRun2B	udel	0.2026	automatic
umd_hcil_run04	umd_hcil	0.2020	automatic
udelRun1B	udel	0.1966	automatic
UNCSILS_HRM	UNCSILS	0.1902	automatic
UNCSILS_TRM	UNCSILS	0.1890	automatic
IRIT100KLTFIDF	IRIT	0.1784	manual
udelRun3B	udel	0.1778	automatic
IRIT-RTDig.33	IRIT	0.1680	automatic
ECNURUNB1	ECNU	0.1610	automatic
prnaTaskB2	prna	0.1463	automatic
ECNURUNB3	ECNU	0.1416	automatic
DALTRECAB1	DalTREC	0.1339	automatic
BJUTllyQE	BJUT	0.1334	automatic
IritSigSDB	IRIT	0.1329	automatic
ECNURUNB2	ECNU	0.1327	automatic
DALTRECMB1	DalTREC	0.1323	automatic
QUBaselineB	QU	0.1288	automatic
UWCMBE1	WaterlooClarke	0.1232	automatic
QUFullExpB	QU	0.1196	automatic
QUExpB	QU	0.1180	automatic
UWCMBE2	WaterlooClarke	0.1035	automatic
BjutNMF1	BJUT	0.1008	automatic
BjutNMF2	BJUT	0.0685	automatic
prnaTaskB1	prna	0.0641	automatic
prnaTaskB3	prna	0.0533	automatic
MPII_LUC_MART	MPII	0.0310	automatic
MPII_COMB_MART	MPII	0.0275	automatic

Table 3: Results of the real-time filtering task for scenario B, showing each run with nDCG@10 scores. Rows are sorted by nDCG@10.