Information Processing and Management 51 (2015) 791–808

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Evaluating document filtering systems over time

Tom Kenter[a,*], Krisztian Balog[b], Maarten de Rijke[a]

[a] *University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands*
[b] *University of Stavanger, Stavanger, Norway*

ARTICLE INFO

Article history:
Received 22 September 2014
Revised 8 February 2015
Accepted 27 March 2015
Available online 8 September 2015

Keywords:
Time-aware information retrieval
Evaluation
Significance testing

ABSTRACT

Document filtering is a popular task in information retrieval. A stream of documents arriving over time is filtered for documents relevant to a set of topics. The distinguishing feature of document filtering is the temporal aspect introduced by the stream of documents. Document filtering systems, up to now, have been evaluated in terms of traditional metrics like (micro- or macro-averaged) precision, recall, MAP, nDCG, F1 and utility. We argue that these metrics do not capture all relevant aspects of the systems being evaluated. In particular, they lack support for the temporal dimension of the task. We propose a time-sensitive way of measuring performance of document filtering systems over time by employing trend estimation. In short, the performance is calculated for batches, a trend line is fitted to the results, and the estimated performance of systems at the end of the evaluation period is used to compare systems. We detail the application of our proposed trend estimation framework and examine the assumptions that need to hold for valid significance testing. Additionally, we analyze the requirements a document filtering metric has to meet and show that traditional macro-averaged true-positive-based metrics, like precision, recall and utility fail to capture essential information when applied in a batch setting. In particular, false positives returned in a batch for topics that are absent from the ground truth in that batch go unnoticed. This is a serious flaw as overgeneration of a system might be overlooked this way. We propose a new metric, aptness, that does capture false positives. We incorporate this metric in an overall score and show that this new score does meet all requirements. To demonstrate the results of our proposed evaluation methodology, we analyze the runs submitted to the two most recent editions of a document filtering evaluation campaign. We re-evaluate the runs submitted to the Cumulative Citation Recommendation task of the 2012 and 2013 editions of the TREC Knowledge Base Acceleration track, and show that important new insights emerge.

1. Introduction

Document filtering is a popular task in information retrieval with many applications (Keiser, 2009; Amigó, Gonzalo, & Verdejo, 2011; Amigó, Gonzalo, & Verdejo, 2013; Robertson & Soboroff, 2002; Frank, Kleiman-Weiner, et al., 2013; Frank, Bauer, et al., 2013). A stream of documents arriving over time is filtered for documents relevant to a set of topics. The distinguishing feature of document filtering, that sets it apart from other document classification tasks, is the temporal aspect introduced by the stream of documents. Because of this temporal dimension, the performance of a system is

[*] Corresponding author.
E-mail addresses: tom.kenter@uva.nl (T. Kenter), krisztian.balog@uis.no (K. Balog), derijke@uva.nl (M. de Rijke).

http://dx.doi.org/10.1016/j.ipm.2015.03.005
0306-4573/© 2015 Elsevier Ltd. All rights reserved.

susceptible to change over time. For example, in a document filtering setting, where topics are being monitored over time, the topics might evolve. A system filtering the stream should be sensitive to this in order to perform well. As another example, a spam classifier should adapt to malignant and cunning adversaries. If it fails to do so effectively, its performance will likely degrade over time.

Standard evaluation metrics measure performance for all returned documents, for a given information need (i.e., query), in a single batch (e.g., P@n, recall, nDCG, AP) and they can be averaged over multiple requests (e.g., macro-precision across a set of queries). Recent studies propose to decompose document streams into sequences (e.g., into slices of equal size) and measure effectiveness on a given time period (Azzopardi, 2009; Dietz, Dalton, & Balog, 2013; Aslam, Ekstrand-Abueg, Pavlu, Diaz, & Sakai, 2013). Then, it becomes possible to monitor the changes in system performance over time and to measure performance as a weighted average of slice-based relevance scores. None of these approaches, however, addresses the question we are interested in: how system performance *changes* over time.

In Fig. 1 the performance of three hypothetical systems is plotted over time. The blue dots represent the score of the system at a given point in time. The gray dotted line represents the average performance of the systems over the entire time span. Clearly, all three systems have the same average performance. However, the performance of System A degrades rather strongly over time, the performance of System B less so, while the performance of system C shows improvement over time. With the metrics currently available there is no way to express this difference. In this paper we propose to capture this difference by employing trend analysis. In short, this entails fitting a straight line to the values of any existing performance metric applied to document filtering systems over time. In Fig. 1 these lines are displayed in orange. The derivative of the fitted line provides a simple and intuitive measure of the amount of change in performance over time. Ultimately, we can compare the performance of the three systems as estimated by trend analysis at the end of the evaluation period (the large orange dots in Fig. 1).

The main contributions of this paper are the following. We analyze the properties and requirements a document filtering metric should meet. We propose to measure performance in batches and to use trend estimation to measure performance over time. We show that traditional macro-averaged true-positive-based metrics, like precision, recall and F1 fail to capture essential information when applied in a batch setting. In particular, documents returned in a batch for topics that are absent from the ground truth in that batch, false positives, go unnoticed. We propose a new metric, aptness, that does capture false positives. We incorporate this metric in an overall score, F_{pra}, and show that this new score does meet all requirements. As an important aspect of evaluation is testing for significant differences in observations, we detail the tests for statistical significance for trend estimation and discuss the assumption that need to hold.

We test our method on the runs submitted to the Cumulative Citation Recommendation task of the 2012 and 2013 editions of the TREC Knowledge Base Acceleration track (*KBA CCR* for short). A re-evaluation of the results in terms of our proposed method shows a different ordering of teams, also at the top end. Moreover, while there were teams beating the baseline in 2013 when judged by the official metrics, our tests show that in fact no team did, when our proposed time-aware evaluation is used.

Additionally, we find that the assumptions needed for valid significance testing hold in a vast majority of cases considered.

The remainder of the paper is organized as follows. Related research is covered in Section 2. In Section 3 we give an overview our time-aware document filtering evaluation method and discuss the required properties. Two key components of our approach, measuring performance per time batches and trend estimation are then presented in Sections 4 and 5, respectively. In Section 5 we also show how significance testing can be performed. In Section 6 we detail our research questions and show results of experiments performed on the runs submitted to two years of the TREC KBA CCR evaluation campaign. We conclude in Section 7.

2. Related work

2.1. Document filtering

Document filtering is the task of identifying relevant items from an incoming stream of content. Different flavors of this problem have been studied in the past. Common to these is that documents arrive sequentially over time and relevance decisions for each topic must be made as soon as the document is processed. Topics represent long-term information needs (often referred to as "profiles") and may evolve over time.

Routing was one of the very first tasks studied at the Text REtrieval Conference (TREC) and ran at the first three editions of TREC (TREC-1-3) (Harman, 1994). Routing systems learn static profiles from training documents, and then rank documents in the test set according to these profiles. Consequently, performance evaluation relies on ranked-based measures. "After TREC-3, a strong argument was made that a more realistic filtering task should be developed in addition to routing." (Soboroff & Robertson, 2003). Early editions of the Filtering track (TREC-4-6) cast the task as a binary classification problem: to refer the document to the user or not. This necessitates a methodological departure from rank-based evaluation to the use of set-based measures (Soboroff & Robertson, 2003). Later editions (from TREC-7) further refine filtering into *batch filtering* and *adaptive filtering* tasks. Systems have to process test documents in chronological order and select a subset of them. This implies the use of thresholding for making the binary decision between selecting or discarding each document. Systems are

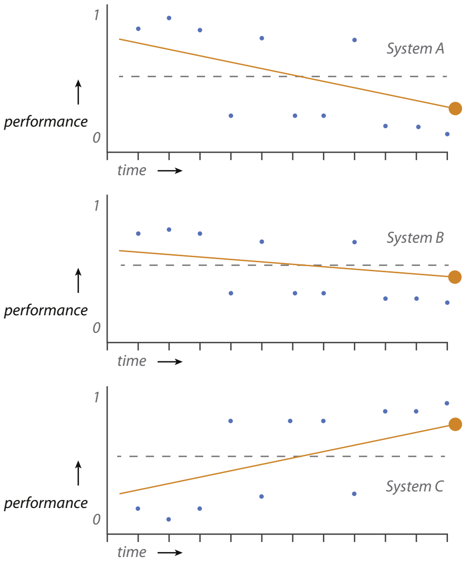

Fig. 1. Performance of three systems over time. Systems A and B degrade, while System C improves over time, but they all have the same average performance over the entire period. We express the change in system performance using the derivative of the fitted line (in orange) and compare performance at what we call the "estimated end-point" (the large orange dots). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

evaluated by calculating the Utility and F-beta measure of the unordered output set. The batch filtering task assumes that the initial profile and threshold remain constant. Adaptive filtering, on the other hand, assumes a scenario where "[e]ach selected document is immediately judged for relevance, and this information can be used by the system to adaptively update the filtering profile and/or adjust the threshold." (Nanas, Uren, De Roeck, & Domingue, 2004). We note that, just as the KBA CCR task that we use as an exemplary document filtering setting throughout this paper, the adaptive filtering task assumes a scenario where topics evolve over time. The additional assumption made in adaptive filtering, however, is that explicit relevance feedback is available to a system immediately after prediction. This assumption is unrealistic in typical production environments, and it is not made in the KBA CCR setting.

Topic Detection and Tracking (TDT) is a closely related research area that is concerned with online monitoring of news streams, with a focus on event-based information organization (Allan, 2002). Specific tasks include story segmentation (identifying topically cohesive sections), topic tracking (monitoring events throughout time), topic detection (organizing news stories that discuss the same topic), first story detection (detecting new events), and link detection (finding out whether or not two stories discuss the same topic). All these tasks are cast as a detection problem and measured in terms of detection cost, which is a weighted sum of miss and false alarm probabilities. Of the above tasks, topic tracking is the most similar to the filtering scenario studied at TREC, even though there are important differences that can be attributed to the definition of a "topic"; we refer to Ault and Yang (2002) for a detailed comparison between the TREC and TDT tasks. An important outcome of this comparison is the finding that "neither T9P [utility] nor C_{trk} [detection cost] makes a good, general metric for information filtering" (Ault & Yang, 2002).

The TREC Knowledge Base Acceleration (KBA) track was launched in 2012 to study another instance of the filtering problem. The task, called *cumulative citation recommendation* (CCR), assumes a scenario with an editor as the end-user, who is responsible for maintaining the profile of a given entity, e.g., an article in a knowledge base, such as Wikipedia. In this setting, topics correspond to entities and "citation-worthy" documents are sought. As before, the corpus is a stream of documents (here, web documents and microblog posts) that are to be processed in chronological order, in particular, hourly batches. Evidently, the system must never use information from the "future" in any way. One key aspect of KBA CCR is that target topics (entities) evolve over time. Therefore, systems can be expected to display changes in performance during test time. For instance, static non-adaptive systems might degrade, as the entities they monitor evolve while they do not. Another point worthy of note is that the CCR task removes some of the unrealistic assumptions made in the TREC Filtering track, namely the assumption of instant feedback and assumption of "no batching of documents or ranking of small sets" (Robertson, 2002).

It is important to emphasize that our evaluation methodology is generic and can be applied to any document filtering task. Nevertheless, to make things more tangible, we will use CCR as a representative of the class of filtering tasks throughout this paper; CCR is the most realistic document filtering setting available so far, both in terms of assumptions made about the user and the underlying data (i.e., topics that evolve over time). See Frank, Bauer, et al. (2013); Frank, Kleiman-Weiner, et al. (2013) for further details on TREC KBA CCR.

2.2. Time-aware evaluation

While IR work that assumes a streaming setting is rapidly gaining attention, the topic has been considered for several decades (see, e.g., Robertson & Soboroff, 2002), with early work going back at least to Luhn (1958). Monitoring IR system performance over time appears to be a more recent interest (Azzopardi, 2009).

A setting very similar to the one addressed in this paper is discussed in Dietz et al. (2013), in which the authors consider time-aware evaluation in a document filtering setting. They are primarily concerned with problems associated with bursty streams (where more documents arrive during bursts). They propose to measure performance in time batches, but rather than taking the average performance over time across batches, they propose to use a weighted average, where every batch is weighted by its importance, which is relative to its burstiness. The important difference between their approach and ours is in this averaging step. By taking an average (whether it is weighted or not) the eventual metric is imperceptive to any increase or decrease in performance over time.

Trend lines have previously been applied to TREC results; see (Sanderson & Zobel (2005)). However, the focus of that article is on different statistical significance tests and on a comparison between them. Samples of various sizes are taken from past TREC runs and (non-linear) trends are fitted to the performance across sample size. The aim of the trend lines is to extrapolate to larger sample sizes so as to estimate hypothetical performance on bigger test sets. Performance across time plays no role in this work.

The metrics used for the recently introduced temporal summarization track of TREC also incorporate time-aware measures of performance (Aslam et al., 2013). In particular, (mean) expected gain and comprehensiveness are calculated from the onset of the evaluation period up to a certain number of seconds after it. The integral of each measure with respect to time is calculated to measure how quickly a system captures relevant information. The main difference with our approach is that we measure performance for every batch across the entire evaluation period, rather than only the first period. Additionally, rather than using the integral to measure how a system performed in a single time batch, we propose to use the derivative of a trend line to measure the *change* in performance across time.

Evaluation over time of text classification systems is important in the area of spam detection. E.g., in Keiser (2009) online learning classification algorithms are evaluated. Progressive performance is plotted as a function of the number of messages covered. As the messages are ordered by time, the coverage axis could be interpreted as a time axis. As such, the setting is nearly identical to the one we address. However, in Keiser (2009) the graphs plotting progressive performance through time are displayed for visual inspection only, and no attempt is made to come up with a metric that quantifies the change in performance over time.

2.3. Trend estimation

Trend estimation is commonplace in many scientific disciplines, ranging from econometrics to climate sciences.

Recent years have witnessed a number of papers where ideas from economy and/or econometrics have inspired algorithmic development in IR (Azzopardi, 2011; Wang & Zhu, 2009). In Bianchi, Boyle, and Hollingsworth (1999) different trend estimation methods are applied to economic data; the data used there is more volatile than ours and the focus lies on how fast a method convergences to a trend and on its ability to pick up turning points. In contrast, we assume that only one trend is present; this is motivated by the limited time span covered by the test sets we use and by the fact that no explicit feedback is available to the systems. Due to the latter aspect, it is unlikely that a change in direction of performance over time occurs: unless explicit relevance feedback is available to a system, it is implausible that, e.g., a decrease in performance is reversed and turned into an improvement.

In climatology data points are typically obtained from natural phenomena. Trends may occur but extreme observations may obscure them. Therefore, robustness to outliers is an issue. The most common way to fit a straight line to a set of points is by using least squares (LS). The advantage is that there is a closed form formula for calculating its parameters. However, "[t]he problem that a single outlying observation may suffice to severely influence the LS regression estimator makes the LS regression a non-robust method" (Muhlbauer, Spichtinger, & Lohmann, 2009). Two alternatives were proposed in Muhlbauer et al. (2009), the least median of squares estimator and the least trimmed squares estimator. Both take a subset of observations (the ones closest to the median, and the ones with the smallest residuals, respectively) and estimate regression parameters from the subset. Rather then using these alternatives in our tests, we use weighted least squares. We do so for the following reasons. Firstly, peaks in the data might actually indicate interesting underlying phenomena and should be investigated, rather than left out or smoothed out (Field, 2013). Secondly, the data sets we use (see Section 6) are sufficiently big, and the larger the sample the estimates are based on, the smaller the effect of outliers will be. Thirdly, as discussed in Section 6.3, the weighted least squares estimate, is more robust to outliers. Lastly, as the primary focus of this paper is to introduce our time-aware evaluation framework; an extensive study of when and how to apply alternative estimation methods is beyond the scope of this paper. Below, we assume that the data we work with is sufficiently consistent to warrant common least squares estimation.

2.4. Significance testing

Tests for comparing measurements over time require special care. Statistical methods for comparing regression coefficients between models, including the *z*-test we use (see Section 5.3), are discussed in Clogg, Petkova, and Haritou (1995). The scenario addressed there differs slightly from ours, as the main focus of the work in Clogg et al. (1995) is on models that are assumed to be extensions of one another.

Homoscedasticity, or similarity of variance across data points, is an important assumption of least squares regression. If it is violated by the use of heteroscedastic data, significance tests are not valid anymore, even if the regression parameters still are unbiased. In Section 5.4 homoscedasticity and its implications on our framework are discussed in more detail. In Hayes and Cai (2007), Long and Ervin (2000) the use of tests based on the heteroscedasticity-consistent covariance matrix is advocated. The heteroscedasticity-consistent standard error estimators derived from this matrix allow for a more robust method in dealing with heteroscadistic data when no assumptions about the source of heterogeneity of variance can be made. The authors introduce four estimators, HC0 up to HC3. They show by extensive tests on synthetic data that hypothesis tests using HC3 standard error estimators lead to the most robust results. In our tests in Section 6 we show results using HC3 standard error estimates as they lead to the most conservative estimates.

3. Measuring performance over time with trend estimation

To measure the performance of a document filtering system over time we propose to use trend estimation. The time period the system is evaluated on is divided into equal-sized batches. We calculate the performance in batches in terms of an *underlying metric*, such as macro-F1, and fit a straight line to the outcomes. The derivative of the fitted line, the slope coefficient, indicates the improvement or degradation of the system over time with respect to the underlying measure. In Fig. 1 the blue dots are the outcomes of the underlying metric per batch, and the orange lines are the fitted trend lines. While the slope coefficient can provide additional insights into system behavior, in itself it is not suitable as a performance metric. A larger coefficient indicates that one system is improving more over time than another. Yet, it can still fall behind considerably in terms of absolute performance. Therefore, we consider system performance at the end of the evaluation period and return the value assumed by the fitted trend line at the most recent point in time, as indicated with the solid orange dots in Fig. 1. The performance at this point in is what we call the *estimated end-point performance*.

In the remainder of the paper we will refer to the trend analysis based on batch evaluation as *trend estimation framework*. The estimated performance at the end of the evaluation period, which we will use to rank document filtering systems by, we will refer to as *time-aware evaluation metric*.

How should we assess the time-aware evaluation metric and the trend estimation framework it is built on? We answer this question in two ways. Below, we discuss the properties that we believe any reasonable document filtering evaluation metric should have. The trend estimation framework is based on two main components: (i) measuring (overall) system performance for a given period of time (referred to as a *time batch*) and (ii) computing estimated end-point performance by fitting a trend line. In Section 4 we discuss evaluation per batch along with the requirements on evaluation metrics that such a setting brings. The trend estimation framework is discussed in Section 5, where we check the properties, the requirements for per batch evaluation, together with statistical significance testing and the assumptions that need to hold for it.

Let us turn to the properties that we believe any reasonable metric used for evaluating IR systems should be have: stability, sensitivity, reliability, and unbiasedness. An additional requirement in the case of evaluating document filtering systems is time-awareness.

By *time-awareness* we mean the ability of a metric to take changes in performance over time into account. As performance of systems might evolve over time, consistent changes throughout the evaluation time period should be reflected by a metric. The metrics currently available perform averaging over the entire evaluation period, and by doing so disregard the temporal dimension.

Stability and *robustness* represent the same concept, under different names. “An evaluation methodology can be said to be stable with respect to some changing variable, or robust to changes in that variable” (Berendsen, de Rijke, Balog, Bogers, & van den Bosch, 2013). Hence, we will use the terms stability and robustness interchangeably. Stability can be measured by the extent to which a ranking of systems is affected by changes in the experimental setting, with respect to the evaluation measure, e.g., by varying the size of the test topic set (Buckley & Voorhees, 2000; Yilmaz & Aslam, 2006). Stability is examined based on a set of systems, e.g., all runs submitted to a TREC task. Therefore, it is not a property of a given evaluation metric, in isolation, but of a particular experimental design, including the test topics and the systems that are subject to comparison.

Sensitivity is the ability to detect changes in the quality of rankings (Hofmann, Whiteson, & de Rijke, 2011). Without sensitive measurement we might reject a small but significant improvement (Radlinski & Craswell, 2010).

Zhou, Lalmas, Sakai, Cummins, and Jose (2013) define *reliability* as the “ability of a metric to detect ‘actual’ performance differences as opposed to those observed by chance.” To help improve reliability of a metric, a large enough difference between scores of systems should be observed before concluding that they are different. In Section 5.3 we discuss what is needed to interpret the difference between two regression coefficients.

A sound metric should have no *bias*, i.e., no preference should come from the metric itself. Unbiasedness of the trend estimation framework means that a sufficiently large set of random scores of the underlying metric randomly distributed over time should produce a flat score. The chances of an exact zero trend being observed in random data are in fact negligible. For large sample sizes, however, the trend in random data will on average be statistically indiscernible from a zero trend at high confidence levels, due to the central limit theorem (Gravetter & Wallnau, 2007).

Our proposed method of measuring performance over time per batch and fitting a trend line to the results can be applied to any metric. It can be applied, e.g., to precision and recall separately, or to any other metric including click-based ones like ERR, if click data is available. As we are interested in our experiments in the overall performance of systems over time, we focus on *F*-measure in the remainder of this article. However, we note that what follows can be applied to any metric that can be calculated per batch.

4. Measuring performance per time batch

To capture performance over time we propose to divide the period a system is evaluated on in time batches of equal size. As discussed below, this introduces some caveats, in particular concerning false positives. We first analyze the requirements for per-batch evaluation and address measuring false positives afterwards.

4.1. Requirements

Fig. 2 shows a graphical overview of five relevance requirements that a per-batch metric has to meet.

The first three cover the standard cases where having more relevant documents is preferred over having less relevant documents (1 and 3) and returning irrelevant documents is punished (2), all else being equal. An overall average in terms of macro-precision and macro-recall, or macro-F_1 covers exactly these cases. Cases 2, 4 and 5 all pertain to the *Growing Quality Constraint* for document filtering metrics (Amigó et al., 2011) which asserts that "adding an irrelevant document to the output [...] must decrease the score.". Interestingly though, in 4 and 5, precision and recall (and hence F_1) will give identical score (namely 0). If we assume System A to be the ground truth in the fifth case, recall is not even well-defined (division by 0).

When taking an overall average (with all documents in a single batch) cases 4 and 5 do not constitute a problem, as these cases will simply never occur (it does not make sense to evaluate a system when there is no ground truth). In a setting with time batches, however, they do.

4.2. Measuring false positives

In a document filtering setting results may be returned for topics that were not listed in the ground truth for a particular batch. Also, batches may occur for which the ground truth mentions no relevant documents at all. For example, in a document filtering setting like TREC KBA CCR, where entities are being monitored over time, it could happen that during a certain day or week, no documents were published concerning any of the entities of interest. In these cases precision is not a meaningful measure, and recall and MAP are actually not defined, as the denominators in both cases are zero. Recall, e.g., is defined as the number of true positives, divided by the sum of true positives and false negatives. If there are no positive instances defined for a batch, there are neither true positives nor false negatives, so the denominator is 0. Hence recall is undefined.[1] Hence, by only looking at recall- and MAP-like metrics (a similar argument can be raised for nDCG, the new metrics introduced in Amigó et al. (2013), and many others) we cannot measure the performance of a system at times there are no true positives. A system might, however, keep returning results (false positives), all of which go unnoticed by the evaluation metrics. A metric should favor systems that return less false positives over ones that return more, even if no positive documents are available in the ground truth (cases 4 and 5 in Fig. 2). Therefore, false positives should be measured more directly when topic tracking or entity monitoring systems are being evaluated in batches over time.

We note that this fact was also recognised by the organisers of the TREC filtering track: "Classical set-based evaluation measures from information retrieval such as raw precision and recall do not behave gracefully for topics with few or no relevant documents. For example, the precision of a system which returns one non-relevant document is zero. The precision of a system which returns one thousand non-relevant documents is also zero. However, the former system is doing a far better job of filtering than the latter." (Hull et al., 1998).

There already are multiple metrics that measure false positives. In the next section we propose yet another way of doing so. In the remainder of this section we will briefly go over the existing metrics and explain why we do not use them. In what follows, we use FP to denote false positives, TP for true positives, FN for false negatives and TN for true negatives.

False discovery rate is defined as $\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TP}}$. As can easily be deduced from this formula, the metric does not measure anything if there are no TPs, as it is always 1 in those cases.

[1] This is not the same as having recall (or MAP) score of zero. A zero result represents a (rather bad) result, while in this case we have an absence, a lack of results. In the tests discussed in Section 6 this is simply handled by leaving out the values for these time batches, which does not affect the least squares fitting of the line.

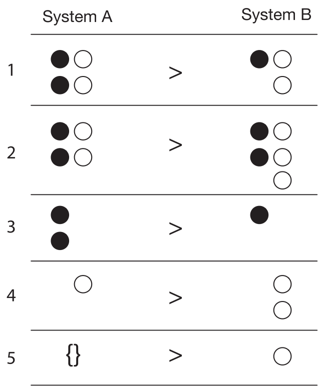

Fig. 2. Graphic display of relevance requirements. Opaque dots represent relevant results, transparent dots represent irrelevant results.

False positive rate is defined as $\frac{\mathrm{FP}}{\mathrm{FP+TN}}$. This is not a practical measure when there are no positive documents in a batch, as the TNs are in fact all documents in the batch that were not returned by the system. The size of the corpora today (the KBA stream corpus 2013, e.g., contains billions of documents) is such that the number of documents in a batch can easily be several orders of magnitude bigger than the number of document returned by a system. Hence, false positive rate will in practice most often be a very small number. In the case that a very small batch occurs, returning only a couple of FP documents for this batch will result in a very high false positive rate, whereas returning many more FPs for a larger batch could result in a much lower false positive rate. This effect is counterintuitive from a user perspective, as the user is oblivious to the size of a particular batch, but not to the amount of FPs presented by the system.

As noted above, the TREC filtering track also recognised the importance of measuring FPs and hence employed the utility measure that directly measures FPs. It measures TPs and subtracts the number of FPs, where both the number of TPs and FPs are multiplied by a factor indicating their respective importance. The score used in the last year the track ran is presented in Robertson and Soboroff (2002) as $T11U = 2\mathrm{TP} - \mathrm{FP}$. To be able to average across topics the score is normalized by its maximum value, which is 2TP. This latter step precludes us from using it, as, for true-positive-less batches, the score is undefined. A similar argument applies for the non-linear utility scores used in an earlier TREC document filtering track (see Hull & Robertson, 1999).

The evaluation measure used at TDT is Normalized Detection Cost, which is a weighted average of miss and false-alarm rates. Following Ault and Yang (2002), we refer to this metric as C_{trk}:

$$C_{trk} = \frac{C_{miss}\frac{\mathrm{FN}}{\mathrm{TP+FN}}P_{ontopic} + C_{fa}\frac{\mathrm{FP}}{\mathrm{FP+TN}}(1 - P_{ontopic})}{\min(C_{miss}P_{ontopic}, C_{fa}(1 - P_{ontopic}))},$$

where C_{miss} and C_{fa} are the costs for misses and false-alarms, respectively, and $P_{ontopic}$ is the a priori probability that a story is on-topic. The evaluation specification states the actual values; at TDT-3, these are $C_{miss} = 1.0, C_{fa} = 0.1$ and $P_{ontopic} = 0.02$. Overall system performance is the macro-average of the system's C_{trk} scores for the individual topics (events). Note that a lower C_{trk} is better, i.e., a perfect system has a 0 score, while a system that retrieves all documents or zero documents gets a C_{trk} score of 1.

The C_{trk} metric cannot be used to measure performance when no TPs occur, because in such an event, no FNs occur either, and hence the first term in the numerator is undefined (division by 0). We note that if this is solved (e.g., by assuming the first term is 0 in these cases) C_{trk} does satisfy all relevance requirements in Fig. 2. But, as "false-alarm rate is defined in terms of the number of non-relevant documents, most of which the user hopefully does not see" (Ault & Yang, 2002), it suffers from the same issue as false positive rate does: it is overly sensitive to small batches, where TN is low.

4.3. Aptness

We propose to measure the aptness of a system by quantifying the number of false positives it returns in the following way:

$$\text{aptness} = \frac{\zeta}{\zeta + \mathrm{FP}}. \tag{1}$$

Here, FP is the number of false positives and ζ is a sensitivity parameter. When ζ is set to a low value (e.g., 1), adding a few false positives already decreases the aptness score considerably. Choosing a higher value for ζ reduces this effect. In the results reported in Section 6 we set ζ to 1.

The output of the aptness score is in the (0,1] scale. The larger the number of false positives, the lower the score. Adding false positives when there are only few has a more drastic effect on the score than adding a few false positives when there are many already. We argue that this is a desired property of the metric in the scenarios considered in this paper.

The official TREC scoring strategy is to implement macro-averaging as summing scores from all entities and dividing by the number of entities (Frank, Bauer, et al., 2013). In the case of macro-precision and macro-recall this means averaging over the number of entities in the ground truth. For aptness this means averaging over the number of (unique) entities in the ground truth and the system run combined.

4.4. Combining scores

A standard way of combining scores is to weigh their contribution to the final score. For the *F*-measure, e.g., this can expressed as

$$\frac{1}{\frac{\alpha}{P} + \frac{1-\alpha}{R}},$$

where α is a parameter that indicates the relative importance of precision over recall. We propose a similar strategy for an overall score that incorporates *p*recision, *r*ecall and *a*ptness that we call F_{pra}:

$$F_{pra} = \frac{1}{\frac{\alpha_P}{P} + \frac{\alpha_R}{R} + \frac{\alpha_A}{A}}. \tag{2}$$

Here, P stands for precision, R for recall and A for aptness. The α_m parameter indicates the relative imprtance of measure m.

However, as a measure can be undefined at a particular time, we have:

$$F_{pra} = \frac{1}{\mathbf{1}(P)\cdot\frac{\alpha_P}{P} + \mathbf{1}(R)\cdot\frac{\alpha_R}{R} + \mathbf{1}(A)\cdot\frac{\alpha_A}{A}},$$

where $\mathbf{1}(m) = 1$ for a measure m if it is defined and 0 otherwise. We assume that $\mathbf{1}(x)\cdot\frac{\alpha_x}{x} = 0$ when x is undefined. We want that $\sum_{m\in[P,R,A]}\alpha_m = 1$ and hence choose $\alpha_m = \frac{1}{\sum_{m\in[P,R,A]}\mathbf{1}(m)}$.

In practice, this means that if all metrics are defined, the score is the harmonic mean of their values. If precision and recall are not defined, the score reduces to the aptness score, which is always defined. Note that as a result, if there is no ground truth for a certain batch, and a system produces no output for this batch, it gets a maximum score of 1, which is a desired effect.

As we can see from Eq. (2), cases 4 and 5 in Fig. 2 are handled correctly by F_{pra} as adding false positives will always reduce aptness, and hence F_{pra}.

For batches for which ground truth documents are available, false positives are taken into account both in the precision and in the aptness score. Just as precision and recall measure two distinct things about true positives, aptness and precision measure two distinct things about false positives. Precision, by considering the number of false positives in relation to the number of true positives, measures how pure results are. Aptness, by only considering false positives, measures how much noise a system produces. The F_{pra} score, by calculating the harmonic mean between the three scores, implicitly implements an underlying user model where the user is interested in three things: how good is the system in finding relevant documents (precision), how many relevant documents does it miss (recall) and how much noise does the system produce (aptness).

To distinguish between different *F*-measures, we use F_{pr} to refer to the batch-based *F*-measure which considers only precision and recall, while F_{pra} is used for the batch-based *F*-measure that considers precision, recall and aptness, as described above. The traditional *F*-measure, averaged over the entire evaluation period, is denoted as F_1.

5. Trend estimation

In this section we provide the details of how we propose to employ trend estimation for evaluation.

5.1. Weighted least squares

As performance is measured per batch, the amount of data a metric is calculated from can differ across time batches. Therefore, rather than using ordinary least squares to fit a line to the measurements in different batches, we should use a weighted least squares estimator. The weight for a time batch is the amount of data the metric is calculated from in this batch, as a proportion of the total amount of data available to the metric over the entire time period. If the entire evaluation period ranges form $t = 0$ until $t = T$, the weight w for F_{pr} or F_{pra}, for a time batch ranging from t_i to t_{i+1} would be:

$$w_{t=t_i}^{t_{i+1}} = \frac{\#documents\ in\ run\ and\ ground\ truth\ combined_{t=t_i}^{t_{i+1}}}{\#documents\ in\ run\ and\ ground\ truth\ combined_{t=0}^{T}}.$$

5.2. Normalizing trends

The size of the batches is a parameter of the trend estimation model. We refer to this as the *time granularity*. If two trends have been estimated on the same data but with different granularity they need to be normalized before they can be compared. If, e.g., one trend is estimated on 1-h batches, while the other is based on 24-h batches, the slope coefficients will not correspond directly. This is simply due to a difference in units of measurement. The first slope coefficient represents the change per hour where the second one represents the change per day, which, if the two trends are in fact the same, is 24 times as much. To compare the two, the two trends can be normalized to the same time unit, e.g., hours or seconds. See Section 6.4 for additional discussion.

5.3. Statistical significance

With respect to statistical significance, there are two questions of interest about the slope coefficients of regression lines. One is whether the observed trend is a trend at all (i.e., if it is distinguishable from a zero trend); this is answered using a *t*-test (Field, 2013). The other question is whether the slope coefficients of two fitted lines are significantly different; we employ a *z*-test to answer this (Paternoster, Brame, Mazerolle, & Piquero, 1998; Clogg et al., 1995).

5.3.1. Testing for a trend

When testing whether an estimated slope coefficient $\hat{\beta}$ is different form a zero trend β_0, the *t*-statistic is given by $t = (\hat{\beta} - \beta_0)/SE(\hat{\beta})$ (Field, 2013), which, as β_0 is 0, reduces to:

$$t = \frac{\hat{\beta}}{SE(\hat{\beta})}, \tag{3}$$

where $SE(\cdot)$ is the standard error of the residuals.[2] The degrees of freedom are the number of observations minus the number of regressors including the intercept (so 2 in our case).

5.3.2. Testing between trends

The statistical significance between two trends is determined by a *z*-test (Paternoster et al., 1998):

$$z = \frac{\hat{\beta}_1 - \hat{\beta}_2}{\sqrt{SE(\hat{\beta}_1)^2 + SE(\hat{\beta}_2)^2}}, \tag{4}$$

where $\hat{\beta}_1$ and $\hat{\beta}_2$ are the slope coefficients of the fitted lines.

As discussed in Section 2 there are multiple ways to calculate the standard error of the slope coefficient $SE(\cdot)$. We show results for HC3 standard error estimates for our tests in Section 6 and do not use conventional standard error estimates.

For the tests just presented to be applicable, a number of conditions have to be met as discussed next.

5.4. Assumptions

For statistical tests on slope coefficients of fitted lines (as detailed in Section 5.3) to be valid, assumptions need to hold concerning linearity, normality, homoscedasticity and independence (Field, 2013). In this section we briefly discuss the assumptions in our setting of filtering documents over time.

5.4.1. Linearity

The linearity assumption simply means that there is a relation between the variables we observe and that this relation is linear. If this assumption does not hold any interpretation of outcomes or statistical significance is meaningless as the model is not right. In the present case for the linearity assumption to hold, the dependent variable, performance, has to have a linear relation to the independent variable, time. Note that this is not a given fact: performance might, e.g., degrade exponentially.

A standard way of testing for a linear relation between two stochastic variables is to calculate Spearman's correlation coefficient. We should note, however, that while values of the coefficient towards the extreme ends of its scale do indicate a correlation, they do not, in fact, tell anything about it being linear or not (Anscombe, 1973). Visual inspection of results and background knowledge about the underlying systems producing them remain invaluable.

In this work, we assume that a linear relation holds between time and performance, and show how the trend analysis framework works if it does. We note that this is a simplifying assumption and it might well be that a trend is occurring that is non-linear. We argue, however, that it is reasonable to assume that, given the short time interval during which systems are being monitored, changes in performance (if observed at all) are relatively constant, and that, therefore, it is reasonable to assume that a linear fit to the performance over time.

[2] The slope coefficients are usually referred to by $\hat{\beta}$ rather than just β to indicate that they are estimates of a (assumed) true underlying function.

5.4.2. *Normality*

For the normality assumption to hold the residuals (from the fitted line) need to be normally distributed. For the purpose of estimating the regression line this assumption is barely important (Gelman, 2007). The normality assumption is of concern, however, for the statistical tests to be valid especially with small sample sizes.

We should note that as the metrics we use as data points are usually on a closed scale (with values typically between 0 and 1) the distribution of residuals can in fact never be truly normal. This is not a problem as long as the distribution is still "sufficiently normal." If, however, *only* score values 0 or 1 occur, the variable is actually categorical, and the assumption does not hold.

Normality can be tested in numerous ways. As reported in Razali and Wah (2011), the Shapiro–Wilk test and Anderson–Darling test are the most powerful tests for normality, and they are comparable in performance. We use the latter in our tests in Section 6.

5.4.3. *Homoscedasticity*

Homoscedasticity concerns the homogeneity of variance. The homoscedasticity assumption is violated when the variance across data points varies too much. Although the parameters estimated by the least squares method are unbiased if the assumption does not hold, the assumption needs to hold for statistical significance tests to be valid (Hayes & Cai, 2007). The residuals of the fitted line can be normally distributed, but if they are heteroscedastic, the least squares estimator can be inefficient, and the confidence intervals for the regression parameters can be inaccurate (Wilcox, 2012). This can, for example, be the case when there are far more measures for one system than for another system it is compared to. Also, in our case where observations are collected over time, homoscedasticity is related to homogeneity of observations over time. If we take the average per time span of several observations this can lead to heteroscedasticity, as averages will be taken over different numbers of observations, and hence different variances or uncertainties over time.

A solution for dealing with heteroscedastic data is to weight each data point by its variance and apply weighted least squares estimation to fit the regression line, rather than ordinary least squares (Field, 2013).[3] However, this method also builds on certain assumptions, in particular about the nature of heteroscedasticity in the residuals. As discussed in Section 2, a more robust method is to use heteroscedasticity-consistent standard error estimators. In results reported in Section 6, we use HC3 standard error estimators to compute the standard errors following (Hayes & Cai, 2007; Long & Ervin, 2000).

5.4.4. *Independence*

The independence assumption holds if the errors from the fitted line (the residuals) are independent of one another. Independence of residuals can be tested for by the Durbin–Watson test (Durbin & Watson, 1951). The values of the statistic range between 0 and 4. A value of 2 indicates no correlation. The test statistic is compared to a lower limit and an upper limit, which depend on the number of variables of the model (2 in our case), the number of observations and a *p*-value. If the test value is below the lower limit there is evidence of positive autocorrelation. If the value is higher than the upper limit there is no evidence. For values in between, the test is inconclusive. A rule of thumb is that values close to 2 are preferred, while values below 1 or above 3 are problematic (Field, 2013).

6. Results

We analyze the runs submitted to the KBA CCR tasks of TREC 2012 and TREC 2013 (KBA'12 and KBA'13 for short) with the time-aware evaluation metric as described above. In particular we are interested in answering the following research questions:

1. Do rankings of systems change when we evaluate document filtering systems with our time-aware evaluation metric, compared to using a time-agnostic method?
2. Does including aptness in F_{pra} make a difference compared to using F_{pr}?
3. Does our time-aware evaluation metric have all required properties as specified in Section 4?
4. Can claims about statistical significance be made using our time-aware evaluation metric?

6.1. *Experimental setup*

Every document returned in a KBA CCR run has a score that indicates its relevance to the topic it is associated with. The official TREC evaluation method performs a sweep over a range of possible values to find the optimal cutoff value for every run. As we evaluate the runs in batches and employ trend estimation, the optimal cutoff value for the new evaluation can differ from the ones reported in the official TREC results. Hence, we perform a cutoff sweep as well (from 50 up to and including 1000 in steps of 50) to find the optimal cutoff values in terms of estimated end point performance for both F_{pr} and for F_{pra}. In the tests below we report only on central/vital documents and we use 1-day time batches unless stated otherwise.

[3] We use weighted least squares and weigh data points by the amount of data their values are based on rather than their variance. In the case of, e.g., simple non-averaged precision or recall in one time batch, there is no such thing as variance, as no average is being computed.

6.2. Time-aware evaluation

The official TREC results provide a ranking of teams based on their best performing run in terms of macro-averaged F_1 (computed over the entire testing period). To answer research questions 1 and 2 we present rankings of the same runs based on the time-aware evaluation metric. For the first ranking we use the same macro-F_1 score that the official TREC results are based on as the underlying metric, computed as the harmonic mean of macro-precision and macro-recall. However, we compute it per batch, and refer to it as macro-F_{pr}. For the second new ranking, we use macro-F_{pra}, averaged in the same fashion, and also calculated per batch. Table 1 lists the rank correlation coefficients between the official TREC results the newly produced rankings, based on the time-aware metrics.

6.2.1. Using F_{pr}

Fig. 3 and 4 show the official results in terms of macro-averaged F_1-score for the runs submitted to KBA'12 and KBA'13,[4] respectively (Frank, Kleiman-Weiner, et al., 2013; Frank, Bauer, et al., 2013). Figs. 5 and 6 show the corresponding plots using estimated end-point macro-F_{pr} performance. Note that for easy reference and comparison we adhere to the respective styles used in the original TREC publications for KBA'12 and KBA'13 plots.

There is a number of interesting observations to make from Figs. 5 and 6. First, for both years, the ranking of runs has changed considerably compared to the "official" plots, which is also reflected by a Kendall's τ figure of 0.49 and 0.67. Most notably in 2012, the hltcoe team, which was ranked second in the official results, scores considerably lower, while the UvA and CWI teams are promoted upwards. The official results for 2013 show a plateau of top achievers with the runs of the BIT team elevated slightly above the plateau. The new ranking in Fig. 6 shows a different picture. The plateau is gone, and uiucGSLIS's best run is promoted to the first position while previously their runs were ranked close to the middle. The UMass_CIIR best-performing run was ranked second; it ends up in the middle region in the new ranking.

Second, for both years, the scores per team show a bigger spread than in the official graphs for some teams. An interesting case in 2012 is, e.g., the UvA. Where all the runs of this team were lumped together in the original graph, we see a spread of scores when ranked by the new metrics. This shows that the different approaches the teams implemented have a more distinct effect than one would be led to conclude from the official result graphs. From these findings we conclude that evaluating the TREC KBA CCR runs in a time-aware fashion leads to a considerably different insights and analysis than the time-agnostic evaluation does.

Third, with our time-aware evaluation method, the absolute scores go up to a higher value, in particular in 2013. This is a result of using macro-averaging per batch using F_{pr}, where false positives returned for entities that were not listed in the ground truth for that batch are ignored by F_{pr} (see Section 4.2). As a result, the macro-F_{pr} scores for some systems are considerably higher for certain batches, which is reflected in the graphs.

6.2.2. Using F_{pra}

The overly positive view that Figs. 5 and 6 might present, because of the disregarded false positives, is corrected for when we use the estimated end-point macro-F_{pra} score for evaluation. Figs. 7 and 8 display the new rankings produced accordingly for KBA'12 and KBA'13.

As expected, the absolute scores have dropped considerably compared to the ones in Figs. 5 and 6 (from max. ~ 0.5 to max. ~ 0.4, and max. ~ 0.8 to max. ~ 0.6 for 2012 and 2013, respectively). This shows that including aptness scores in the metric has a noticeable effect, which is also reflected by the rank correlation scores between rankings based on F_{pr} and F_{pra} as reported in Table 1.

In 2012 the ranking at the top has changed again compared to the rankings based on macro-F_{pr}. We still see the same spread of scores we observed in Fig. 5.

In 2013 the most striking observation is that at the end of the evaluation period, no run is doing better than the oracle baseline, neither in terms of estimated end-point macro-F_{pr} nor in terms of estimated end-point macro-F_{pra}. This has several implications. The oracle run uses a hand-picked set of surface form names. It assigns a "vital" rating to every document that matches a surface form of an entity and assigns a confidence score based on the length of the observed name (Frank, Bauer, et al., 2013). Apparently, having a very well-tuned set of surface form names per entity is crucial. Adaptivity, however, is important. Of the participating teams in 2013, four teams submitted runs for systems that were sensitive to changes at run time — uiucGSLIS (Efron, Willis, Organisciak, Balsamo, & Lucic, 2013), BIT (Wang, Song, Lin, & Liao, 2013), UAmsterdam (Kenter, 2013) and IRIT (Abbes, Pinel-Sauvagnat, Hernandez, & Boughanem, 2013). If the teams are ranked by the time-agnostic official TREC score, Fig. 4, there seems to be a plateau of 'best teams' (uiucGSLIS, PRIS, UDel (Liu & Fang, 2013), UMass (Dietz & Dalton, 2013) and BIT). If we look at the overview as produced by the time-aware F_{pra} metric, Fig. 8, we see that the only top achievers are BIT and uiucGSLIS, while the performance of the other teams is lower. Interestingly, these are the only two teams of the five high achievers as measured by the official TREC measure, that have time-aware systems. This shows that adaptivity turns out to be of key importance for performance over time (i.e., these systems do best at the end of the evaluation period). This distinction is lost when the original metric is used.

[4] The graph of the official KBA'13 results lists some additional runs that were submitted for an additional task called KBX. As this task was part of a slot filling experiment (some runs even were in the slot filling run format), and as such actually closer to the KBA SSF task, we omit these runs in our overview.

Table 1
Kendall's τ rank correlation coefficients between different rankings.

	TREC official vs. F_{pr}	TREC official vs. F_{pra}	F_{pr} vs. F_{pra}
2012	0.49	0.49	0.85
2013	0.67	0.64	0.67

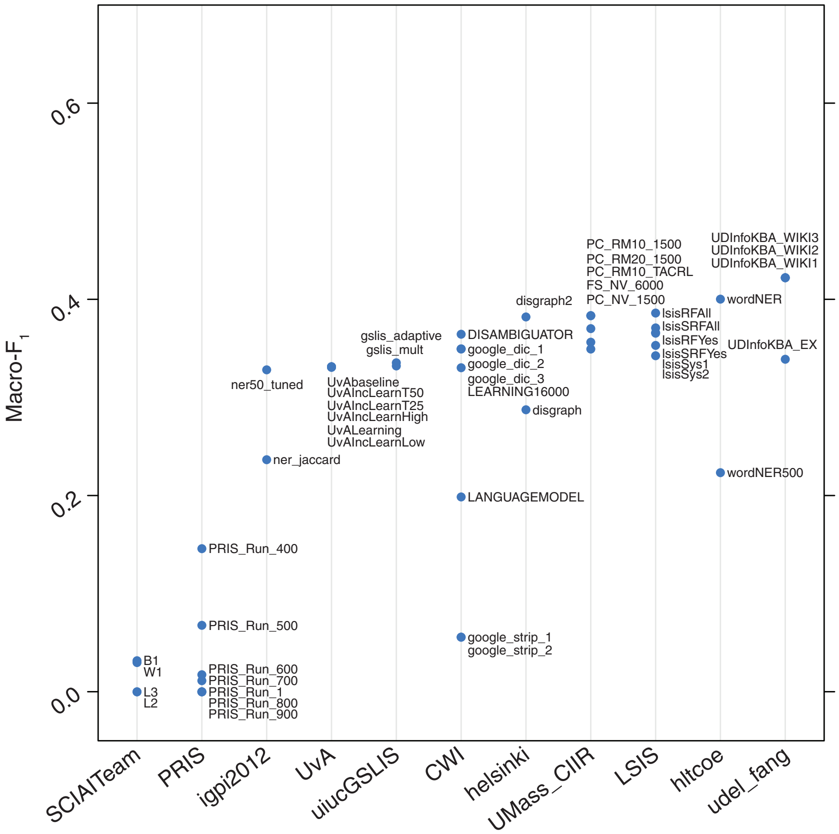

Fig. 3. Official KBA'12 results in terms of macro-F_1. Graph copied verbatim from Frank, Kleiman-Weiner, et al. (2013), except for the label on the *y*-axis.

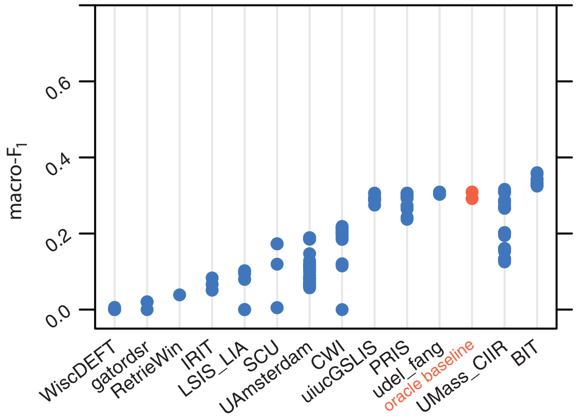

Fig. 4. Official KBA'13 results in terms of macro-F_1. Graph copied verbatim from Frank, Bauer, et al. (2013), except for the label on the *y*-axis.

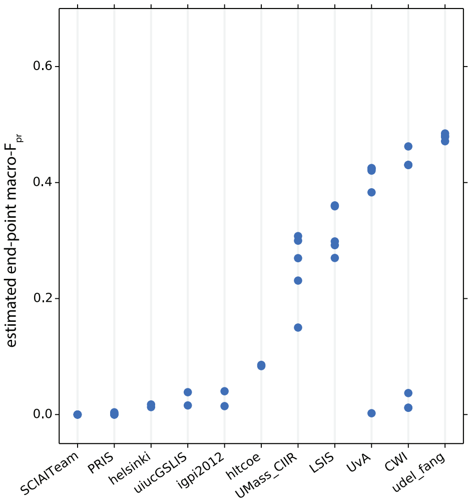

Fig. 5. KBA'12 results in terms of estimated end-point F_{pr}.

6.3. Analysis of properties

To answer research question 3 we analyze the properties and requirements discussed in Section 4.1 in the context of our proposed method. It is important to distinguish between the underlying metric that the trend estimation framework incorporates (F_{pr} or F_{pra}), and the slope coefficient it produces. The properties as discussed in this section pertain to trend estimation and the slope coefficient, not to the underlying metric.

6.3.1. Time-awareness

The time-aware evaluation metric we propose incorporates the temporal dimension of the document filtering task in two ways. Firstly, the slope coefficients estimated by trend analysis quantify how much a system is improving/degrading over time. Secondly, the estimated end-point performance values express the estimated performance of a system at the end of the evaluation period, taking changes in performance over time into account.

6.3.2. Stability

Stability can be measured by the extent to which a ranking of systems is affected by changes in the experimental setting. The trend estimation model as proposed in this paper has one parameter, which is the time granularity. Robustness with respect to this parameter would mean that changes in time granularity should not affect the fitted line or its slope coefficient. To test the stability of the trend estimation framework we compare the trends of all KBA'12 and KBA'13 runs across different levels of granularity for macro-precision, -recall, -aptness, -F_{pr} and -F_{pra} for 20 different cut-off levels. We normalize the slope coefficients to second level and compute the difference for every run between the trend estimated at a granularity of 1-day level and 7-day level and 30-day level respectively. Table 2 shows the statistics for the comparisons. We observe

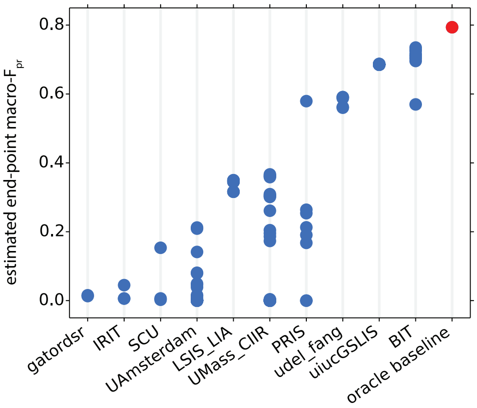

Fig. 6. KBA'13 results in terms of estimated end-point F_{pr}.

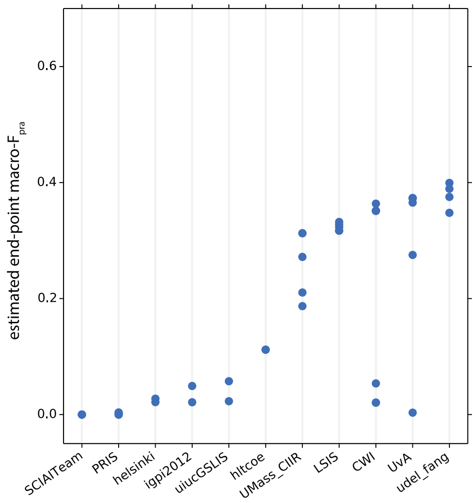

Fig. 7. KBA'12 results in terms of estimated end-point F_{pra}.

that the differences and deviations from the mean value are very close to 0, and conclude that the slope coefficients are stable under different levels of time granularity in all cases.

6.3.3. Sensitivity

Sensitivity is the ability to detect changes in the quality of rankings (Radlinski & Craswell, 2010; Hofmann et al., 2011). In the present setting this means that the trend estimation framework, and the slope coefficients it produces, should be sensitive to meaningful differences between systems. The significance tests in Section 5.3 describe how differences between rankings should be calculated.

As discussed in Section 2 the ordinary least squares method is sensitive to outliers, especially if they appear at the extreme ends of the scale (i.e., the start or end of the time period under consideration). In this paper we apply weighted least squares to fit a straight line to the data (see Section 5.1). This is very similar to the ordinary least squares method, only we weight the data points by the amount of data they are based on.

There is a caveat when for a batch both the ground truth and the results a system returned are empty. As described above, the F_{pra} score will be 1 in this case, but the weight for the score, as used for the weighted least square fit, is 0, as no documents are observed. This can have a negative effect on the fitted trend lines.

We should note that weighting data points as we propose in itself makes the method more robust to outliers. There is a risk that in a time batch with few observations, extreme values might cause a substantial deviation. However, as only few observations are involved, the weight of this data point will be low. And conversely, for a data point to be weighted heavily and have a big impact on the slope coefficient of the fitted line, it has to be based on a considerable number of observations, and hence it is not an outlier.

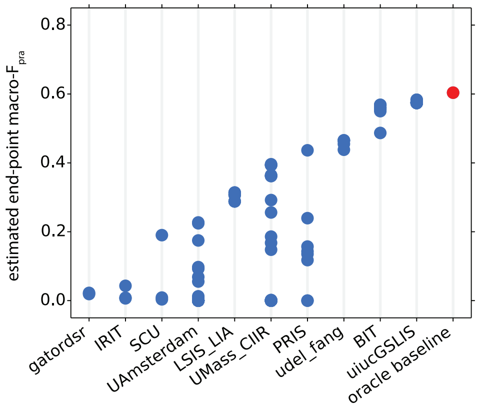

Fig. 8. KBA'13 performance in terms of estimated end-point F_{pra}.

Table 2
Comparison of differences between normalized trends of macro-averaged metrics (precision, recall, aptness, F_{pr} and F_{pra}) between 1-day granularity level and 7-day and 30-day granularity levels.

Measure	Mean	Std. dev.	Min	Max
KBA'12 – 7-day				
pr	1.649e−10	1.401e−08	−7.254e−08	2.494e−07
rc	9.779e−10	1.881e−09	−4.163e−09	1.044e−08
apt	4.704e−09	5.762e−08	−2.139e−07	5.637e−07
F_{pr}	6.533e−10	2.973e−09	−1.016e−08	9.951e−09
F_{pra}	−2.230e−10	3.110e−09	−1.218e−08	1.234e−08
KBA'12 – 30-day				
pr	−5.245e−10	1.329e−08	−1.142e−07	1.546e−07
rc	2.263e−09	3.645e−09	−7.378e−09	1.610e−08
apt	1.226e−08	1.377e−07	−2.066e−07	1.370e−06
F_{pr}	8.503e−10	3.821e−09	−1.619e−08	1.135e−08
F_{pra}	1.066e−10	5.520e−09	−1.978e−08	1.260e−08
KBA'13 – 7-day				
pr	1.855e−09	1.486e−08	−1.035e−07	1.889e−07
rc	−5.774e−10	1.117e−09	−4.406e−09	3.069e−09
apt	−5.847e−10	3.999e−09	−5.892e−08	2.496e−08
F_{pr}	−3.870e−10	1.296e−09	−5.131e−09	4.467e−09
F_{pra}	−6.222e−10	1.165e−09	−5.386e−09	2.357e−09
KBA'13 – 30-day				
pr	1.777e−09	1.486e−08	−1.036e−07	1.888e−07
rc	−9.259e−10	1.761e−09	−6.683e−09	3.448e−09
apt	5.687e−10	1.139e−08	−8.532e−08	4.333e−07
F_{pr}	−4.304e−10	1.867e−09	−7.798e−09	5.506e−09
F_{pra}	−7.954e−10	1.853e−09	−8.424e−09	4.338e−09

6.3.4. Reliability

In the context of trend estimation, reliability means that random changes observed over time should not be mistaken for actual changes in performance over time. Any noise introduced by random chance is accounted for by the least squares algorithm by design, as it models a line by $y = b + Wx + \in$, where $\in$ represents the estimated errors. If the estimated errors account for all the variance in that data, no trend will in fact be observed. Furthermore, if only random noise is observed, this results in a relatively flat line being fitted by the weighted least squares method, which would be statistically indiscernible from a zero trend (see our discussion on bias in Section 3).

6.4. Applicability

To answer research question 4 it is important to see if the assumptions needed for the statistical tests hold in a majority of cases. As described in Section 5.4, the normality assumption and the assumption of independence of errors are particularly important. To investigate if these assumptions hold for large numbers of runs regardless of the choice of metric or cutoff level, we analyze all KBA'12 and KBA'13 runs in terms of macro-precision, -recall, -aptness, -F_{pr} and -F_{pra}, at every cutoff level and at three different levels of time granularity: 1-day level, 7-day level and 30-day level. In 2012 there were 45 runs in total, which gives us 4500 results (45 runs · 20 cut-off values · 5 metrics) at every level of granularity. In 2013 there were 112 runs in total, leading to 11,200 results at every granularity level. Table 3 provides an overview of the number of cases in which a particular assumption was met for both years. A large portion of runs do not produce any results at lower cut-off values. The Durbin–Watson assumption trivially never holds in these cases. As this is not informative, we leave these runs out in Table 3 and only report on non-zero runs. The numbers for KBA'12 are lower, which might be caused by a larger number of poor-performing runs (i.e., a lot of very low, and hence very similar scores).

We see a consistent pattern, where at the finest level of granularity (1-day batches) the assumptions hold in most cases. An important implication of this finding is that when dealing with results as submitted to an evaluation campaign widely used in the IR community such as TREC KBA, the trend estimation framework as proposed here can be employed to make claims about significance of results in a vast majority of cases.

6.5. Analysis of systems

To see whether a particular choice of algorithms or parameter settings makes a substantial change in performance of a system, we can perform significance tests as described in Section 5.3. As an example, the BIT team that submitted runs to KBA'13 might be interested, based on the results displayed in Fig. 8, to know whether there is a significant difference between their best and worst run. Fig. 9 shows the results of the two runs over the entire evaluation period analyzed in 24 h batches. The relevant statistics are listed in Table 4. As we can see from this table, both the normality assumption

Table 3
Validity of Anderson–Darling and Durbin–Watson assumptions for multiple cut-off levels, at different levels of granularity.

Granularity	# Non-0 runs	A–D	D–W	A–D & D–W
KBA'12				
30-day	3758	1211	2916	1119 (30%)
7-day	3758	1291	3527	1246 (33%)
1-day	3774	2928	3595	2796 (74%)
KBA'13				
30-day	8537	1706	7637	1645 (19%)
7-day	8783	6881	7034	5498 (63%)
1-day	8715	8706	7752	7743 (89%)

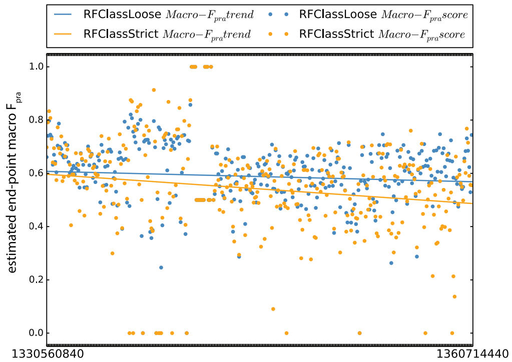

Fig. 9. Visual comparison of two runs submitted by the BIT team to KBA'13, in terms of estimated end-point Macro-F_{pra}.

Table 4
Overview of relevant statistics for two runs of the BIT team as submitted to KBA'13, based on macro-F_{pra} trend estimation. HC3 s.e.: HC3 standard errors, A–D: Anderson–Darling test statistic, D–W: Durbin–Watson test statistic. The last column shows the t-test statistic and p-value for a H_0 hypothesis that the trend line is equal to 0 (see Eq. (3)).

Run	Slope	HC3 s.e.	A–D (p-val)	D–W	t (p-val)
RFClassLoose	−1.11e−4	4.87–05	5.93 (1e−14)	1.58	2.29 (0.2)
RFClassStrict	−3.14e−4	7.03–05	5.81 (2e−14)	1.66	4.47 (0.05)

and the independence assumption hold. While the runs start out roughly similar, RFClassStrict's performance (orange[5] line in Fig. 9) drops while the performance of RFClassLoose (blue line in Fig. 9) stays more or less the same. In fact, we can see from Table 4 that the slight trend in the results for RFClassLoose is significantly different from a zero-trend, while the more pronounced trend in the RFCClassStrict results is, at a p-value of 0.05. To see if the two trends differ significantly from one another, we use the numbers from Table 4 in Eq. (4). We get a z-statistic of 2.37 which leads us to conclude that the two trends are indeed different, at a p-value of 0.02, for a two-tailed test.

7. Conclusions

In this paper we have proposed to measure the performance of document filtering systems in a time-aware manner. The underlying idea behind our approach is to measure overall system performance in time batches, fit a trend line to the results, and consider the value assumed by the fitted trend line at the end of the evaluation period (referred to as estimated end-point performance) as the time-aware evaluation metric.

Our analysis of requirements has revealed that existing measures fail to correctly deal with batches for which the ground truth mentions no relevant documents at all. We have introduced the concept of aptness and incorporated it into a new variant of the F-measure, F_{pra}. We have evaluated the runs submitted to TREC KBA CCR in 2012 and 2013 using the trend estimation framework and found that the order of top-performing systems changes. Perhaps the most remarkable and important finding is that from the official TREC 2013 KBA CCR results, based on time-agnostic metrics, it would seem that the oracle baseline run was beaten by several systems. However, we show by re-evaluation using our proposed time-aware metrics that in fact none of the systems is able to outperform the oracle run. As the oracle baseline is static and does not

[5] For interpretation of color in Fig. 9, the reader is referred to the web version of this article.

attempt to learn during runtime, this has strong implications concerning (the need for) adaptivity strategies for this task and corpus.

Additionally, we have shown that the assumptions needed for statistical significance tests with the new method hold in a vast majority of the cases for the TREC KBA CCR runs for both years in which the track ran.

Our experiments have focused exclusively on the TREC KBA CCR task, as a representative of a document filtering problem. The proposed trend estimation framework and time-aware evaluation metric, however, are not limited to the task of document filtering, but can also be applied to other retrieval tasks, where a more or less monotonic increase or decrease in performance of the retrieval system over time can reasonably be assumed. Dealing with non-linear trends is left for future work.

Acknowledgements

We are grateful to John R. Frank and Ellen Voorhees of NIST for providing us with essential data.

This research was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 288024 (LiMoSINe) and nr 312827 (VOX-Pol), the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, the Center for Creation, Content and Technology (CCCT), the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project number 027.012.105, the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.

References

Abbes, R., Pinel-Sauvagnat, K., Hernandez, N., & Boughanem, M. (2013). Irit at trec knowledge base acceleration 2013: Cumulative citation recommendation task. In *Proceedings of the Twenty-Second Text Retrieval Conference (TREC 2013)*.

Allan, J. (2002). *Topic detection and tracking: Event-based information organization*. Kluwer Academic Publishers..

Amigó, E., Gonzalo, J., & Verdejo, F. (2011). A comparison of evaluation metrics for document filtering. In *ECIR '11* (pp. 38–49).

Amigó, E., Gonzalo, J., & Verdejo, F. (2013). A general evaluation measure for document organization tasks. In *SIGIR '13* (pp. 643–652).

Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician, 27*(1), 17–21.

Aslam, J., Ekstrand-Abueg, M., Pavlu, V., Diaz, F., & Sakai, T. (2013). Trec 2013 temporal summarization. In *Proceedings of the Twenty-Second Text Retrieval Conference (TREC 2013)*.

Ault, T. G., & Yang, Y. (2002). Information filtering in trec-9 and tdt-3: A comparative analysis. *Information Retrieval, 5*(2–3), 159–187.

Azzopardi, L. (2009). Usage based effectiveness measures: Monitoring application performance in information retrieval. In *CIKM '09* (pp. 631–640).

Azzopardi, L. (2011). The economics in interactive information retrieval. In *SIGIR '11* (pp. 15–24).

Berendsen, R., de Rijke, M., Balog, K., Bogers, T., & van den Bosch, A. (2013). On the assessment of expertise profiles. *Journal of the Association for Information Science and Technology, 64*(10), 2024–2044.

Bianchi, M., Boyle, M., & Hollingsworth, D. (1999). A comparison of methods for trend estimation. *Applied Economics Letters, 6*(2), 103–109.

Buckley, C., & Voorhees, E.M. (2000). Evaluating evaluation measure stability. In *SIGIR '00* (pp. 33–40).

Clogg, C. C., Petkova, E., & Haritou, A. (1995). Statistical methods for comparing regression coefficients between models. *American Journal of Sociology*, 1261–1293.

Dietz, L., & Dalton, J. (2013). Umass at trec 2013 knowledge base acceleration track. In *Proceedings of the Twenty-Second Text Retrieval Conference (TREC 2013)*.

Dietz, L., Dalton, J., & Balog, K. (2013). Time-aware evaluation of cumulative citation recommendation systems. In *Proceedings of the SIGIR 2013 workshop on time-aware information access*.

Durbin, J., & Watson, G. S. (1951). Testing for serial correlation in least squares regression. ii. *Biometrika, 38*(1/2), 159–177.

Efron, M., Willis, C., Organisciak, P., Balsamo, B., & Lucic, A. (2013). The university of illinois graduate school of library and information science at trec 2013. In *Proceedings of the twenty-second text retrieval conference (TREC 2013)*.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. Sage.

Frank, J. R., Kleiman-Weiner, M., Roberts, D. A., Niu, F., Zhang, C., & Ré, C., et al. (2013). Building an entity-centric stream filtering test collection for TREC 2012. In *TREC '12*.

Frank, J. R., Bauer, S. J., Kleiman-Weiner, M., Roberts, D. A., Tripuraneni, N., Zhang, C., et al. (2013). Evaluating stream filtering for entity profile updates for. *TREC 2013 working notes*. NIST November 2013.

Gelman, A. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.

Gravetter, F. J., & Wallnau, L. B. (2007). Statistics for the behavioral sciences. CengageBrain. com.

Harman, D. (1994). Overview of the third text retrieval conference (TREC-3). In *Proceedings of the third text REtrieval conference (TREC-3)*. NIST.

Hayes, A. F., & Cai, L. (2007). Using heteroskedasticity-consistent standard error estimators in ols regression: An introduction and software implementation. *Behavior Research Methods, 39*(4), 709–722.

Hofmann, K., Whiteson, S., & de Rijke, M. (2011). A probabilistic method for inferring preferences from clicks. In *CIKM '11* (pp. 249–258).

Hull, D. A., & Robertson, S. E. (1999). The TREC-8 filtering track final report. In *TREC*.

Hull, D. A., et al. (1998). The trec-7 filtering track: description and analysis. NIST SPECIAL PUBLICATION SP (pp. 45–68).

Keiser, V. L. (2009). Evaluating online text classification algorithms for email prediction in TaskTracer. In *Conference on Email and Anti-Spam*.

Kenter, T. (2013). Filtering documents over time for evolving topics – the university of amsterdam at trec 2013 kba ccr. In *Proceedings of the twenty-second text retrieval conference (TREC 2013)*.

Liu, X., & Fang, H. (2013). A related entity based approach for knowledge base acceleration. In *Proceedings of the twenty-second text REtrieval conference (TREC 2013)*.

Long, J. S., & Ervin, L. H. (2000). Using heteroscedasticity consistent standard errors in the linear regression model. *The American Statistician, 54*(3), 217–224.

Luhn, H. P. (1958). A business intelligence system. *IBM Journal of Research and Development, 2*(4), 314–319.

Muhlbauer, A., Spichtinger, P., & Lohmann, U. (2009). Application and comparison of robust linear regression methods for trend estimation. *Journal of Applied Meteorology and Climatology, 48*(9), 1961–1970.

Nanas, N., Uren, V., De Roeck, A., & Domingue, J. (2004). Beyond TREC's filtering track. In *LREC 2004* (pp. 1651–1654).

Paternoster, R., Brame, R., Mazerolle, P., & Piquero, A. (1998). Using the correct statistical test for the equality of regression coefficients. *Criminology, 36*(4), 859–866.

Radlinski, F., & Craswell, N. (2010). Comparing the sensitivity of information retrieval metrics. In *SIGIR '10* (pp. 667–674).

Razali, N. M., & Wah, Y. B. (2011). Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson–Darling tests. *Journal of Statistical Modelling and Analytics, 2*(1), 21–33.

Robertson, S. (2002). Introduction to the special issue: Overview of the TREC routing and filtering tasks. *Information Retrieval, 5*(2-3), 127–137.
Robertson, S. E., & Soboroff, I. (2002). The TREC 2002 filtering track report. In *The Eleventh Text REtrieval Conference Proceedings (TREC '02)*. NIST.
Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity, and reliability. In *SIGIR '05* (pp. 162–169).
Soboroff, I., & Robertson, S. (2003). Building a filtering test collection for TREC 2002. In *SIGIR '03* (pp. 243–250).
Wang, J., & Zhu, J. (2009). Portfolio theory of information retrieval. In *SIGIR '09* (pp. 115–122).
Wang, J., Song, D., Lin, C.-Y., & Liao, L.. Bit and msra at trec kba ccr track 2013. In *Proceedings of the Twenty-Second Text Retrieval Conference (TREC 2013)*.
Wilcox, R. R. (2012). *Introduction to robust estimation and hypothesis testing*. Academic Press.
Yilmaz, E., & Aslam, J. A. (2006). Estimating average precision with incomplete and imperfect judgments. In *CIKM '06* (pp. 102–111).
Zhou, K., Lalmas, M., Sakai, T., Cummins, R., & Jose, J. M. (2013). On the reliability and intuitiveness of aggregated search metrics. In *CIKM '13* (pp. 689–698).