Time-aware Evaluation of Cumulative Citation Recommendation Systems

Laura Dietz and Jeffrey Dalton
Center for Intelligent Information Retrieval
School of Computer Science
University of Massachusetts, Amherst
{dietz,jdalton}@cs.umass.edu

Krisztian Balog
University of Stavanger
N-4036 Stavanger, Norway
krisztian.balog@uis.no

ABSTRACT

The goal of stream filtering systems is to identify relevant items over time. However, systems are often evaluated in a time-agnostic fashion, where results are evaluated as a batch. This work introduces a time-aware evaluation paradigm to study time-dependent characteristics of system effectiveness, such as performance degeneration over time. A particular challenge is posed by bursts in the volume of relevant documents in the ground truth, caused by specific events and trends. We introduce burst-aware weighting to arrive at a time-aware comparison across systems. As a motivating application, we re-evaluate the submissions to the TREC 2012 Knowledge Base acceleration track. Our evaluation paradigm is able to consistently distinguish teams by performance. We confirm that choices of time-granularity and burst-aware weighting schemes affect the results.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software

Keywords

Time-aware evaluation, information filtering, knowledge base acceleration, cumulative citation recommendation

1 Introduction

In order to stay up-to-date about new developments in particular areas, a growing number of people subscribe to microblogs, blogs, and news. Stream filtering systems monitor these information sources over time and alert users about new relevant information. Although stream filtering has a strong temporal element to it, systems are often evaluated as a batch, with no awareness of how prediction successes are distributed over time. We argue that it is important to the user to have good filtering results at any point in time, and this should be reflected in the evaluation paradigm.

In the following, we focus on the filtering task introduced by the TREC 2012 Knowledge base acceleration track (KBA). The cumulative citation recommendation task (CCR) is to identify stream documents that are "central" to a given set of target entities from Wikipedia, to enable timely updates of articles. Since KBA systems are trained on a period immediately before the evaluation time range, one might expect system performance to degrade over time due to topic drift. In this paper we devise ways to evaluate CCR in a time-aware manner. Our work represents a first attempt towards the more general problem of temporal evaluation for streaming collections.

We particularly address situations in which the number of ground truth documents changes drastically over time, as in the example query [Mario Garnero] depicted in Figure 1.

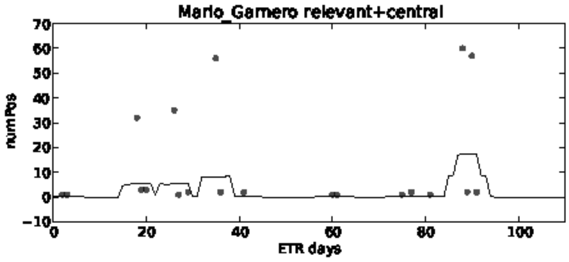

Figure 1: Number of true relevant documents for the KBA target entity [Mario Garnero]. Each point represents a day and the solid line is a seven day moving average.

The following two usage scenarios motivate our time-aware paradigm:

U1 A system developer wants to know how fast the performance of the system degenerates after the last training period. She only has access to a ground truth of bursty nature and she is afraid that this might occlude the true performance over time.

U2 An IT consultant has to choose among different stream filtering systems for a news agency. His client is anxious not to miss important hypes, therefore cares in particular about the system's quality at periods of high intensity bursts.

We focus on a usage scenario where a user checks the filtered documents at different time points, where all documents that arrived within the last period are equally recent. Alternative usage models are studied by Dong et al. [2].

Our evaluation framework is based on three main components: (i) subdividing the entire time interval into slices, (ii) measuring the relevance of documents within a given slice, and (iii) aggregating slice-based relevance scores into a single system-level score. We consider instantiations of this framework that divide the time interval into uniform slices and measure relevance within each slice using standard IR metrics, such as mean average precision, normalized discounted cumulative gain, R-precision, and the area under the ROC curve. The evaluation framework is designed to be generic enough to incorporate non-uniform time slicing and a variety of metrics (not

TAIA 13 August 1, 2013, Dublin, Ireland.
ACM xxx-xxx-xxx.

only rank-based but also set-based) for evaluating effectiveness in each slice.

2 Related work

The CCR task studied at TREC KBA is a stream filtering task related to previous work in the TREC Filtering track [7] and Topic Detection and Tracking (TDT) [1]. In filtering tasks, systems make a binary decision to accept or reject a document when it arrives. The result is a set of unranked documents that pass the filter. Similar to CCR, the primary evaluation measures are set-based: scaled utility [7] and a variant of the F-measure (F-Beta) [6]. Zhang et al. [8] extend filtering evaluation to account for novelty and redundancy. It is also related to the task of routing, which produces ranked lists of documents according to a profile. Previous research in these areas evaluates systems using batch evaluation across the entire test time period. In contrast, in this work we introduce new evaluation methods that capture how filtering effectiveness changes over time. Our framework can use both the set based measures from filtering as well as ranked evaluation used in routing.

In this work we model the temporal dynamics [5] of filtering system across points in time, creating a time series of effectiveness evaluation points. Particular interest has been devoted to treating recency, the fact that recent relevant documents are more valuable to a user then older relevant documents. A solution and evaluation paradigm is proposed by Dong et al. [2], discounting relevant documents by age. This idea is compatible with our framework, however it does not apply to our user model. Furthermore, Jones and Diaz [4] characterize the temporal categories of queries: atemporal (no regularities), temporally unambiguous (a single spike), and temporally ambiguous (several episodes). These categories give rise to different aggregation paradigms in our framework.

3 Example Application: TREC KBA

This section describes the task of our example application TREC 2012 KBA track, the corresponding evaluation methodology and highlights challenging issues.

3.1 Cumulative Citation Recommendation

Knowledge base acceleration (KBA) refers to activities aimed at reducing efforts associated with the maintenance of knowledge bases, like Wikipedia. In 2012, the Text REtrieval Conference (TREC) series launched a KBA track [3]. The task studied there is cumulative citation recommendation (CCR): Given a textual stream consisting of news and social media content and a target entity from a knowledge base (Wikipedia), generate a score for each document based on how pertinent it is to the input entity. The motivation is to enable Wikipedia editors to generate updates in a timely manner in response to news events.

The KBA 2012 stream corpus consists of web data crawled from news, forums, blogs, and URLs shortened at bitly.com between October 2011 and May 2012. The collection contains approximately 367 million English documents. The data set is divided into two periods, the training time range (TTR) from 2011 October through December and the evaluation time range (ETR) from 2012 January to May.

The KBA 2012 topic set contains 29 target entities, represented by their corresponding Wikipedia articles. For each of these entities, the filtering system can access the entity's Wikipedia page and sample training documents from the training time range to build an entity profile and possibly train a supervised model.

3.2 Official Evaluation Methodology

Each stream document is annotated with relevance judgments on a four point scale (garbage, neutral, relevant, and central). The evaluation considers two ways of arriving at a set of positive ground truth documents: only central documents, or the union of relevant and central documents.

CCR systems are required to process the collection in stream order and to assign a confidence score between (0,1000] to each citation-worthy document. The official evaluation metrics are precision, recall, F1, and scaled utility. As these measures apply to a predictions of sets, a cutoff threshold τ on the confidence score divides the stream into positive/negative prediction sets, to be compared against the positive/negative classes defined in the ground truth. Measures are computed for each query entity and averaged to arrive at the final system score (i.e., macro-averaging is used).

3.3 Evaluation Challenges

Below, we identify two main challenges with regards to the current TREC KBA evaluation that we address.

Stream nature. Despite the temporal aspects of the prediction task, the official evaluation is agnostic to time-dependent characteristics of the system. A system may perform well on average, but may not perform well during important bursty events. Going further, systems may perform extremely well right after the training period, but degrade in effectiveness as time progresses. The difference to systems with continuous quality is not captured in the batch evaluation.

Cutoff threshold. The cutoff threshold τ is a hyper-parameter to the official evaluation paradigm. It is an open issue how to set the cutoff to guarantee fair comparison across systems.

We address both of these challenges. First, to capture the stream nature we divide the stream into a series of evaluation time intervals. For the second, we adapt the approach taken in evaluating routing tasks, and perform ranked evaluation using the provided system confidence levels. To combine these two, the slice-based relevance scores are aggregated into a single system-level score. Our approach does not require a confidence threshold parameter.

4 Time-aware evaluation

In this section we present our methodology for evaluating retrieval over a streaming collection. The development of a time-aware evaluation paradigm involves dealing with slicing the time interval, measuring performance per slice, and choosing appropriate aggregation schemes. While our primary focus throughout this paper is on the CCR task, we introduce an approach that constitutes a general framework for retrieval tasks with streaming nature.

4.1 Slicing time

We describe two different ways of slicing time. *Uniform slicing* divides time into intervals of equal length (t_i). *Non-uniform slicing*, on the other hand, allows for slices of varying length; this relates to a scenario where the user checks the stream at random time periods, or with higher frequency during bursts. Further, we make two simplifying assumptions: (i) slices are non-overlapping and (ii) we are unconcerned about slices that do not contain any relevant documents according to the ground truth. Figure 2 illustrates the two different slicing mechanisms.

Formally, I is a slicing of the entire time interval, where $i \in I$ is a given slice, defined in terms of start and end times: $i = [t_s, t_e]$.

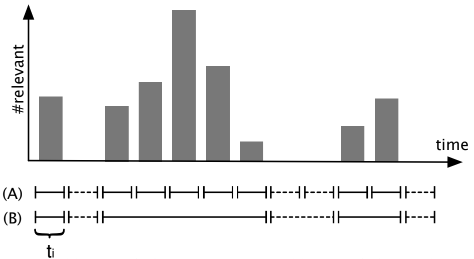

Figure 2: Uniform (A) and non-uniform (B) slicing. Number of relevant documents per time-slice. Dashed lines mark slices without relevant documents.

For brevity we only consider uniform slicing in this work, although the suggested evaluation paradigm also applies to non-uniform slicing.

4.2 Measuring slice relevance

Given a specific slice, the quality of filtered documents within that slice i.e., documents with timestamps within the slice interval) are evaluated using standard metrics.

We assume a usage scenario where at each check of the stream, a ranked list of documents is presented to the user, and therefore focus on ranking metrics and only examine slices with at least one relevant document. The slice-based effectiveness is computed as follows.

Let $\mathbf{d} =< d_1, \ldots, d_n >$ be a ranked list of documents, i is a time interval, $\mathbf{d}_i$ is a ranked list of documents within the interval i. We write $m(\mathbf{d}_i, q)$ to denote the evaluation score for the document ranking $\mathbf{d}_i$ given the query topic q.

We note that it is also possible to incorporate set-based metrics for $m(\mathbf{d}_i, q)$. It would lead to a slightly different usage scenario, where the user is assumed to process all documents with a relevance score above a predefined threshold. Since set-based metrics would involve an additional threshold parameter τ, we leave their exploration for future work.

4.3 Aggregating slice relevance

The final step in our evaluation framework is responsible for the aggregation of slice-based relevance scores. We propose a probabilistic formulation to estimate the likelihood of relevance, given a ranking, an input query q, and a slice-base evaluation metric m. Formally,

$$P(r = 1|\mathbf{d}, q, m) = \sum_{i \in I} P(r = 1|\mathbf{d}_i, q, i)P(i|q), \qquad (1)$$

where $P(r = 1|\mathbf{d}_i, q, i)$ indicates slice-based relevance and is approximated with $m(\mathbf{d}_i, q)$. Further, $P(i|q)$ denotes the importance of the time period i. When all slices are equally important (usage scenario U1), it is $P(i|q) = \frac{1}{I}$. In usage scenario U2, we deem time slices of bursts more important. Following the intuition that slices are more important, the more relevant documents they contain, we take $P(i|q)$ to be proportional to the number of relevant documents in that time period, denoted as $\#R(i, q)$:

$$P(i|q) = \frac{\#R(i, q)}{\sum_{i \in I} \#R(i, q)}. \qquad (2)$$

4.4 Correcting rank metrics for slice relevance

The above paradigm requires that the measure $P(r = 1|\mathbf{d}_i, q, i)$ is comparable across different time slices. We generate synthetic data to study several ranking measures with respect to their robustness against the changes in the total number of relevant documents R and ranking lengths $D = |d_i|$. We assume that the quality of a system is modeled by random perturbation of an ideal ranking. We generate rankings for given perturbation level θ and measure the ranking quality with a set of commonly used rank measures: MAP, NDCG, and R-Precision. We compute average value of each measure value across 1000 generated rankings for each setting of D and R. We vary $D \in \{10, 100, 200\}$ and $R \in \{\frac{D}{2}, \frac{D}{10}\}$.

We find that expectations of MAP and R-precision correlate well with the perturbation level without further corrections. The NDCG measure needs to be corrected by taking the cut-off rank R (equals the number of relevant documents), henceforth NDCG@R. The ROC-AUC measure correlates when rescaled to [-1,+1].

Table 1: Atemporal ranked evaluation.

Team	Run	MAP	NDCG@R	R-Prec
UvA	UvAIncLearnHigh	0.512	0.522	0.504
udel_fang	UDInfoKBA_WIKI1	0.482	0.600	0.542
LSIS	lsisRFAll	0.480	0.573	0.547
CWI	google_dic_3	0.454	0.462	0.455
UMass_CIIR	PC_RM10_1500	0.438	0.539	0.523
uiucGSLIS	gslis_adaptive	0.334	0.587	0.448
hltcoe	wordNER500	0.039	0.244	0.063
igpi2012	ner_jaccard	0.038	0.090	0.060
helsinki	disgraph2	0.019	0.027	0.022

Table 2: Temporal evaluation with uniform weighted slices.

Team	Run	MAP-weeks	MAP-days
UvA	UvAIncLearnHigh	0.540	0.619
udel_fang	UDInfoKBA_WIKI1	0.527	0.609
LSIS	lsisRFAll	0.523	0.604
CWI	google_dic_3	0.485	0.621
UMass_CIIR	PC_RM10_1500	0.497	0.591
uiucGSLIS	gslis_adaptive	0.365	0.411
hltcoe	wordNER500	0.061	0.070
igpi2012	ner_jaccard	0.038	0.053
helsinki	disgraph2	0.031	0.064

Table 3: Temporal evaluation with slices weighted by relevance.

Team	Run	MAP-weeks	MAP-days
UvA	UvAIncLearnHigh	0.568	0.640
udel_fang	UDInfoKBA_WIKI1	0.537	0.613
LSIS	lsisRFAll	0.546	0.622
CWI	google_dic_3	0.516	0.631
UMass_CIIR	PC_RM10_1500	0.519	0.602
uiucGSLIS	gslis_adaptive	0.386	0.431
hltcoe	wordNER500	0.051	0.063
igpi2012	ner_jaccard	0.052	0.068
helsinki	disgraph2	0.033	0.067

5 Experiments on TREC KBA runs

We now perform a full evaluation on all of the TREC KBA 2012 runs. In this evaluation the systems rank the stream documents by confidence with a given evaluation slice period. In all of these experiments we use uniform time slices and we leave non-uniform temporal slices for future work.

For all the experiments we use consider the following options for $m(\mathbf{d}_i, q)$: mean average precision (MAP), normalized discounted cumulative gain at rank R (NDCG@R), and precision at rank R (R-prec), where R is the number of relevant document within the slice. We follow convention in treating unjudged documents as non-relevant. For these experiments we use binary relevance, with documents that are annotated as "central" or "relevant" treated as relevant documents and all others as non-relevant.[1]

[1] More plots and analyses are available online at http://ciir.cs.umass.edu/~dietz/streameval/

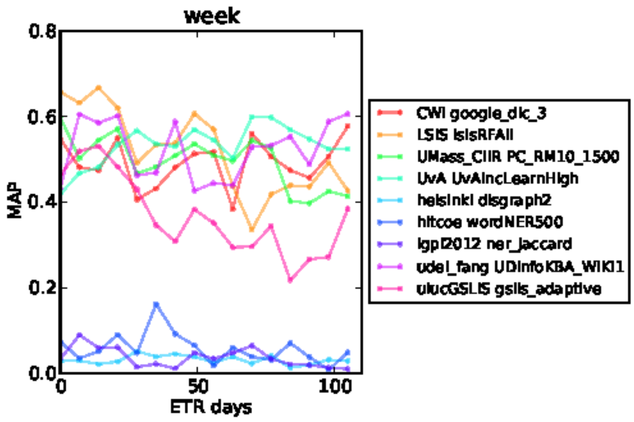

Figure 3: MAP with uniform weighting over time.

5.1 Ranked Evaluation

We first introduce ranked evaluation over the entire time period. This corresponds to batch evaluation similar to the TREC stream routing task. We compute these metrics for all runs from all teams. The best performing run by MAP was selected from each team and is shown in Table 1. Teams with zero effectiveness are omitted.

One key issue in the original set-based evaluation was differentiation between the runs. In contrast, rank based results show a wide spectrum of effectiveness with clear differences between the teams.

Our results show that UvA, udel_fang, and LSIS are the top three teams with respect to all three metrics. Where the UvA has high recall (measured in MAP), udel_fang, and LSIS have higher precision (measured in NDCG@R).

5.2 Uniform Temporal Slicing

In this section we perform a temporal evaluation with uniformly weighted time slices based on days and weeks. The results are shown in Table 2. We visualize the effectiveness of teams over the evaluation period using MAP in Figure 3.

We observe that for MAP the results are very similar to the atemporal evaluation. This indicates that the system effectiveness for most systems is stable over time. However in Figure 3 we observe that the effectiveness of LSIS and uiucGSLIS degrade over time. For the day-by-day evaluation in Table 2. CWI improves to be the top ranked system, comparable to UvA, demonstrating the importance of a slice granularity that reflects the user model.

5.3 Weighted Aggregation

In the previous section, all time slices are weighted uniformly. We now use the weighted aggregation model described in Section 4.3. The results are shown in Table 3.

Compared to the uniform evaluation, numbers for all teams are going up. udel_fang improves the least, where UvA, uiucGSLIS, and UMass_CIIR are gaining 0.2 in MAP-day.

We examine this phenomenon in more detail in Figure 4, where the MAP scores are scaled so that the area under the curve corresponds to scores in Table 3.

The figure shows that UvA consistently outperforms uiucGSLSIS in the uniform weighting, especially during the sparse period between day 40 to 80 (compare to Figure 1). However, the weighted analysis confirms that uiucGSLSIS performs reasonable during the bursts. As above we see that it is important that the weight aggregation matches the user model.

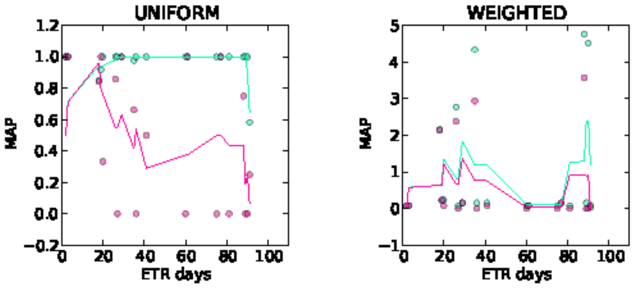

Figure 4: Influence of weighted aggregation comparing UvA (blue) with uiucGSLSIS (magenta) for entity [Mario Garnero].

6 Summary and outlook

We present a framework for temporal evaluation that tracks system effectiveness over time and applied it to revisit the evaluation of the TREC 2012 KBA CCR task. We show that ranked evaluation measures can effectively be used to compare CCR systems without the need for a confidence cutoff. We gain some insights into differing performance among teams, specifically, we are able to tell which systems degrade in performance over time. Whether uniform weighting or burst-aware weighting is appropriate, depends on the requirements of the user.

In future work, we plan to further consider the impact of burstiness and non-uniform importance weighting. The KBA Year 1 data contains only few entities with bursty ground truth—and yet we observe a difference between batch evaluation and slice-aggregated evaluation. When filtering microblog streams and longer time periods, time-aware evaluation will be even more important.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors' and do not necessarily reflect those of the sponsor. We especially thank John Frank for making the submissions to TREC KBA available to us.

References

[1] J. Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer, 2002.

[2] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz. Towards recency ranking in web search. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 11–20. ACM, 2010.

[3] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff. Building an entity-centric stream filtering test collection for TREC 2012. In *TREC '12*, 2013.

[4] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3), July 2007.

[5] J. Kleinberg. Temporal dynamics of on-line information streams. In *Data Stream Management: Processing High-Speed Data Streams*. Springer, 2004.

[6] C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

[7] S. E. Robertson and I. Soboroff. The TREC 2002 filtering track report. In *TREC'02*, 2003.

[8] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proc. of SIGIR*, SIGIR '02, pages 81–88. ACM, 2002.