A Study of Realtime Summarization Metrics

Matthew Ekstrand-Abueg
mattea@ccs.neu.edu
Northeastern University

Richard McCreadie
richard.mccreadie@glasgow.ac.uk
University of Glasgow

Virgil Pavlu
vip@ccs.neu.edu
Northeastern University

Fernando Diaz
fdiaz@microsoft.com
Microsoft Research

ABSTRACT

Unexpected news events, such as natural disasters or other human tragedies, create a large volume of dynamic text data from official news media as well as less formal social media. Automatic real-time text summarization has become an important tool for quickly transforming this overabundance of text into clear, useful information for end-users including affected individuals, crisis responders, and interested third parties. Despite the importance of real-time summarization systems, their evaluation is not well understood as classic methods for text summarization are inappropriate for real-time and streaming conditions.

The TREC 2013-2015 Temporal Summarization (TREC-TS) track was one of the first evaluation campaigns to tackle the challenges of real-time summarization evaluation, introducing new metrics, ground-truth generation methodology and dataset. In this paper, we present a study of TREC-TS track evaluation methodology, with the aim of documenting its design, analyzing its effectiveness, as well as identifying improvements and best practices for the evaluation of temporal summarization systems.

1. INTRODUCTION

Crisis events such as natural disasters or other human tragedies often precipitate massive interest from a wide variety of users [10]. Furthermore, the rise in popularity of e-newspapers and social media has resulted in vast quantities of event-related information being generated and consumed online. On the other hand, much of this published information is redundant, of poor quality, out-of-date or inaccurate [6]. Hence, the general public is increasingly relying on real-time summarization services to get a concise overview of an event, such as the BBC live news timeline, the CNN mobile application, or a government emergency alert system.

The aim of real-time summarization is to generate an updating summary for an event over time, containing all of the important information that the user might want to know, while avoiding information redundancy and latency in reporting. Because of the wide variety of information sources and the large volume of content, manually generating such summaries is costly. As such, there has been a recent upsurge in research into how to build such updating summaries automatically. Indeed, evaluation campaigns such as NTCIR's Temporal Information Access task[1] and the TREC Temporal Summarization[2] and Real-time Summarization[3] tracks are dedicated to supporting research in this area.

However, one of the key takeaway messages that has emerged from these evaluation campaigns is that robust evaluation of real-time summarization systems is difficult, and there is not yet a consensus on the best way to achieve it. In particular, although real-time summarization bears similarities to tasks such as information filtering [14] and topic tracking [1], the evaluation methods used in these other domains are not suitable for real-time summarization, since they do not consider content redundancy. Meanwhile, the text summarization community [16] has primarily focused on multi-document summarization (MDS). However, MDS evaluation makes three key assumptions that do not hold in the real-time summarization scenario: 1) input documents are processed offline as a batch (instead of as a stream); 2) input documents are known to be relevant to the topic; and 3) summaries are of fixed length. As a result, new evaluation methodologies and metrics are needed to effectively evaluate and compare real-time summarization systems.

The main evaluation campaign for real-time summarization was the TREC Temporal Summarization Track, which ran for three years between 2013 and 2015. This track introduced new evaluation metrics, a novel two-stage ground-truth generation methodology, and a new dataset, all with the aim of better enabling the evaluation of real-time summarization systems. However, to-date, the evaluation methodology used within the track has not been analyzed or empirically validated.

The contributions of this study are four-fold. First, we document the design of the evaluation methodology and the reasoning behind this design. Second, we analyze the effectiveness of this evaluation methodology in terms of agreement with user preferences and reusability. Third, based on the reusability analysis we propose and evaluate an enhancement to the methodology. Fourth, we discuss best practices for temporal summarization evaluation.

[1] https://sites.google.com/site/ntcirtemporalia/
[2] http://www.trec-ts.org/
[3] https://github.com/trecrts/trecrts-eval

2. RELATED WORK

Realtime summarization has been previously attempted, but generally on a smaller scale. For instance, early work attempting to summarize an event used that event's Wikipedia page as a gold standard summary [9]. Our work extends this methodology, but uses manual extraction of time-stamped *nuggets*, or key facts, from the entire Wikipedia edit stream rather than the final page created. In addition to this previous work, we will briefly review prior work in evaluation of summarization and discuss why realtime summarization requires novel evaluation metrics.

Multidocument summarization (MDS) refers to the batch summarization of document sets. MDS approaches take as input a set of (clean) documents about a topic to be summarized and generates a fixed length summary, normally by extracting sentences from those documents [16]. The MDS task was originally proposed as an evaluation task at the Document Understanding Conference (DUC)[4]. The evaluation of MDS approaches was later continued at the Text Analysis Conference (TAC) [7]. MDS performance is usually evaluated using the ROUGE [12] suite of metrics, and we will compare to multiple popular variants [13]. While MDS experimentation can evaluate batch summarization, it is less suited to realtime summarization. First, realtime summarization considers a stream of documents instead of a batch; thus the single gold standard summary looks more like a timeline than a paragraph. Second, MDS considers a fixed length summary, whereas realtime summarization cannot know *a priori* anything about the length of the best summary. These first two points mean that ROUGE is inappropriate for realtime summarization. Third, MDS assumes that the input documents are relevant whereas realtime summarization considers a stream of arbitrarily relevant content.

Our evaluation methodology is related to work studying *retrospective MDS*, where systems process the entire stream–or batch extracts from the stream–at once, instead of incrementally. The work of Allan *et al.* [2] studies extracting sentences from a large batch of news documents. Like our methodology, this work uses an idealized summary for evaluation. However, this prior work assumes complete judgments of all updates in the stream, something that is impossible in our scenario. Tweet timeline generation considers retrospective summarization of a stream of tweets for a given topic [18]. This methodology is driven more by a semi-automatic clustering of tweets and, as a result, is subject implicit biases in the corpus, which we attempt to mitigate using a canonical record such as a Wikipedia page.

As mentioned earlier, information filtering [14] can be considered a document level version of our task. The evaluation methodology used in that work, however, considers a simple notion of relevance, not capturing subtopic/factual structure or novelty. Although some work has studied novelty in the context of filtering [20], it does not have an explicit representation of subtopics. As such, evaluation criterion cannot estimate notions like subtopic recall, which is captured in out methodology by the nuggets contained within a document/stream. Topic Tracking [1] focuses on the same document level decision with single-level relevance.

3. EVALUATION METHODOLOGY

In this section, we describe our proposed methodology for evaluating realtime summarization systems. In particular, our methodology has three parts: Stream Simulation, Information Nuggets, and Performance Metrics.

[4]http://www-nlpir.nist.gov/projects/duc/

3.1 Stream Simulation

Given an archive of timestamped documents, we can simulate a realtime summarization environment by providing documents to the system in temporal order. Given a query at time t, a system then emits zero or more sentences as updates as it receives each document after time t, the set of which we call a summary. Simulation provides a controlled, reproducible environment in which to assess how realtime summarization systems behave. Our evaluation metrics use this stream to grade the extent to which a summary's updates are relevant, comprehensive, and novel to the query.

3.2 Information Nuggets

The second part of our evaluation methodology requires the construction of a gold standard for each event, against which we can evaluate system-produced summaries. In MDS, the gold standard is represented by one or more textual summaries written by human assessors about each event. A system-generated summary can then be evaluated in terms of how similar it is to these gold-standard summaries, where the assumption is that good system summaries will be similar to the gold standard.

The gold standard in realtime summarization is a manually constructed timeline, consisting of a set of timestamped subtopics or *information nuggets* representing the unique pieces of information that a user following the event would like to know. For example, for the query 'hurricane sandy,' may include nuggets such as 'Sandy made direct hit on Jamaica,' '41 killed in the Caribbean and one in Bahamas,' and other nuggets related to landfall in New York and its aftermath. We will discuss one method for creating the set of nuggets in Section 4.

Given a query, we evaluate a summary by manually matching its updates with these information nuggets. While the nugget curation process scales with the size of the event, the matching process requires km manual comparisons for k updates and m nuggets. We will revisit this issue in Section 6.

3.3 Performance Metrics

Our design of metrics covers different aspects of evaluation: precision, comprehensiveness (recall), novelty, brevity and latency. We present results testing the appropriateness and robustness of the first four metrics. Factors like latency are very important to realtime summarization evaluation but estimating parameters such as latency penalties raises issues beyond the scope of this paper.

We believe that during and immediately following events, the most broadly useful summaries follow push notification alerts, although deciding exactly what is critical new information is a difficult task. As such, while we proposed a set of metrics for the task to cover all aspects independently, the cumulative metric targets this model.

Our precision metric, referred to as *gain*, is the sum of the relevance of each matching nugget. For a summarization system producing an update stream $\mathcal{S}$, gain is computed as:

$$\mathbf{G}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{u \in \mathcal{S}} \sum_{n \in \mathbf{M}(u)} \mathbf{g}(u, n) \quad (1)$$

where $\mathbf{M}(u)$ is the set of nuggets matching u and $\mathbf{g}(u, n)$ measures the utility of matching u with n. We adopt a gain-based metric that can integrate costs such as latency.

year	runs	events	nuggets/event
2013	28	10	119.667
2014	24	15	92.9333
2015	45	21	47.4286

Table 1: TREC Temporal Summarization track statistics.

Our comprehensiveness metric, referred to as *comprehensiveness*, is the proportion of all nuggets matched by the system updates,

$$\mathbf{C}(\mathcal{S}) = \frac{1}{|\mathcal{N}|} \sum_{u \in \mathcal{S}} \sum_{n \in \mathbf{M}(u)} \mathbf{g}(u, n) \quad (2)$$

where $\mathcal{N}$ is our set of nuggets. Like the F-measure, our combined metric, $\mathcal{H}$, is the harmonic mean of $\mathbf{G}$ and $\mathbf{C}$,

$$\mathcal{H}(\mathcal{S}) = 2 * \frac{\mathbf{C}(\mathcal{S}) * \mathbf{G}(\mathcal{S})}{\mathbf{C}(\mathcal{S}) + \mathbf{G}(\mathcal{S})} \quad (3)$$

In order to reward novelty, we only allow each nugget to be matched by at most one update in the system's summary. We ignore later matches to updates in the summary for computing Equations 1 and 2, considering redundant content nonrelevant. We do not account explicitly for brevity in our metrics; we enforce it in our experimentation by constraining updates to be selected from segmented sentences from the stream documents.

4. DATA

We collected data from three years of the TREC Temporal Summarization track.[5] Each year's data consists of three components. The first component includes the evaluation events from 2012-2014 with Wikipedia entries providing fine-grained documentation of the event progression [11] and having substantial representation in the target corpus, the KBA Stream Corpus.[6] Assessors at NIST used the revision history of these pages to construct each topic's nugget set (Section 3.2), forming the ideal summaries. The second component of the data consists of participant runs. For three years, track participants designed summarization systems and generated summaries following the protocol described in Section 3.1. Participants provided confidence values for individual updates which allows for selective evaluation and analysis. We present summary statistics for this data in Table 1. The final component of our data consists of a manual matching between the nuggets in the ideal summary and the systems' output. Because the union of all submitted runs still resulted in an enormous set of updates, the set of updates inspected and matched by assessors was sampled from this union. Specifically, each run was guaranteed to have its top 60 updates (as determined by the system's confidence values) judged by assessors. The system updates were then pooled for matching. NIST assessors matched these pooled updates to the previously created nuggets. We conservatively expanded the set of matches using near duplicate detection [4].

5. USER EVALUATION

While Section 3 presented the track metrics and their desired properties, here we demonstrate that these metrics reflect what users consider to be a high quality event summary.

[5]http://www.trec-ts.org/
[6]http://trec-kba.org/

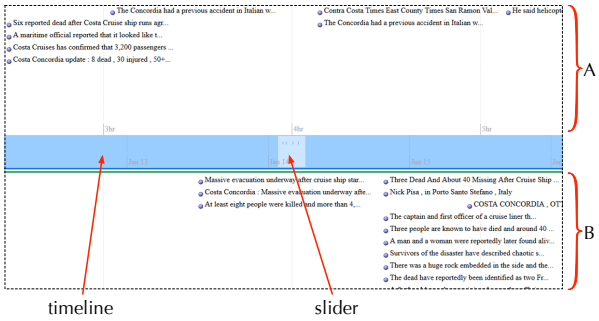


Figure 1: Pairwise visualization of event summary extracts.

To this end, we conducted a controlled experiment measuring the correlation of system ordering by human judges and our metrics for the 2014 track data. Our approach elicits pairwise judgments from assessors, as has previously been done for retrieval evaluation [17]. Under this approach, assessors were shown pairs of summaries (or rather summary extracts–as discussed further below), and were asked to state a preference for one summary or the other.

5.1 Experimental Setup

We sampled 1,000 summary pairs of systems such that each system appeared in the same number of pairs for each event. Duplicate summary pairs were omitted. Because full system summaries could include hundreds or thousands of updates, they were truncated to ease the cognitive load during assessment. Specifically, we truncated summaries so that they included the top thirty updates as sorted by the system's confidence value.

An assessor was presented with instructions describing the task followed by a short overview of the current event. The assessor then opened a webpage that visualized the two timelines and was instructed to scroll the timeline until they had read all 30 updates in each summary (Figure 1). The placement of each timeline in the 'A' vs 'B' slot was randomly selected. Afterward, the assessor was asked to state which of the two summaries they preferred or that they had no preference. Furthermore, we asked assessors to select one or more explanations for the decision.These explanations included: topicality, coverage, quality, redundancy, and timeliness.

We recruited assessors through the CrowdFlower[7] crowdsourcing marketplace, which aggregates multiple existing marketplaces. Assessors were provided with a single page containing six summary pairs for assessment, as prior research has indicated that larger batch sizes can result in better quality work [15]. In order to address unreliability in judgments, we employed the following quality assurance techniques. First, following best practices [3], three individual crowd workers provided preferences for each pair of summaries. The final assessment produced was the majority vote across the three assessors. Summary pairs for which no majority agreement was reached were dropped from the evaluation. Second, work submitted was subject to a speed trap of 60 seconds per page. This detected automatic bots and assessors that were simply randomly selecting labels. Assessors that submitted a page in under 60 seconds were flagged and removed from the evaluation. Third, to avoid over-reliance on individual assessors, the maximum number of assessments that any one assessor could contribute was set to 100. Fourth, as summaries were in English, we restricted the geographic regions that could participate in the

[7]http://www.crowdflower.com/

labeling task to only those from the United States, Canada and the United Kingdom. Finally, upon attempting the job, workers were subject to a 1 page test (summary 6 pairs), for which their accuracy was compared against a manually created gold standard (comprised of 32 examples). Workers were required to answer 4 or more of these test questions correctly to continue. We paid US $0.15 for each set of 6 summary pairs assessed. In total, 3,000 assessments were collected (1,000 summary pairs * 3 unique assessors).

5.2 Results

One hundred unique workers attempted the task, of which 77 passed the 1 page test. Approximately 40% of the crowd workers completed the maximum number of judgments (96 pairs/16 pages of work), followed by a 'tail' of workers who completed fewer assessments, which is typical behavior in crowdsourced tasks [3]. The average time it took the workers to assess a pair of summaries was 25 seconds, or 2 minutes 32 seconds per page. Out of these impressions, assessors made a preference judgment (stated a preference for either summary A or summary B) 67.1% of the time and preferred neither in the remaining 32.9% of cases. Inter-worker agreement over the preferences was 81.25%, which indicates that the labels were of high quality and the task was of appropriate difficulty. The most frequently selected reason for the preference was 'coverage' (40.7%) followed by, 'quality' (19.4%), 'topical' (15.4%), timeliness (10.5%), and redundancy (10.2%), with 3.7% of preference judgments providing no reason.

We counted the number of times a system was preferred in pairwise impressions. We refer to this as the system's win rate. In Figure 2, we compare each system's win rate with the metrics discussed in Section 3. The results in Figure 2 show a significant correlation between our comprehensiveness metric and assessor preference. This is understandable, as an assessor is likely to be able to judge some amount of coverage when our pool is limited to the top 30 updates per team. Although we *were* somewhat surprised that assessors were able to estimate a recall-oriented metric better than the precision-oriented metric, this phenomenon is consistent with previous literature [8]. Gain shows a weaker, but still reasonable correlation, and our combined metric shows significant correlation, supporting its use for our evaluation as an effective single measure for system performance.

Although we observed good response rates of assessors providing reasons for preferences, we found that only 'redundancy' was correlated with our metrics. We found a statistically significant correlation with comprehensiveness ($\tau : 0.42, \rho : 0.56, r = 0.62$ with $p < 0.05$ for all measures). Assessors claimed that coverage was important, but we did not find that they were actually able to detect differences. It seems that assessors appear averse to redundant content, whose presence can degrade comprehensiveness.

Additionally, we compare our results to a baseline using the ROUGE evaluation framework. As we do not have traditional summaries, we concatenate all nuggets into a summary to serve as the gold standard, and concatenate all updates per system into a single summary to serve as the participant summary. While this is may not be optimal, to our knowledge there do not exist other established evaluation frameworks for such types of summaries; indeed one of the purposes of the TREC Track was to establish such a method. As shown in Table 2, correlation with track metrics was lower for the ROUGE than it was for the win-rate described earlier. This was true for all popular ROUGE variants tested, including ROUGE 2, 4 and SU-4 P, R, and F metrics, except ROUGE-4P having reasonable kendall correlation, but poorer pearson and spearman.

Metric	Kendall	Pearson	Spearman
ROUGE-2P	0.30	**0.54**	0.43
ROUGE-2R	-0.11	0.36	-0.13
ROUGE-2F	0.13	**0.58**	0.21
ROUGE-4P	**0.35**	**0.48**	0.43
ROUGE-4R	0.13	0.18	0.16
ROUGE-4F	0.22	0.42	0.33
ROUGE-SU4P	0.05	0.02	0.12
ROUGE-SU4R	-0.15	0.41	-0.16
ROUGE-SU4F	0.18	**0.64**	0.29

Table 2: Correlation values for ROUGE metrics vs user study winRate using kendall, pearson, and spearman coefficients.

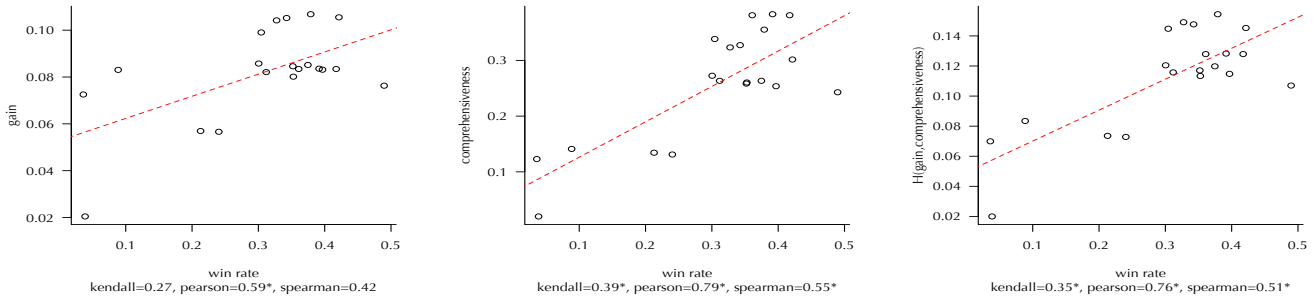
5.3 Pair-preference case analysis

To better understand the disagreement between study assessors and the primary track $\mathcal{H}$ metric, we identified 80 cases with strong disagreement out of the 946 pairs assessed and manually analyzed them. We observed that in all of these cases, a recall-orientated system (that produces a verbose summary containing thousands of updates) was being compared to a precision-oriented summary (that typically produced 40-100 updates). In these cases, the human assessors preferred the summaries produced by the recall-orientated systems, while the metric $\mathcal{H}$(gain, comprehensiveness) prefers the precision-orientated one. This can be explained by the fact that the user study truncates summaries to the 30 with the highest score, and hence the user will not penalize these systems for returning excessive content (since it is not shown). In contrast, the $\mathcal{H}$ metric contains a gain component that will penalize large numbers of redundant or off-topic updates.

Furthermore, we observed that users prefer timelines which concentrated the updates in the first few hours of an event, the initial view of the assessment interface. For this reason, precision-orientated summarization systems that delayed returning content until later in the event (typically systems that returned updates on end-of-hour boundaries) were penalized by the user study. This is a source of disagreement between the user study and the track $\mathcal{H}$ metric, since the latency decay used within the $\mathcal{H}$ metric was quite forgiving, in that it allowed for multiple hours to pass before late reporting of information was penalized significantly. Our metrics, without redesign, can made far more sensitive of latency, if so desired. In general, from this study, we can draw two main conclusions. First, verbose summaries are difficult to evaluate with users since they need to be truncated to display, which is a source of evaluation error. Second, users of this timeline interface tend to prefer to see updates near the beginning of the event, indicating that for evaluating timeline summaries, the latency discount function used should more strongly penalize systems that return information late.

6. REUSABILITY

Our user study demonstrates that the method described in Section 3 provides a good metric for systems participating in the manual nugget matching process. However, in order for our simulation to be of value for future research, it must be reusable after the initial experiments and evaluation have been completed [5]. In particular, the data and associated metrics should ideally be able to accurately determine the



(a) gain (b) comprehensiveness (c) H(gain,comprehensiveness)

Figure 2: Performance for all participating systems scored by the three primary track metrics vs the controlled experiment win rate for the 2014 track. Starred correlations have $p < 0.05$.

performance of novel systems compared to systems which participated in the original manual assessment without requiring additional manual effort.

In this section, we explore the robustness of our metrics in the presence of new runs with potentially missing judgments. We quantify the robustness of the metrics when entire sets of runs are suppressed from contributing to the pool. This simulates the behavior of a new system being developed using preexisting data.

	Holdout τ (τ_{AP})	
Year	**System**	**Team**
2013	0.97 (0.85)	0.95 (0.82)
2014	0.97 (0.87)	0.82 (0.75)
2015	0.95 (0.89)	0.90 (0.93)

Table 3: Kendall's τ (τ_{AP}) values comparing the track scores vs scores holding out individual/all systems for each team.

6.1 Evaluation with Missing Judgments

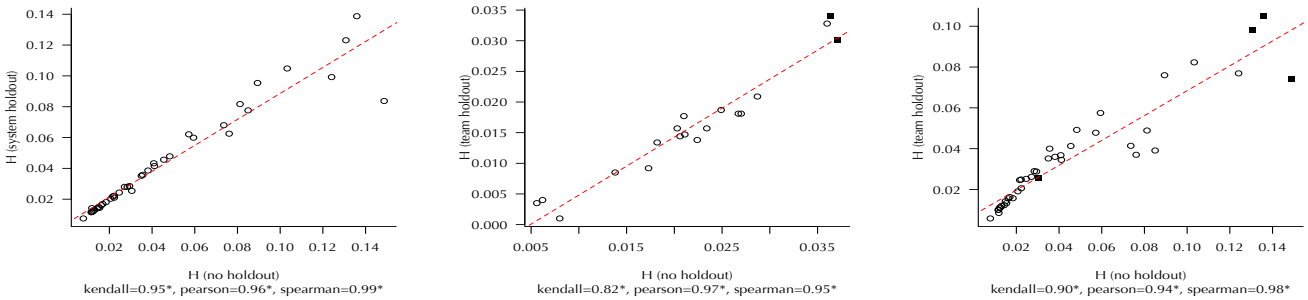
To analyze reusability, we perform two procedures which mimic the addition of new systems. In the first, we remove from the pool all updates submitted by a system which were not submitted by any other system. We perform the standard evaluation procedure for that system, report its score using the primary track metric $\mathcal{H}$, and repeat for each submitted system. This method simulates a new system to be evaluated without ability to collect and match additional updates. Therefore, it provides a good representation of whether or not we can accurately determine the relative performance of this new system compared to the systems submitted to the track. It is actually more difficult than simply having future systems, as we artificially shrink the test collection size to perform this analysis.

Next, we repeat the same procedure, instead removing all unique updates for all systems submitted by the same team rather than for a single system. This corrects for systems which are simply small variations of a single system submitted by the same group, a common practice in these track settings and analogous to having a new team submit systems after the track evaluation is complete. Again we compare the system rankings to the original ranks.

As a standard method of comparing the similarity of two ranked lists, we compute the Kendall's τ values to compare the original system rankings to each of the two holdout methodologies. Kendall's τ measures the proportion of pairs of items which are ranked identically across two different rankings. However, τ gives equal weight in the metric to pairs at the bottom of the list as to pairs at the top. But when evaluating system performance, it is usually more important if one high-performing system outperforms another than is it for two low-performing systems. As such, we also compute τ_{AP} values, as this method better describes the correlation between the top ranks in the list, or top performing systems in our setting [19]. This is important, as the goal of a new system is to outperform existing systems, and the evaluation methodology should be able to accurately assess this case.

The Kendall's τ and τ_{AP} results for our holdout experiments are reported in Table 3 for all tracks. We also plot the results to more fully analyze the concordance of the methods, as both τ and τ_{AP} can hide subtle, yet important swaps. The holdout results data are shown in Figure 3, holding out individual systems on TS15 track (left), and holding out teams (middle TS14, right TS15). Overall, there is high concordance between the original results and the holdout results, suggesting the utility of the dataset for additional experiments. Concordance at top ranks was much higher for the 2013 and 2014 tracks, with the exception of 2014 team holdouts, which is why we show the remaining three plots here. In the 2015 track, we note two or three high-performing systems which stand as outliers to the general consistency, and cause the generally lower τ_{AP} numbers, although they remain in an acceptable range. This is often found in TREC datasets, where one or two teams submit systems trying to retrieve particularly unique results. In fact it was true in this case as well, as the top 2015 team outperformed others primarily due to high recall and having a high proportion of unique updates than the other submissions.

Looking more closely at the systems and scores for the 2014 team holdouts, we see that the Comprehensiveness is fairly high for all systems (Figure 2), but the Gain is low, and in particular, this means that a small proportion of submitted updates were relevant. This is primarily due to the large number of updates provided by many teams. As such, if even a small number of those previously relevant updates are found not relevant, the scores can fluctuate wildly. In the 2015 track, teams had better estimated the number of updates needed, so the scores are better overall and are less sensitive to small variations in the dataset. The 2013 track had more nuggets per event than the other two, also improving its resilience to lost matches in the holdout process.



(a) system holdout (2015) (b) team holdout (2014) (c) team holdout (2015)

Figure 3: Reusability analysis comparing the scores for each system as reported by the track versus removing from the pool unique updates submitted by that system for the left plot for the 2015 track, and for that entire team for middle and right plots for the 2014 and 2015 tracks respectively.

7. DISCUSSION

The results of our controlled experiment demonstrated a correlation between our metrics and human judges. We were surprised by the lack of strong correlation with gain, since differences in precision would be more recognizable given our high precision presentation (only the top 30 updates were displayed). This may result from assessors' difficulty in distinguishing between systems with very minute differences in gain. At the same time, our experiment could be improved by conducting an evaluation of system performance in real-time. Indeed, this is one of the motivations for the TREC 2016 Realtime Summarization track.

Although the datasets differed in terms of number of participating teams and systems submitted, number of topics, size of those topics, and number of nuggets and matches found, the evaluation methodology resulted in reusable data with strong correlations to the original sets despite the challenging task of holding out a significant portion of the systems involved in the pooling. This robustness strongly suggests that the data and metrics are correlated to users across track years, and that the same methodology could be used in future stream summarization tasks that want to simultaneously optimize for the diverse criteria here, such as microblog summarization, social media news feeds, news outlet information aggregation, and crisis responder information feeds.

Additionally, while our simulation-based evaluation method measures the effectiveness of systems ignoring efficiency concerns, much like the Cranfield approach, this system could be easily adapted to rate efficiency as well by running in real-time and processing updates when they are provided to the evaluation system. This could allow for on-line training of these systems in conjunction with crowd worker annotations. Again, these ideas are being explored by the TREC 2016 Realtime Summarization track.

In closing, our studies provide strong evidence supporting the methodology used in the TREC Temporal Summarization track. While this track has run for three years, our experiments are the first to demonstrate the validity and utility of the methodology. The results provide evidence that the collection is reusable for future research and we provide a novel algorithm able to improve the robustness when judgments are incomplete. Our results, taken together, provide strong support for using this methodology for realtime summarization tasks in the future.

8. REFERENCES

[1] J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Inf. Retrieval. 2002.

[2] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of new topics. In *Proc of SIGIR*, 2001.

[3] O. Alonso and R. Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *Proc. of ECIR*, 2011.

[4] G. Baruah, A. Roegiest, and M. D. Smucker. The effect of expanding relevance judgements with duplicates. In *Proc. of SIGIR*, 2014.

[5] B. Carterette, E. Gabrilovich, V. Josifovski, and D. Metzler. Measuring the reusability of test collections. In *Proc. of WSDM*, 2010.

[6] Y. Chen, N. J. Conroy, and V. L. Rubin. News in an online world: the need for an automatic crap detector. In *Proc. of ASIS&T*, 2015.

[7] H. T. Dang and K. Owczarzak. Overview of the TAC 2008 update summarization task. In *Proc. of TAC*, 2008.

[8] M. Dostert and D. Kelly. Users' stopping behaviors and estimates of recall. In *Proc. of SIGIR*, 2009.

[9] Q. Guo, F. Diaz, and E. Yom-Tov. Updating users about time critical events. In *Inf. Retrieval*. Springer, 2013.

[10] M. Imran, C. Castillo, F. Diaz, and S. Vieweg. Processing social media messages in mass emergency: A survey. *ACM Comput. Surv.*, July 2015.

[11] B. Keegan, D. Gergle, and N. Contractor. Hot off the wiki: Structures and dynamics of wikipedia's coverage of breaking news events. *American Behavioral Scientist*, 2013.

[12] C.-Y. Lin. ROUGE: a Package for Automatic Evaluation of Summaries. In *Proc. of ACL*, 2004.

[13] C.-Y. Lin and F. Och. Looking for a few good metrics: Rouge and its evaluation. In *NTCIR Workshop*, 2004.

[14] H. P. Luhn. A business intelligence system. *IBM J. Res. Dev.*, 1958.

[15] R. Mccreadie, C. Macdonald, and I. Ounis. Identifying top news using crowdsourcing. *Inf. Retrieval*, 2013.

[16] A. Nenkova and K. McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 2011.

[17] P. Thomas and D. Hawking. Evaluation by comparing result sets in context. In *Proc. of CIKM*, 2006.

[18] Y. Wang, G. Sherman, J. Lin, and M. Efron. Assessor differences and user preferences in tweet timeline generation. In *Proc. of SIGIR*, 2015.

[19] E. Yilmaz, J. A. Aslam, and S. Robertson. A new rank correlation coefficient for information retrieval. In *Proc. of SIGIR*, 2008.

[20] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proc. of SIGIR*, 2002.