An Exploration of Evaluation Metrics for Mobile Push Notifications

Luchen Tan, Adam Roegiest, Jimmy Lin, and Charles L. A. Clarke
David R. Cheriton School of Computer Science
University of Waterloo, Ontario, Canada
{luchen.tan, aroegies, jimmylin}@uwaterloo.ca, claclark@gmail.com

ABSTRACT

How do we evaluate systems that filter social media streams and send users updates via push notifications on their mobile phones? Such notifications must be relevant, timely, and novel. In this paper, we explore various evaluation metrics for this task, focusing specifically on measuring relevance. We begin with an analysis of metrics deployed at the TREC 2015 Microblog evaluations. A simple change to the metrics, reflecting a different assumption, dramatically alters system rankings. Applying another metric, previously used in the TREC Microblog evaluations, again yields different system rankings. We find little correlation between a number of "reasonable" evaluation metrics, which suggests that system effectiveness depends on how you measure it—an undesirable state in IR evaluation. However, we argue that existing evaluation metrics can be generalized into a framework that uses the same underlying contingency table, but places different weights and penalties. Although we stop short of proposing the "one true metric", this framework can guide the future development of a family of metrics that more accurately models user needs.

1. INTRODUCTION

This paper explores the problem of evaluating push notification techniques on social media streams in a filtering application. We assume an infinite stream of social media posts such as Twitter, against which the user issues an arbitrary number of standing queries representing "interest profiles", analogous to topics in traditional ad hoc retrieval. For example, the user might be interested in poll results for the 2016 U.S. presidential elections and wishes to be notified whenever new results are published. The system's task is to identify relevant tweets from the stream and send those updates directly to the user's mobile phone via push notifications. Since such notifications are often associated with an auditory or visual cue upon arrival, each imposes a non-trivial cognitive burden on the user (even if ignored). Thus, careful control of the volume of notifications is critical to successful push strategies. This paper explores evaluation metrics for such a task.

At a high level, push notifications should be relevant, timely (provide updates as soon after the actual event occurrence as possible), and novel (users should not be pushed multiple notifications that say the same thing). Accordingly, an evaluation metric should reward systems for updates that satisfy these three main criteria. As the design space is vast, in this short paper we focus on relevance, adopting existing notions of novelty and timeliness.

We start with an analysis of metrics from the TREC 2015 Microblog track, which operationalized such a push notification task, and then re-assess submitted runs after making a minor tweak to reflect a different assumption about the user model. We then re-assess submitted runs using variants of a metric that has been applied in previous iterations of the same evaluation. Using score and rank correlations, we compare system effectiveness as measured by each metric. Our results are surprising: we find little correlation between the different metrics. This means that the answer to "which system is better" depends on how you measure it, which is undesirable from an evaluation perspective.

The contribution of this paper is twofold. First, we present the novel and surprising finding discussed above: any number of reasonable evaluation metrics give rise to significantly different system rankings. We discuss and analyze why, tracing the issue to the handling of days for which there are no relevant tweets. Second, we argue that the different existing evaluation metrics we applied can be generalized into a framework that uses the same underlying contingency table, but places different weights and penalties. Although we do not propose the "one true metric", we believe this framework can guide the future development of an evaluation metric that more accurately models user needs.

2. BACKGROUND

The application described in the introduction was operationalized in the TREC 2015 Microblog track as the so-called "scenario A" variant of the real-time filtering task [3]. Over the official evaluation period, which spanned ten days during July 2015, participating systems "listened" to Twitter's live tweet sample stream to identify relevant tweets with respect to 225 topics, 51 of which were later assessed. Each system identified up to ten tweets per day, which were putatively delivered to hypothetical users. In total, 14 groups submitted 37 runs to the evaluation. Data from this evaluation provides the starting point for our analysis.

The assessment workflow for the track was as follows:

SIGIR '16, July 17 - 21, 2016, Pisa, Italy

ISBN 978-1-4503-4069-4/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2911451.2914694

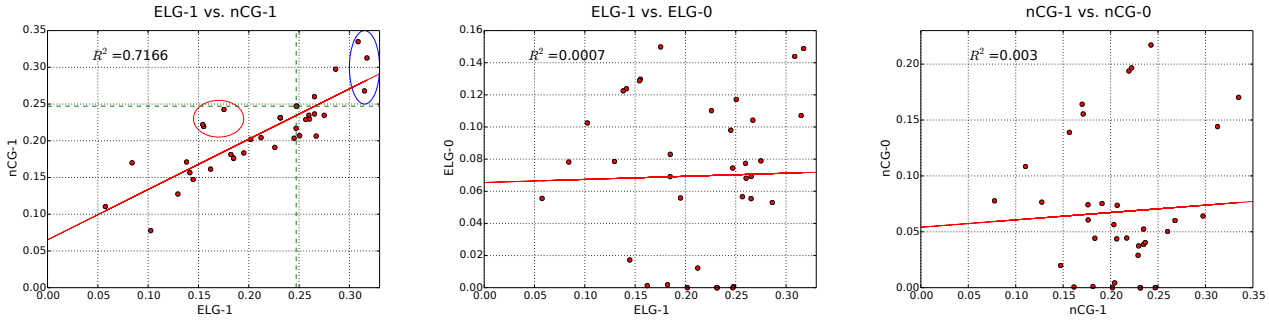

Figure 1: ELG-1 vs. nCG-1 (left), ELG-1 vs. ELG-0 (middle), and nCG-1 vs. nCG-0 (right) for all submitted runs. Plots show that the treatment of "silent days" has a large impact on system scores.

first, tweets returned by the systems were assessed for relevance using a traditional pooling process. Relevant documents were then semantically clustered into groups containing tweets that share substantively similar information. We refer the reader to previous papers for more details [5].

The two metrics used to evaluate system runs were expected latency-discounted gain (ELG) and normalized cumulative gain (nCG). These two metrics are computed for each topic for each day in the evaluation period (explained in detail below). The score for a topic is the average of the daily scores in the evaluation period. The score of a system run is the average of the scores across all topics.

Expected latency-discounted gain (ELG) was adapted from the TREC Temporal Summarization track [2]:

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

where N is the number of tweets returned and $\mathrm{G}(t)$ is the gain of each tweet: non-relevant tweets receive a gain of 0, relevant tweets receive a gain of 0.5, and highly-relevant tweets receive a gain of 1.0.

A key aspect of this metric is its handling of redundancy and timeliness: a system only receives credit for returning one tweet from each cluster. Furthermore, a latency penalty is applied to all tweets, computed as $\mathrm{MAX}(0, (100-d)/100)$, where the delay d is the time elapsed (in minutes, rounded down) between the tweet creation time and the putative time the tweet was delivered. That is, if the system delivers a relevant tweet within a minute of the tweet being posted, the system receives full credit. Otherwise, credit decays linearly such that after 100 minutes, the system receives no credit even if the tweet was relevant.

The second metric is normalized cumulative gain (nCG):

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

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

Due to the setup of the task and the nature of interest profiles, it is possible (and indeed observed empirically) that for some days, no relevant tweets appear in the judgment pool. In terms of evaluation metrics, a system should be rewarded for correctly identifying these cases and not pushing non-relevant content. If there are *no* relevant tweets for a particular day and the system returns zero tweets, it receives a score of one (i.e., perfect score) for that day; otherwise, the system receives a score of zero for that day. This applies to both ELG and nCG.

It is worth mentioning that despite superficial similarities, our task is very different from document filtering in the context of topic detection and tracking (TDT) [1]. TDT is concerned with identifying *all* documents related to a particular event—with an intelligence analyst in mind—which requires keeping track of false alarms and missed detections. In contrast, we are focused on identifying a small set of the most relevant updates to push to users, grounded in interactions with mobile devices. Furthermore, in TDT, systems must make online decisions as soon as documents arrive, whereas in our case systems can choose to push older content (subjected to the latency penalty), thus giving rise to the possibility of algorithms operating on bounded buffers. For these various reasons, TDT evaluation tools such as the decision error tradeoff (DET) curve and derivative metrics provide inspiration, but are not directly applicable.

3. ANALYSIS OF "SILENT DAYS"

In Figure 1 (left), we show a scatterplot of the official ELG scores (which we call ELG-1 for reasons that will become clear shortly) vs. nCG (specifically, nCG-1, for the same reasons). Although there is an overall correlation between ELG-1 and nCG-1 across all submitted runs, we do note that in particular cases ELG-1 and nCG-1 are capturing different aspects of effectiveness: for example, the top three runs in terms of ELG-1 (circled in blue) exhibit relatively large differences in nCG-1. There are also cases in which systems achieve high nCG-1 relative to their ELG-1 scores (the runs circled in red).

One interesting aspect of ELG-1 and nCG-1 is their handling of days in which there are no relevant documents: for rhetorical convenience, we call days in which there are no relevant tweets for a particular topic (in the pool) "silent days", in contrast to "eventful days" (where there are relevant tweets). In both ELG-1 and nCG-1, for a "silent day", the only two possible scores are one (if the system remained silent) or zero (if the system pushed any tweet). This means that an empty run (a system that never returns anything) may have a non-zero score based on how many silent days there are in each topic. As it turns out, an empty run will score 0.2471 in ELG-1 and nCG-1, shown as dotted lines in Figure 1 (left). Since this was the first year of this TREC evaluation, systems achieved high scores by simply returning few results, in many cases for totally idiosyncratic reasons—for example, the misconfiguration of a score threshold.

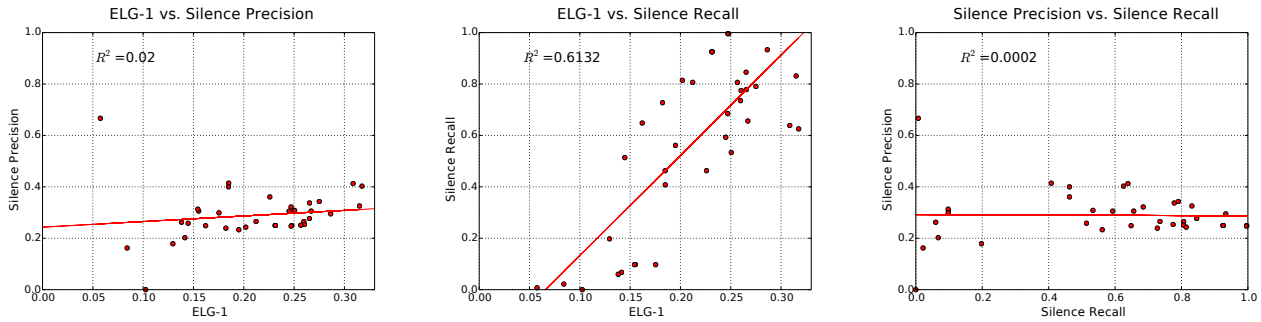

Figure 2: Characterizing the effects of "silent days": silence precision vs. ELG-1 (left), silence recall vs. ELG-1 (middle), and silence precision vs. silence recall (right). Systems score well by learning when to "shut up".

As an alternative, what if we did not reward systems for remaining silent? That is, on a silent day, all systems receive a zero score, no matter what they did. We call these variant metrics ELG-0 and nCG-0 (in contrast to ELG-1 and nCG-1). We can justify this from the user perspective in that for a silent day, the user does not obtain any relevant information regardless of system output (since there *are* no relevant documents). In this case, how would the user know to "reward" a system for remaining silent? That is, properly determining a silent day requires global knowledge (e.g., from pooling), which no individual user has access to.

In Figure 1, we show scatterplots of ELG-1 vs. ELG-0 (middle) and nCG-1 vs. nCG-0 (right). We see no discernible relationship between each pair of metrics, which suggests that the handling of silent days *is the most critical part of each metric*, in that different (reasonable) formulations yield dramatically different results and system rankings. In fact, we would go as far as saying that effectiveness under ELG-1 and nCG-1 is primarily dominated by a system's ability to identify the silent days. Under both metrics, systems do well by learning when to "shut up".

This observation is further illustrated by the scatterplots in Figure 2, where we show silence precision vs. ELG-1 (left), silence recall vs. ELG-1 (middle), and silence precision vs. silence recall (right) for all runs. Silence precision and recall follow the usual definitions of precision and recall, but with respect to identifying the silent days. Since each topic has an equal number of days, there is no difference between micro- and macro-averaging. Across all the topics, 24.7% of all days are completely silent, while another 6.7% have relevant but redundant material. We see that there is a slightly positive (but very weak) correlation between ELG-1 and silence precision. The middle graph, in effect, shows that systems achieve a high ELG-1 score by achieving a high silence recall—i.e., getting a good score is dominated by a system knowing when to "shut up". Although systems with comparable silence recall can differ substantially in ELG-1, we were surprised by how much of the variance in the official metric can be explained by silence recall alone.

The right graph in Figure 2 shows the tradeoffs systems make with respect to precision and recall. On the right edge of the plot are cases where the systems are almost always quiet, achieving nearly perfect recall; in the left lower corner is a system that never "shuts up", and hence its precision and recall are both zero. It is interesting to note that some systems perform poorly in both precision *and* recall—they don't push content when there's relevant content and don't "shut up" when there's no relevant content.

To our knowledge, we are the first to make this observation about the huge impact of silent vs. eventful days in the current evaluation of push notification. However, we withhold judgment as to whether the current TREC metrics represent the "right" approach: from the user perspective, since push notifications are associated with high cognitive effort (because they may interrupt the user), perhaps we should force systems to focus on learning when to "shut up". On the other hand, having such highly binarized scores on the silent days creates many issues for system tuning, since it creates discontinuities in the objective. We observe similar issues when trying to optimize a metric such as precision at rank one for question answering.

4. GAIN AND PAIN

What are other reasonable ways in which we can evaluate the push notification task? A simple and intuitive utility-based metric would be to reward "gain" based on delivery of relevant information and to deduct "pain" based on delivery of non-relevant information. In fact, the TREC 2012 Microblog track employed exactly such a metric, called T11U [4], itself derived from the linear utility metrics used in the TREC filtering tracks.

We adopt a slightly different but mathematically equivalent formulation as follows:

$$\text{T11U} = \alpha \cdot \mathcal{G} - (1 - \alpha) \cdot N_x \tag{3}$$

where $\mathcal{G}$ is total gain, N_x is the number of non-relevant documents pushed, and α controls the relative weight of gain vs. pain. Note that in T11U, the total gain factors in different relevance grades, the latency penalty, and the treatment of redundant tweets in exactly the same way as ELG.

In Figure 3, we show a scatterplot of ELG-1 vs. T11U with $\alpha = 0.66$, which was the value used in TREC 2012. This value can be understood as setting the gain of a relevant notification (highest relevance grade, no temporal penalty, not redundant) equal to the pain of returning two non-relevant updates. With this setting, we do see reasonable positive correlation between ELG-1 and T11U overall, but this correlation is highly misleading. According to T11U, very few systems achieve positive utility overall—that is, the gain from pushing relevant content is not sufficient to offset the pain from pushing non-relevant content. Furthermore, the relatively large cluster of runs which score around zero in T11U vary widely in ELG-1, from around 0.2 to over 0.3, with the highest T11U score being somewhere in the middle of this band. In other words, poorly-performing systems

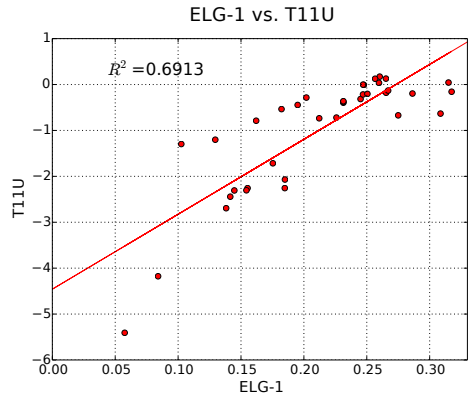

Figure 3: ELG-1 vs. T11U.

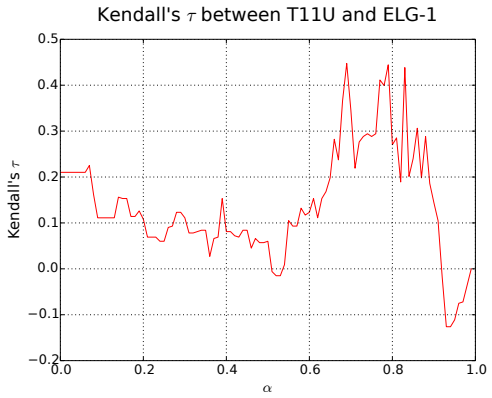

Figure 4: Kendall's τ between T11U and ELG-1 for different values of α.

score low in both ELG-1 and T11U, but beyond that, T11U and ELG-1 exhibit a weak correlation at best.

Of course, absolute scores and relative system rankings depend on the α parameter that balances gain vs. pain, and the setting of $\alpha = 0.66$ was arbitrary. What would the evaluation results look like for different settings of α? The answer is shown in Figure 4, where we sweep the α parameter and compute Kendall's τ with respect to ELG-1 for each setting. The results show that Kendall's τ varies substantially, ranging from moderate correlation to non-existent and even slightly negative correlation. We have shown above that high correlations can be misleading, and this plot shows that $\alpha = 0.66$ is around the highest Kendall's τ we can obtain regardless. These results suggest that T11U and ELG-1 are measuring quite different aspects of effectiveness.

5. TOWARD A GENERAL FRAMEWORK

Let us take stock of our findings so far: we have evaluated runs from the TREC 2015 Microblog track using official as well as alternative metrics (ELG-1, ELG-0, nCG-1, nCG-0, and T11U). In comparing the metrics, we observe many inconsistencies in terms of both system scores and relative rankings. In other words, "which system is better" depends on what measure we use. From an evaluation perspective, this is not desirable because researchers lack consistent guidance for algorithm development.

To address this issue, we propose a general evaluation framework built around the contingency table shown in Table 1. At the core, our framework is utility-based in that gain is rewarded for pushing relevant content ($+G_E$) and pain is deducted for pushing non-relevant content ($-P_E$ and $-P_0$). However, a key insight is the explicit separation of eventful and silent days, which our ELG-1 vs. ELG-0 experiments have shown to be critical in system evaluations.

System action	"eventful days"	"silent days"
Pushed relevant	$+G_E$	-
Pushed not-relevant	$-P_E$	$-P_0$
Stayed silent	$-S_E$	$+S_0$

Table 1: The contingency table for a general evaluation framework for push notifications.

Our evaluation framework is general in that the metrics we have examined in this paper can be viewed as specific instantiations of the parameters in Table 1. For example, T11U sets G_E and P_E (based on α) but ignores the final row, and furthermore does not make the distinction between eventful and silent days. ELG-1 and ELG-0 make different choices on S_0, how systems should be rewarded for staying silent on silent days, but both set P_E and P_0 to zero. That is, no pain is deducted for pushing non-relevant content.

Using the framework presented in Table 1 as a guide, we can imagine a family of metrics beyond those already presented. For example, we might augment T11U by creating a distinction between eventful and silent days, thus arriving at a metric that is closer to ELG-1 or ELG-0. We might set P_E differently from P_0 to create more nuanced distinctions in a T11U-like metric. Different ratios between these weights also give rise to emphasis on different aspects of the push notification problem.

The question remains on how to properly set the gain and pain weights in the contingency table—and we presently provide no concrete answer, expect to say that further studies in user modeling are necessary. For example, we have presented two plausible scenarios (ELG-1 and ELG-0) for the treatment of silent systems on silent days: a user study is necessary to decide which alternative (or neither) matches user preferences. Although the development of the "one true metric" is beyond the limited scope of this short paper, our framework contributes to a step toward that goal.

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

6. REFERENCES

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

[2] J. Aslam, M. Ekstrand-Abueg, V. Pavlu, F. Diaz, R. McCreadie, and T. Sakai. TREC 2014 Temporal Summarization Track overview. *TREC*, 2014.

[3] J. Lin, M. Efron, Y. Wang, G. Sherman, and E. Voorhees. Overview of the TREC-2015 Microblog Track. *TREC*, 2015.

[4] I. Soboroff, I. Ounis, C. Macdonald, and J. Lin. Overview of the TREC-2012 Microblog Track. *TREC*, 2012.

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