A Comparison of Nuggets and Clusters for Evaluating Timeline Summaries

Gaurav Baruah,[1] Richard McCreadie,[2] and Jimmy Lin[1]
[1] David R. Cheriton School of Computer Science, University of Waterloo, Ontario, Canada
[2] School of Computing Science, University of Glasgow, Scotland, the United Kingdom
{gbaruah,jimmylin}@uwaterloo.ca,richard.mccreadie@glasgow.ac.uk

ABSTRACT

There is growing interest in systems that generate timeline summaries by filtering high-volume streams of documents to retain only those that are relevant to a particular event or topic. Continued advances in algorithms and techniques for this task depend on standardized and reproducible evaluation methodologies for comparing systems. However, timeline summary evaluation is still in its infancy, with competing methodologies currently being explored in international evaluation forums such as TREC. One area of active exploration is how to explicitly represent the units of information that should appear in a "good" summary. Currently, there are two main approaches, one based on identifying nuggets in an external "ground truth", and the other based on clustering system outputs. In this paper, by building test collections that have both nugget and cluster annotations, we are able to compare these two approaches. Specifically, we address questions related to evaluation effort, differences in the final evaluation products, and correlations between scores and rankings generated by both approaches. We summarize advantages and disadvantages of nuggets and clusters to offer recommendations for future system evaluations.

1 INTRODUCTION

In many information-seeking scenarios, users desire results that are relevant, diverse (i.e., cover many facets of the users' needs), non-redundant (i.e., contain no repeated information), and timely (i.e., contain up-to-date information). These characteristics are especially desirable for systems that generate timeline summaries for events or topics from document streams, where the input contains a large amount of non-relevant or redundant information [10, 15, 20]. The output of timeline summarization systems are frequently referred to as *updates* since they are usually generated incrementally as the system processes a stream of documents.

Current evaluation methodologies for timeline summaries are built on identifying atomic units of information, which serve as the basis for calculating the relevance, diversity, redundancy, and timeliness of the summary. To date, there exist two competing methodologies to generate these atomic information units:

The first is the *nugget-based* methodology, which was originally developed for question answering [23], but was recently adapted to evaluate temporal summarization systems at TREC [3]. Nuggets represent abstract "atoms" of information that may manifest in different updates (sentences from newswire articles) returned by systems. For example, the updates "damages total around $4 billion" and "left a path of destruction estimated to cost around four billion" might be said to both contain the nugget "$4 billion damages". Nuggets form a many-to-many relationship with system updates. In temporal summarization, nuggets are defined by human assessors based on analysis of Wikipedia pages about those events [3].

The second is the *cluster-based* methodology, which was developed for the TREC 2014 Microblog Track [13]. Here, tweets comprise system updates, which are assumed to be atomic by fiat, and human annotators manually cluster these updates into semantic equivalence classes. As with nuggets, updates in the same cluster can express the same concept using different words. However, by design each update can only belong to one cluster. With this methodology, no external information is used.

The nugget-based and cluster-based evaluation methodologies represent different tradeoffs in evaluation design. Nuggets are more fine grained and can incorporate external knowledge, which allows for more nuanced evaluation. However, the downside is that nugget annotations are costly. Cluster-based evaluations make the opposite tradeoff: the annotation process is much more lightweight, but at the expense of conflating different facets that may be present in the returned results.

To date, evaluations have either adopted one methodology or the another, which leads to the obvious question: How do the nugget-based and cluster-based evaluation methodologies compare? The best way to answer this question is to analyze one or more test collections that have been evaluated with *both* methodologies. Since no such test collection exists, we organized an effort that applied the cluster-based evaluation methodology to data from the TREC 2013 and 2014 Temporal Summarization test collections, which had already been "nuggetized".

Contributions. The main contribution of our work is a comparative analysis of nugget- and cluster-based evaluations over the same set of topics and systems, which supports a fair comparison of many aspects of the evaluation. Our work integrates two evaluation approaches that have until now been disparate and incomparable. Specifically, we tackle three main questions:

(1) How do the two evaluation methodologies compare in terms of effort?
(2) Can we characterize quantitative and qualitative differences between nuggets and clusters?

CIKM'17, November 6–10, 2017, Singapore.

ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: http://dx.doi.org/10.1145/3132847.3133000

(3) Do system scores and rankings generated using clusters correlate with those generated using nuggets?

Our results show that there is substantial agreement between the two approaches in terms of both scores and system rankings, and hence both are reasonable options for evaluating timeline summaries. However, neither approach is a one-size-fits-all solution. In particular, nugget-based evaluations provide more fine-grained estimates of the information content of updates and support richer failure analyses. However, this comes at a steep cost in terms of evaluation time and money: based on our case study, we find that nugget annotations take three to five times more effort than cluster annotations. Furthermore, for some topics nuggets may underestimate recall since systems are often not rewarded for returning less important information. On the other hand, cluster-based evaluations take far less effort and better capture overall recall, but produce labels that are less effective at distinguishing important from trivial or non-salient information within a summary.

2 BACKGROUND AND RELATED WORK

The idea that information retrieval systems should return fine-grained units of information is of course not new. These have been called aspects, facets, sub-topics, nuggets, or clusters in the literature, and the thread of work exploring retrieval of these units dates back at least two decades [17, 26].

A variety of summarization evaluation methodologies have previously been proposed in the literature. Early work in summarization evaluation focused on estimating the quality of fixed-length textual summaries, such as the output of multi-document summarization (MDS) systems [16]. In addition to manual assessments, these evaluations were based on comparing the system summary to one or more gold standard summaries produced by humans. In this case, a "good" system summary is one that is textually similar to the gold standard summaries. The ROUGE [11] suite of metrics have been used to measure n-gram overlap between system-generated summaries and gold-standard summaries for many years as part of the Document Understanding Conferences (DUCs) [7] and Text Analysis Conferences (TACs) [8].

However, due to the widespread adoption of social media platforms for sharing information in real-time, as well as push-based systems developed by online news outlets to provide real-time content, temporal or timeline summaries have become popular. A timeline summary is comprised of a series of (approximately) sentence-length timestamped updates relevant to an event or topic. These updates are typically shown to the user as a single aggregate summary (timeline), which is augmented with new updates over time. Twitter Moments is one current example of a deployed timeline summarization system.[1] A variety of automatic timeline generation approaches have been recently proposed, which take as input a stream of text items (e.g., sentences or tweets) and select a subset of them to emit into a timeline summary [15, 20, 25, 27].

From an evaluation perspective, there are two notable differences between timeline summaries and the classic MDS-style retrospective summaries that came before. First, timeline summaries are variable-length and evolve over time. This makes comparative evaluation against a static gold-standard summary problematic, since current tools like ROUGE [11] and its temporal extensions [6, 10] assume that system outputs and gold-standard summaries are of (roughly) equal length, and that the gold-standard summaries do not change over time. Second, the value of a timeline summary is in part based on how up-to-date the information content is, which is not captured by ROUGE-like metrics. Hence, a new evaluation methodology for timeline summaries was needed.

The answer to these evaluation challenges is to explicitly represent the "information atoms" that a "good" summary should contain. Specifically, the nugget-based and cluster-based evaluation methodologies offer competing realizations of this basic idea. We are not aware of any work that has examined the differences between them. In this paper, we perform exactly such a comparison, with the aim of determining the advantages and limitations of each, as well as producing recommendations for when each methodology should be used in future evaluations. To begin, we describe both approaches in more detail.

2.1 Nugget-Based Evaluations

Although the nugget-based evaluation methodology was first developed for question answering [23] and applied in a number of large-scale evaluations in the 2000s, here we describe its most recent deployment in the TREC Temporal Summarization Tracks [3], which took place from 2013 to 2015. Each evaluation comprised a set of events (akin to topics in *ad hoc* retrieval) such as the 2012 Pakistan garment factory fires or the 2012 Buenos Aires train crash. Participants were given a collection of newswire articles, which they processed in time order to simulate a live document stream. The system's task was to incrementally select relevant, non-redundant sentences from the incoming documents to return as the summary updates. To account for differences in the definition of a "sentence", each document in the collection was pre-segmented.

In this context, nuggets represent atomic facts relevant to the events, expressed as short natural language phrases. The intuition is that a perfect timeline should include all of the information represented by the nuggets and information contained in each nugget should only be reported once (i.e., system output should avoid redundant content). Furthermore, each nugget can be considered to have a lifetime, where the nugget is "born" at the time that the information "becomes known" (e.g., via publication in a news article) and "dies" at a later point in time when the information contained within the nugget is no longer useful or becomes outdated. A good timeline summary should include a nugget as soon as possible after it becomes known.

It bears emphasizing that although nuggets are identified by short natural language phrases, they represent concepts independent of the surface lexical realization. For example, the nugget "the train crashed at the buffer stop" might manifest in system updates as "slammed into the end of the line", "smash into a barrier", or "hit the barrier at the end of the platform".

The nugget-based evaluation methodology includes two major phases: nugget extraction (colloquially called "nuggetization") and nugget matching. In the first phase, the ground truth nuggets for a particular event are created. In the second phase, instances of the nuggets in system updates are identified. In the TREC Temporal Summarization Tracks, both phases involved human effort (by NIST

[1] https://twitter.com/i/moments

assessors). From the record of which nuggets were found in which system updates, various metrics to quantify output quality can be straightforwardly computed. Here, we provide a brief overview, but we refer the reader to the TREC Temporal Summarization Track guidelines and overview papers [1, 2]:

Nugget Extraction. Building on earlier work [10], the Temporal Summarization Tracks took advantage of Wikipedia articles as high-quality sources of information nuggets. Using a custom annotation interface, assessors manually analyzed the history of the Wikipedia page corresponding to an event (i.e., the stream of edits made by human contributors) for the duration of that event's lifetime. The assessors defined new information nuggets as they encountered novel information about the event that they considered important enough to be included in a "good" summary. Each of the information nuggets was assigned a grade based on its importance in the assessor's opinion.

Nugget Matching. Given the ground truth nuggets as an "answer key", assessors then identified instances of individual nuggets contained in each system's updates. In practice, output from participating systems was first pooled, on which near-duplication was then applied, and only unique updates were annotated with nuggets. Each update can contain zero, one, or more nuggets. Note that since nuggets represent concepts, the nugget matching process requires understanding the semantics of each update—taking into account linguistic phenomena such as synonyms, paraphrasing, etc.

2.2 Cluster-Based Evaluations

The biggest source of complexity in nugget-based evaluations is the many-to-many mapping between nuggets and system updates. The methodology assumes that each unit of system output (a sentence in the case of temporal summarization) may contain multiple atomic information units, and hence the need to enumerate something finer-grained. In contrast, the starting point of the cluster-based methodology is a declaration, by fiat, that the unit of system output *is* the atomic unit of information. Based on this assumption, it now suffices to group system updates into semantic clusters (i.e., semantic equivalence classes). The ideal summary should include one and only one member of each cluster. Given a particular clustering of system outputs (e.g., from pooling), it is straightforward to compute various metrics of output quality, and the temporal ordering of updates assigned to the same cluster can be used to measure the timeliness of an update.

The cluster-based evaluation methodology was first developed for the Tweet Timeline Generation (TTG) Task in the TREC 2014 Microblog Track [13, 24] and used in subsequent TREC evaluations involving tweets. Given their usage in practice to communicate concisely and strict character limits, it seemed reasonable to consider tweets themselves the atomic units of information. The 2014 TREC evaluation involved retrospective information seeking, while subsequent evaluations explored prospective information seeking [14], but the core task is essentially the same: the system's task is to retrieve tweets that are relevant with respect to a statement of information need.

The TTG protocol proceeded in two steps: First, system outputs (tweets) were pooled and assessed for relevance. Second, the relevant tweets were then clustered. This was accomplished via a custom annotation interface that presented one tweet at a time in chronological order (earliest first). For each tweet, the assessor could either add it to an existing cluster if she believed the tweet to be substantively similar to those tweets. Otherwise, the assessor could create a new cluster. By design, each tweet can only be assigned to a single cluster. The interpretation of "substantively similar" is left to the assessor—no doubt there will be cases of partial or ambiguous matches. In these cases, it is left up to the discretion of the human assessor which of the available clusters is the most relevant, or if the creation of a new cluster is warranted.

The final output of the clustering process is a set of tweet clusters, where each cluster is comprised of a list of tweets. All tweets in the same cluster are assumed to convey the same information. Note that despite many potentially worrisome simplifications made in the cluster-based evaluation methodology, Wang et al. [24] concluded that metrics based on these clusters correlate with human preferences. They also showed that system rankings are stable with respect to different clusters by different assessors. Note that this meta-evaluation considered only the application of clustering to tweets; surely, the validity of the assumptions would depend on the type of information unit being clustered.

3 BUILDING TEST COLLECTIONS

This work aims to compare the nugget- and cluster-based evaluation methodologies. If there existed test collections that contained both nugget and cluster annotations, this task would be easy, but unfortunately no such test collection exists, since all previous evaluations adopted one or the other approach. Thus, we had to manually build such doubly-annotated test collections ourselves.

There were three possible approaches: We could start with a recent nugget test collection, say, data from the TREC Temporal Summarization Tracks, and apply the cluster-based methodology to re-annotate the data. Alternatively, we could start with a recent cluster test collection, say, from the TREC Microblog Tracks, and apply the nugget-based methodology. Finally, we could start from scratch and build a test collection with both nuggets and clusters. We adopted the first approach for three reasons: First, it made sense to build on existing test collections to leverage previous investments in evaluation resources and to offer comparisons with previous results. Second, the cluster-based approach seemed simpler and therefore would likely involve less effort. Finally, recall that the cluster-based approach assumes that each update is atomic. The longer the update, the more questionable this assumption becomes: for tweets, this claim might be believable, but it would be highly doubtful for, say, paragraph or documents. Thus, it would be more interesting to apply clustering on longer updates to test the limits of this assumption. Since updates in Temporal Summarization are sentences from newswire articles (and on average longer than tweets), it made sense to attempt clustering on sentences.

We focused on 24 events from the TREC 2013 and 2014 Temporal Summarization Tracks.[2] The pool of 23,764 updates for these events comprised the input to the clustering workflow. As described in Section 2.2, the original formulation of TTG was divided into two steps: the assessors first removed non-relevant updates, and then

[2]Note that these topics are numbered from 1 to 25, however topic 7 from the 2013 edition of the track was dropped from evaluation by the track organizers due to lack of relevant updates in the pool, and hence there are 24 topics total.

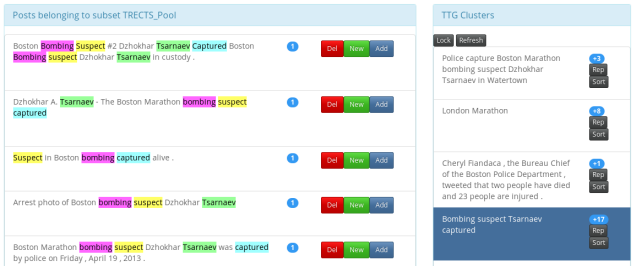

Figure 1: Our semantic clustering interface showing the current clusters in the right-hand pane and the list of updates in the left-hand pane.

semantically clustered those that remained. One of the reasons for this was that the assessors who performed the relevance judgments differed from those who clustered the tweets (due to resource limits at the time of the evaluation). To better streamline the annotation process and to increase efficiency, we built a custom assessment interface that combined both aspects.

A screenshot of this interface is shown in Figure 1. The updates for an event are shown in the left-hand pane, initially shown in time order. The current clusters for the event are shown in the right-hand pane. For each update, the assessor has three options:

(1) She can delete the update by pressing the 'Del' button next to the update. This is equivalent to marking the update as not relevant and removing it from consideration.
(2) She can create a new semantic cluster and add the update to the new cluster by pressing the 'New' button next to the update.
(3) She can add the update to an existing semantic cluster by clicking on the cluster (which highlights that cluster in the list) and then by pressing the 'Add' button next to the update.

Additionally, within the cluster pane on the right-hand side, other controls are provided to make the assessment process more efficient: Next to each cluster, the 'Rep' button opens a dialogue box that enables the assessor to update the description of the semantic cluster. The 'Sort' button next to each cluster performs an automatic sort of the updates based on textual similarity to the cluster's description, enabling the assessor to quickly find other updates that should also be added to the cluster.

We recruited four assessors to re-annotate the Temporal Summarization datasets in the manner described above. These assessors were either students or recent graduates of a Masters program in library and information science. They were compensated at the rate of $15 USD per hour. In the final product used for subsequent analyses, of the 24 topics, we discarded two topics: TS14.24 and TS14.25. The clusters produced for these topics were not usable due to a misinterpretation of instructions by one of the assessors. Hence, for the remainder of this paper, system metrics and measures of effort reported for both the nugget and clustering methodologies are computed over the remaining 22 topics.

To support future research into temporal summarization, we have made our complete dataset containing the additional assessments publicly available to the community.[3]

[3]http://dx.doi.org/10.5525/gla.researchdata.410

4 ANALYSIS

As the result of our efforts, for 22 topics from the TREC 2013 and 2014 Temporal Summarization Tracks, pooled system updates have been processed using the nugget-based evaluation methodology (the original evaluation) and the cluster-based evaluation methodology (by assessors we recruited). With this doubly-annotated dataset, we wish to answer three research questions:

(1) How do the two evaluation methodologies compare in terms of effort?
(2) Can we characterize quantitative and qualitative differences between nuggets and clusters?
(3) Do system scores and rankings generated using clusters correlate with those generated using nuggets?

We address each of the above questions in turn.

4.1 Evaluation Effort

In total, the four assessors we recruited took 71 hours 16 minutes to cluster 21,635 updates for 22 topics, or about 5 assessments per minute. As a point of comparison, the NIST assessors reportedly spent 375 hours assessing 23,764 updates for the 24 topics in the TREC 2013 and 2014 Temporal Summarization Tracks, or about 1 assessment per minute. Note that we do not have fine-grained breakdowns by topic from NIST, so for comparison purposes, we simply interpolate the NIST figures, which translates to 341 hours for the 22 topics we are analyzing here. From a simple calculation, we find that the nugget-based methodology requires about five times more effort than the cluster-based methodology.

How do we account for the much greater effort associated with the nugget-based methodology? Although nugget extraction (from Wikipedia) represents an additional stage that is not present in the cluster-based methodology, it seems unlikely that this alone explains the large difference in time spent during assessment. There must be more hidden complexities.

To better understand the breakdown of effort, we analyzed database logs generated by the assessment interface used by the NIST assessors during the evaluation periods, which were kindly provided by the organizers of the TREC Temporal Summarization Tracks. We manually divided the log data into identifiable assessment sessions based on when new nuggets or matches were added to the database holding the evaluation product. We then total the time spent during these sessions to arrive at an estimate of the total time. We emphasize that these are estimates, compared to the figures from the cluster annotation interface, for which we have more detailed logs and hence are more accurate.

We observe that it took approximately 45.5 hours for the NIST assessors to extract the ground truth nuggets from Wikipedia for the topics in TS 2013 and 2014. The nugget matching process took another 155.5 hours. From these figures, we find that the nugget-based approach takes about three times more effort. Note that the nugget matching process alone takes about two times longer than the entire clustering process, since nuggets are much more fine grained (more details below). In particular, an increase in the number of nuggets for a topic has the double effect of increasing the amount of time it takes for nugget extraction *as well as* matching the nuggets in the system updates. Furthermore, the matching process increases substantially in complexity as the number of nuggets

to match against increases, because a system update can contain multiple nuggets. Conceptually, the assessors are populating a sparse $N \times U$ matrix where N is the number of nuggets and U is the number of updates in the pool [5]. In contrast, an update can only belong to one cluster by design.

The sum of the observed times for nugget extraction and nugget matching from the logs indicates that there are around 175 hours unaccounted for, since the NIST assessors reported spending 375 hours in total. This can be partially explained by assessor activities that are not observable from the logs, but were communicated to us from NIST. For nugget extraction, the assessors spent time reading articles about the events before assessing system updates. For nugget matching, some assessors printed out the nuggets for a topic and tried to memorize them to increase matching efficiency. Also, for the 2014 topics, from the logs, we noted a set of assessments for three topics that were batch inserted into the assessment database at the end of the evaluation period. We believe that these topics were assessed offline (i.e., not using the assessment interface), and hence the effort would not be included in the figures above (thus making those figures a lower bound).

In summary, we observe from this case study that the nugget-based evaluation methodology is substantially more costly in terms of assessment effort than the cluster-based evaluation methodology: from our analysis, about three to five times more effort. We qualify these findings, however, by noting that we are comparing highly-experienced, professional NIST assessors with assessors that were recruited from a university environment. It would have been ideal if the *same* assessors performed both evaluations, but unfortunately such a study was not practical. Nevertheless, since our assessors were either students or recent graduates of a Masters program in library and information science, we would expect their outputs to be reasonable in quality.

In the terminology of Bailey et al. [4], we would characterize our assessors as "silver" in contrast to the "gold" standard NIST assessors. However, they found that "silver" judgments can nevertheless be a reasonable proxy. Regardless, our focus here is on effort (time), and there is no principled reason to believe that our assessors are inherently faster than NIST assessors. In fact, quite the opposite: we would expect seasoned NIST assessors to be more efficient. Nevertheless, any inherent difference in quality (which we examine in Section 4.3) is unlikely to account for the large differences in assessment effort. Thus, we are confident in our conclusion that the nugget-based methodology requires more assessor effort than the cluster-based methodology. This seems intuitive and is borne out empirically.

4.2 Nugget vs. Cluster Differences

In our next set of analyses, we attempt to descriptively characterize differences between nuggets and clusters. Of the 22 topics that were included in our final dataset, there were 21,635 pooled updates. Of these, 12,929 (59.76%) were judged relevant according to the cluster-based methodology (i.e., added to a cluster). In contrast, with the nugget-based methodology, NIST assessors only found 8,131 (37.58%) relevant updates, i.e., that matched at least one nugget. This gap might be partially explained by differences between "gold" vs. "silver" assessors: Bailey et al. [4] observed that non-gold assessors appear to be less discerning in their judgments and thus tend to find more material relevant. However, in Section 4.3 we offer a more principled explanation of why some updates may be considered relevant with clusters but not with nuggets.

Topic	Updates	Nuggets	Clusters	Rel. w/ nuggets	Rel. w/ clusters
TS13.1	779	56	58	431	498
TS13.2	912	89	60	381	561
TS13.3	762	139	65	211	533
TS13.4	1463	97	62	410	1064
TS13.5	1069	108	61	82	830
TS13.6	1517	418	82	493	619
TS13.8	1128	88	19	172	849
TS13.9	873	45	10	168	664
TS13.10	610	37	9	287	492
TS14.11	1149	226	54	392	462
TS14.12	813	72	14	184	306
TS14.13	668	68	8	328	537
TS14.14	1382	76	57	448	618
TS14.15	908	45	27	315	270
TS14.16	905	72	63	554	403
TS14.17	1002	48	51	770	460
TS14.18	1076	89	34	409	889
TS14.19	926	97	18	341	640
TS14.20	760	35	22	341	603
TS14.21	1225	124	37	869	983
TS14.22	766	116	35	228	297
TS14.23	942	138	33	317	351
Total	21635	2283	879	8131	12929
Mean	983.41	103.77	39.95	369.59	587.68
Median	919	88.5	36	341	549

Table 1: Descriptive statistics for our dataset comprising 22 topics from TS 2013 and TS 2014.

A detailed topic-by-topic breakdown of our dataset is presented in Table 1, which shows the number of updates in the pool, the number of discovered nuggets and clusters, as well as the number of updates that were found to be relevant based on the nugget and cluster judgments. Aggregate statistics are shown at the bottom of the table. We see that there are over twice as many nuggets as there are clusters per topic—in fact, with the exception of TS13.1 and TS14.17, all topics have more nuggets than clusters. The largest difference comes from TS13.6 (Hurricane Sandy),[4] where the nugget-based methodology generated 418 nuggets, compared to only 82 clusters. This suggests that the nugget-based approach is attempting to capture more information, or at least information at a more fine-grained level.

To examine these differences in more detail, we manually analyzed nuggets and clusters for a single topic, TS14.15 (Port Said Stadium riot)[5] which has 45 nuggets and 27 clusters. As a first step, we organized all nuggets and clusters into general categories (i.e., themes), and then compared each in terms of coverage. Our results

[4] http://en.wikipedia.org/wiki/Hurricane_Sandy
[5] http://en.wikipedia.org/wiki/Port_Said_Stadium_riot

Category	N	C	Relationship
What Happened	12	4	Nuggets provide more diverse coverage of what occurred during the event. The clusters capture similar information but in a more coarse-grained manner.
Killed / Injured Reports	9	4	The nuggets are more detailed, specifying where deaths/injuries occurred and how. The clusters capture mentions of killed/injured counts more generically.
Reactions	7	10	All nuggets have corresponding clusters. Additional clusters capture reactions from notable actors.
Contextual Information	4	0	Contextual information, such as information about the lead up to the riot is only covered by the nuggets.
Conspiracy Discussions	4	1	Multiple perspectives are represented as nuggets. These diverse perspectives were all mapped to a single cluster.
Generic Event Information	3	0	Three nuggets were defined covering the initial report of the event and date information. There were no corresponding clusters; instead, the information is covered implicitly within other clusters.
Who Was Involved	3	0	There are explicit nuggets representing mentions of specific people involved. Clusters from other categories implicitly capture this information.
Later Protest	3	8	The clusters provide broader coverage of the protests that occurred on the following days after the original event, in comparison to the nuggets.

Table 2: Breakdown of nuggets (N) and clusters (C) into categories for TS14.15 "Port Said Stadium riot".

are summarized in Table 2, which lists the category, number of nuggets (N) and clusters (C) that belong to each category, and a description of the differences.

We see that the relative distribution of nuggets and clusters varies greatly across categories. Although there are more nuggets than clusters overall, some categories have more associated clusters than nuggets. This indicates that the information covered by each differs. For example, the "Killed / Injured Reports" category is covered in more detail by nuggets, while the "Later Protest" category is covered in more detail by the clusters. For the purposes of evaluating timeline summaries, this suggests that each methodology favors different types of content.

This analysis also shows that there are nuggets for which the corresponding clusters do not contain *any* coverage. Two of these categories are "Who Was Involved" and "Generic Event Information": this appears to be the result of information co-occurrence within the updates. In particular, during the reporting of an event, there are some pieces of information that never appear on their own, but instead appear alongside other pieces of information. For instance, there are a variety of updates along the lines of "40 killed in Port Said stadium riot". Under the nugget-based methodology, such updates would receive credit for covering the nuggets "40 killed" ("Killed / Injured" category) and "Port Said stadium riot" ("Generic Event Information" category). However, under the cluster-based methodology, the assessor needs to select a single cluster for the update. Typically, the update is added to the cluster covering what the assessor sees as the most important information, which was "40 killed" in this case. If "Port Said stadium riot" (Generic Event Information) is never the most important piece of information in an update, it will never receive its own cluster.

The prevalence and distribution of nuggets vs. clusters is determined to a large extent from their sources. Recall that for the nugget-based approach, the assessors first identified nuggets based on an external resource (Wikipedia in our case), whereas clusters are directly formed from system updates. Thus, clusters are more likely to emphasize easy-to-obtain information, whereas nuggets are closer to an external "objective truth".

Method	Adjusted Rand	Adjusted MI
random	0.1734	0.1302
highest grade	0.1735	0.1312
earliest	0.1657	0.1287
most popular	0.1709	0.1302

Table 3: The Adjusted Rand Index and Adjusted Mutual Information between clusters and different techniques for "projecting" nuggets into clusters.

Another way to compare nuggets and clusters is to take advantage of standard metrics used to compare two clusterings of the same data. One common metric is the Adjusted Rand Index [19], a measure of similarity between two clusterings that is corrected for chance groupings (ranging from -1 to $+1$); another common metric is Adjusted Mutual Information [21], a variant of the more familiar mutual information metric, but corrected for agreement solely due to chance. We can take the nuggets and "project" them into clusters based on what nuggets were assigned to each update. There are four reasonable ways in which this could be accomplished: for a given relevant update, (i) select a *random* nugget as the cluster label; (ii) select the highest-graded nugget as the cluster label (breaking ties randomly); (iii) select the *earliest* nugget as the label; (iv) select the most *popular* nugget across all relevant updates as the label.

These four heuristics describe different ways to "project" nuggets into clusters, which we can then compare against the output of the cluster-based evaluation. Results in terms of the Adjusted Rand Index and Adjusted Mutual Information are shown in Table 3. These values are quite low, indicating that there are substantial differences between the nuggets and the clusters. However, we observe high agreement between the results of each heuristic. As a point of

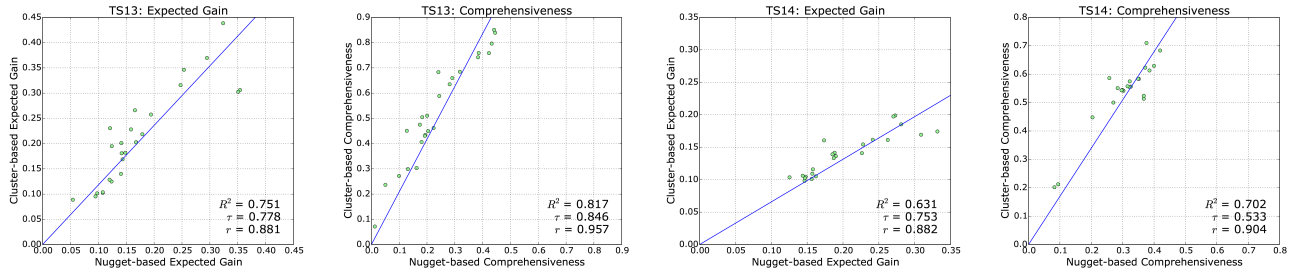

Figure 2: Scatterplots showing correlations between metrics computed using the nugget- and cluster-based evaluation methodologies based on participants' runs. Each plot shows the result of a linear regression as well as Kendall's τ and Pearson's r.

reference, in the analysis of the clustering task on tweet data by Wang et al. [24], they reported an Adjusted Rand Index of 0.445 for clusters generated from scratch by two different assessors. The agreement we observe here is much lower, which suggests that at a fundamental level, forming nuggets and clusters involve very different processes.

In summary, we observe substantial differences between nuggets and clusters, and can characterize the differences as follows: Overall, nuggets are more fine-grained, and since they are generated from an external source, they can cover information that is not retrieved by any system. Clusters are coarse-grained and can conflate information that is explicitly covered by multiple nuggets.

4.3 Score and Rank Correlations

The ultimate goal of any evaluation is, of course, to assess the effectiveness of systems for accomplishing a particular task. Thus, we want to know: Do system scores and rankings generated using clusters correlate with those generated using nuggets? To answer this question, we adopt two metrics originally proposed by Qi et al. [9, 10] and further refined by the TREC Temporal Summarization Tracks: expected latency gain, shown in Equation (1), and comprehensiveness, shown in Equation (2).

Given a set of updates $\mathcal{D}$, expected latency gain is defined as:

$$\mathbf{ELG}_{\mathbf{V}}(\mathcal{D}) = \frac{1}{\sum_{d \in \mathcal{D}} \mathbf{V}(d)} \sum_{d \in \mathcal{D}} \mathbf{G}(d, \mathcal{D}) \tag{1}$$

where, $\mathbf{V}(d)$ computes the verbosity normalization for update d, and $\mathbf{G}$ captures the amount of gain contained in the update. Once a nugget is observed in an update, subsequent occurrences of the nugget do not contribute to gain (i.e., systems are not rewarded for returning the same information multiple times). For simplicity and ease of interpretation, we consider each update to have unit verbosity, i.e., $\mathbf{V}(d) = 1$ for all d. However, we did repeat our analysis with the verbosity normalization as actually defined in the Temporal Summarization Track; our findings are not affected.

To keep our analysis consistent across nugget- and cluster-based evaluations, we maintain binary relevance for nuggets (since our clusters do not have relevance grades). The original definition of the gain function also includes a temporal discount to penalize systems for returning "late" information—since our focus here is on output quality and not timeliness, we removed this confound by eliminating the latency penalty from $\mathbf{G}$. Thus, we refer to the metric more accurately as *expected gain*. Finally, in all cases, we removed all updates that were not judged from $\mathcal{D}$.

Comprehensiveness is defined as:

$$\mathbf{C}(\mathcal{D}) = \frac{1}{\sum_{n \in \mathcal{N}} \mathbf{R}(n)} \sum_{d \in \mathcal{D}} \mathbf{G}(d, \mathcal{D}) \tag{2}$$

where $\mathbf{R}(n)$ is the relevance grade for nugget n. As explained above, for our experiments $\mathbf{R}(n) = 1$ for all $n \in \mathcal{N}$. We used the same definition of $\mathbf{G}$ as in Equation (1).

Summarizing, we can interpret expected gain as precision and comprehensiveness as recall. Evaluation with clusters is the same, except that the set of nuggets $\mathcal{N}$ is replaced by the set of clusters. With clusters, expected gain captures the fraction of updates having a unique cluster membership, and comprehensiveness computes the fraction of clusters represented in a system's updates.

Figure 2 shows correlations between the scores produced by the nugget-based and cluster-based evaluation methodologies based on participants' runs. We separately plot topics from TS 2013 and TS 2014, for expected gain and comprehensiveness. For each plot, we show the results of a linear regression and report the R^2 value. In addition, we show Kendall's τ and Pearson's r. Kendall's τ is most commonly used to capture the robustness of evaluation results with respect to different judgments because it captures rank correlations: for information retrieval experiments, we are in general more concerned with system rankings and less concerned with the absolute values of the metrics. With the exception of comprehensiveness for TS 2014, the Kendall's τ values we observe are in the range generally considered to be "good agreement".

To gain a better understanding of the differences between nugget- and cluster-based results, we conducted a topic-by-topic analysis as follows: for each topic, we generated a scatterplot where each point represents a pair of runs from that particular evaluation. The x coordinate denotes the difference in the nugget-based metric between the two runs and the y coordinate denotes the difference in the cluster-based metric (similar to the analyses by Qian et al. [18]). Since the absolute scores for each topic vary, we normalize the differences to between -1 and $+1$ with respect to the maximum absolute score difference to better focus on the metric correlations. Figure 3 shows these scatterplots for expected gain, and Figure 4 shows these scatterplots for comprehensiveness. All topics are shown using Tufte's "small multiples" visualization technique: the

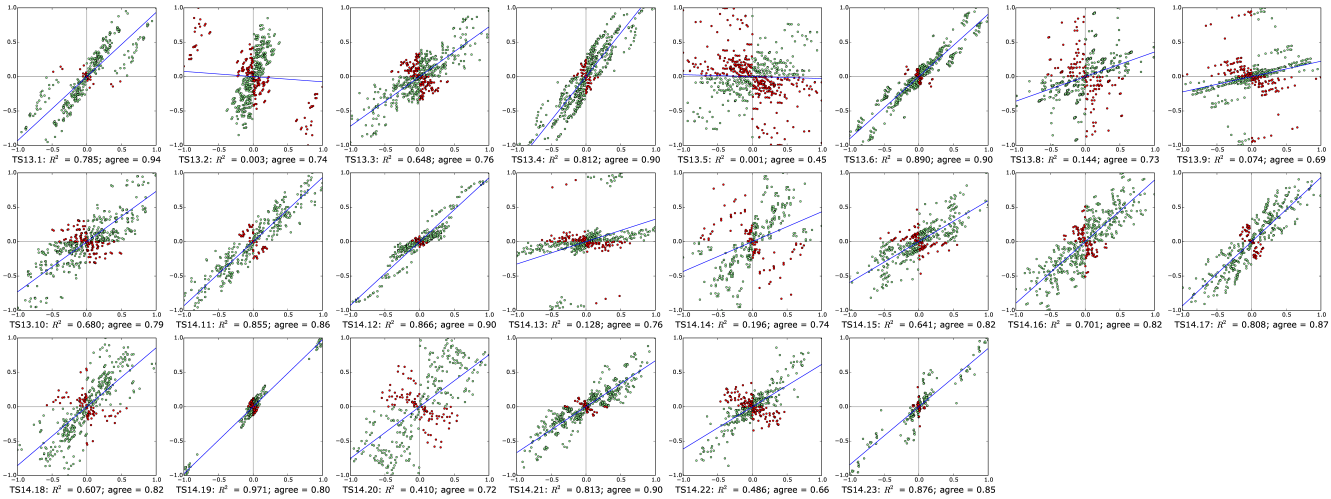

Figure 3: For expected gain, scatterplots organized in "small multiples" for each topic, showing differences in the nugget-based metric (x axis) vs. cluster-based metric (y axis) with the results of a linear regression.

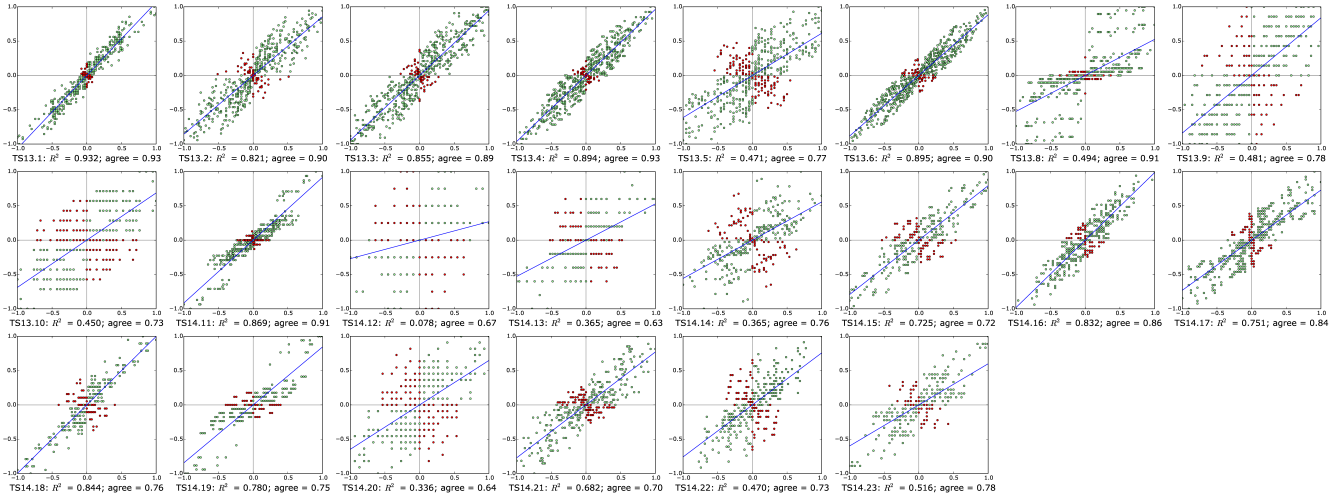

Figure 4: For comprehensiveness, scatterplots organized in "small multiples" for each topic, showing differences in the nugget-based metric (x axis) vs. cluster-based metric (y axis) with the results of a linear regression.

main goal is to convey an overall sense of metric correlations across topics, as opposed to showing any particular topic in detail.

In each individual scatterplot we show the results of a linear regression. Perfect metric correlation would manifest as a perfectly linear relationship. Points in the upper right (first) and lower left (third) quadrants are those in which the nugget-based and cluster-based metrics agree—these points are shown in green. Points in the upper left (second) and lower right (fourth) quadrants are those in which the metrics disagree—these points are shown in red. Ideally, we would want points close to the diagonal $y = x$: that is, changes in one metric yields a proportional change in the other metric. Points in the first and third quadrants but away from the diagonal $y = x$ indicate overall agreement (i.e., both metrics agree on which system is better), but not on the magnitude of the differences. Note that we are not particularly concerned with red points clustered around the origin: although these represent disagreements, they are cases where the metric differences are small [22].

The percentage of pairs in agreement (i.e., in the first and third quadrants) is 79.2% for expected gain and 79.6% for comprehensiveness across all topics. In general, we do see that for most topics, points lie fairly close to the diagonal $y = x$. For the expected gain measure (Figure 3), 12 of 22 topics have a pairwise agreement of over 80% with only one topic (topic TS13.5) showing more disagreements than agreement (45%). For comprehensiveness (Figure 4), 9 of 22 topics have pairwise agreement over 80%, with the lowest

agreement being 63% for topic TS14.13. Echoing the results in Figure 2, there seems to be closer overall agreement in expected gain than comprehensiveness.

However, some topics exhibit high levels of disagreements. One source of disagreement is immediately obvious for comprehensiveness: for topics with relatively few clusters, the range of values that is possible for comprehensiveness is by definition limited. TS14.12 and TS14.13 are two examples. This is a direct effect of the fact that clusters are more coarse-grained than nuggets.

To gain more insight, we also performed some manual analyses. Consider the expected gain for topic TS13.5 "Hurricane Isaac": it has the lowest rank correlation (Kendall's $\tau = -0.115$) as well as the lowest agreement. From Table 1, we see that the cluster-based evaluation found over 10 times more relevant updates than the nugget-based evaluation, which seemed odd. This resulted in many more submitted updates contributing to the expected gain, leading to the observed evaluation disagreement. For instance, the following sentence was marked as relevant under the cluster-based evaluation but not relevant under the nugget-based evaluation:

> Mississippi River flowed backwards due to Isaac. This velocity hydrograph shows the velocity of the Mississippi River during Hurricane Isaac's landfall.

On the one hand, it is reasonable to consider this update relevant, as it talks about an effect of Hurricane Isaac. However, when building a summary of the event, it can be argued that this update does not contain any important nuggets of information, as "Mississippi River flowed backwards" is mostly an attention-grabbing headline. Recall that under the nugget-based evaluation, the Wikipedia page for the event[6] was used as the ground truth source of the nuggets. This Wikipedia page covers a variety of information about Isaac's effect on the state of Mississippi, but not the river directly. Hence, no nuggets were explicitly created about the river during nugget extraction, and therefore this update was correctly judged as non-relevant during the matching phase. In contrast, under the cluster-based evaluation, this update was considered relevant to a broad cluster about "flooding in Mississippi".

This result highlights a key difference between the two evaluation methodologies. The nugget-based evaluation is more focused on 'key' information about an event, driven by information contained in an external ground-truth source. This means that systems will not always receive improved scores for identifying more relevant updates, if information contained in those updates was not important enough to be captured in the external source. In contrast, under the cluster-based methodology, relevant updates are likely to enter the pool and be added to a cluster, and hence will contribute to the final scores. This enables a better estimate of recall within a summary, but can reward systems for returning trivial or otherwise non-salient content.

One caveat of our discussion is the confound between gold (NIST) vs. silver (our) assessors, previously mentioned Section 4.1. However, we cannot think of any plausible systematic differences in the assessors that could affect our overall findings. Much of what we observed seems to be attributable to the nature of the information needs and system output, not assessor characteristics.

[6] https://en.wikipedia.org/wiki/Hurricane_Isaac_(2012)

5 DISCUSSION

Having quantitatively and qualitatively analyzed the nugget-based and cluster-based evaluation methodologies, we next discuss their advantages and disadvantages:

Nugget-based methodology: The core advantage of this approach is that it uses an explicit definition of what a "good" summary is, based on an external reference source. This has three useful consequences. First, by scoring updates against a set of explicit information nuggets, the relative difference between updates that cover a single piece of information and those that cover multiple pieces of information can be better expressed, as illustrated in Section 4.2. This may result in more accurate summary information coverage estimates for individual updates. Second, it is easy to explain why a summary timeline received a particular score for an event during failure analysis (e.g., it missed nuggets X and Y, or it returned too many redundant updates covering Z). Third, by using manually-authored Wikipedia pages as the basis for nugget extraction, the resulting information nuggets will only cover information that was considered sufficiently important to be included in the Wikipedia page. The effect of this is that systems returning trivial or non-salient information will not be rewarded.

However, the nugget-based methodology is built on the assumption that Wikipedia (at a particular moment in time) provides a comprehensive ground truth from which information nuggets can be extracted. If a timeline summarization system includes some useful or important information that was not contained in Wikipedia (for example, a future system that uncovers a new fact that all previous systems and human editors missed), then the system will not get credit for it. Or alternatively, a fact didn't come to light about an event until much later, after nuggets had been extracted from the Wikipedia page. Furthermore, we might imagine a future scenario where timeline generation systems are used to help humans update Wikipedia pages, which would introduce systematic biases in an evaluation that relies on Wikipedia as the source of ground truth. However, to our knowledge no such system is being deployed in this capacity, so we're safe, at least for now.

Of course, the single biggest drawback of the nugget-based methodology is its cost, as we detailed in Section 4.1. Our results show that this approach requires anywhere from three to five times more effort compared to the cluster-based approach.

Cluster-based methodology: The most notable advantage of the cluster-based approach is the amount of effort required, which is far less than the nugget-based approach. Furthermore, as clusters are not dependent on an external data source like Wikipedia, this methodology avoids omissions from incompleteness in the ground truth. This, of course, is a double-edged sword, because pooling depends on having systems that contribute relevant material, so for difficult topics, clusters might be lacking in coverage.

With clustering, there is a one-to-one relation between updates and clusters. This may be problematic in cases where an update covers multiple pieces of information that appear independently. For instance, consider the update "3 people were killed and another 315 were injured in a train crash at Once station in Buenos Aires".

There are multiple atomic pieces of information included in this update, such as "3 people were killed", "315 were injured" and that it happened at "Once station in Buenos Aires". If we assume that at the point that this update was examined the assessor had already created two relevant clusters, one that represents information about "fatality counts" and one that represents "event location" information, then it is unclear which of these clusters the update should be added to, since it covers both. The assessor is forced to decide, which may magnify assessor differences. In addition, the assessor is forced to decide when a piece of information is substantially different to warrant a new cluster. For example, if we have a cluster that represents the information "2 people were killed", and then the assessor encounters an update that says "3 people were killed", she has a choice: Either she can add the update to the existing cluster, thereby generalizing that cluster from "2 people were killed" to "information about fatalities", or she can create a new cluster to represent the new information that "3 people were killed". It is not clear which option is better.

There is one final issue worth discussing regarding the reusability of nugget- and cluster-based test collections. We are not aware of any systematic comparison, so the best we can do is to offer some informed commentary. Nugget-based test collections are not reusable, in the sense that scoring a system that did not participate in the original evaluation requires manual effort (the nugget matching process) and yields a score that is not comparable to other systems (due to assessor differences in the matching). Although automatic techniques based on *n*-gram overlap alleviate this to some extent [12], particularly during system development, accurate summative evaluations still require human effort. However, it is not clear if cluster-based test collections are reusable either—although judgments can be reused, the task is precision focused, and thus all existing evaluations have used relatively shallow pools. This is a question that warrants further study.

6 FINAL RECOMMENDATIONS

Based on our analyses, neither nuggets nor clusters represent a one-size-fits-all solution, as both have advantages and disadvantages. Instead, we offer some general recommendations:

- If cost and effort are important considerations, use the cluster-based approach.
- When using the cluster-based approach, it is important that a diverse set of systems contribute to the pool, since that represents the entire universe of information that will be considered in the evaluation.
- For particularly difficult events or in cases where diverse systems are not available, using an external source of ground truth might be preferable.
- If a focus on 'key' information or accurate failure analysis is desired, then nuggets are preferable. Wikipedia indirectly captures human editorial judgments on what's important. On the other hand, if an external source is not available, particularly for smaller events, nuggets are not a workable option.

Bottom line: nuggets or clusters? Like most questions that involve tradeoff, it depends.

7 ACKNOWLEDGMENTS

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the EC co-funded SUPER (FP7-606853) project. Findings, conclusions, or recommendations expressed do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. 2014. TREC 2014 Temporal Summarization Track Guidelines.
[2] Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. 2015. TREC 2015 Temporal Summarization Track Overview. In *TREC*.
[3] Javed Aslam, Matthew Ekstrand-Abueg, Virgil Pavlu, Fernando Diaz, and Tetsuya Sakai. 2013. TREC 2013 Temporal Summarization. In *TREC*.
[4] Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. 2008. Relevance Assessment: Are Judges Exchangeable and Does it Matter? In *SIGIR*. 667–674.
[5] Gaurav Baruah, Haotian Zhang, Rakesh Guttikonda, Jimmy Lin, Mark D. Smucker, and Olga Vechtomova. 2016. Optimizing Nugget Annotations with Active Learning. In *CIKM*. 2359–2364.
[6] John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2011. Nouveau-ROUGE: A Novelty Metric for Update Summarization. *Computational Linguistics* 37 (2011), 1–8.
[7] Hoa Dang. 2005. Overview of DUC 2005. In *DUC*.
[8] Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 Opinion Question Answering and Summarization Tasks. In *TAC*.
[9] Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Fernando Diaz. 2016. A Study of Realtime Summarization Metrics. In *CIKM*. 2125–2130.
[10] Qi Guo, Fernando Diaz, and Elad Yom-Tov. 2013. Updating Users about Time Critical Events. In *ECIR*. 483–494.
[11] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *ACL Workshop On Text Summarization*.
[12] Jimmy Lin and Dina Demner-Fushman. 2006. Methods for Automatically Evaluating Answers to Complex Questions. *Information Retrieval* 9, 5 (2006), 565–587.
[13] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2014. Overview of the TREC-2014 Microblog Track. In *TREC*.
[14] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2015. Overview of the TREC-2015 Microblog Track. In *TREC*.
[15] Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014. Incremental Update Summarization: Adaptive Sentence Selection based on Prevalence and Novelty. In *CIKM*. 301–310.
[16] Ani Nenkova and Kathleen McKeown. 2011. Automatic Summarization. *FnTIR* 5, 2–3 (2011), 103–233.
[17] Paul Over. 1997. TREC-6 Interactive Report. In *TREC*.
[18] Xin Qian, Jimmy Lin, and Adam Roegiest. 2016. Interleaved Evaluation for Retrospective Summarization and Prospective Notification on Document Streams. In *SIGIR*. 175–184.
[19] William M. Rand. 1971. Objective Criteria for the Evaluation of Clustering Methods. *JASA* 66, 336 (1971), 846–850.
[20] Luchen Tan, Adam Roegiest, Charles L. A. Clarke, and Jimmy Lin. 2016. Simple Dynamic Emission Strategies for Microblog Filtering. In *SIGIR*. 1009–1012.
[21] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information Theoretic Measures for Clusterings Comparison: Is a Correction for Chance Necessary? In *ICML*. 1073–1080.
[22] Ellen M. Voorhees. 1998. Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. In *SIGIR*. 315–323.
[23] Ellen M. Voorhees. 2003. Overview of the TREC 2003 Question Answering Track. In *TREC*. 54–68.
[24] Yulu Wang, Garrick Sherman, Jimmy Lin, and Miles Efron. 2015. Assessor Differences and User Preferences in Tweet Timeline Generation. In *SIGIR*. 615–624.
[25] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary Timeline Summarization: a Balanced Optimization Framework via Iterative Substitution. In *SIGIR*. 745–754.
[26] ChengXiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. In *SIGIR*. 10–17.
[27] Chunyun Zhang, Zhanyu Ma, Jiayue Zhang, Weiran Xu, and Jun Guo. 2015. A Multi-level System for Sequential Update Summarization. In *QSHINE*. 144–148.