On Sparsity and Drift for Effective Real-time Filtering in Microblogs

M-Dyaa Albakour
University of Glasgow, UK
dyaa.albakour@glasgow.ac.uk

Craig Macdonald
University of Glasgow, UK
craig.macdonald@glasgow.ac.uk

Iadh Ounis
University of Glasgow, UK
iadh.ounis@glasgow.ac.uk

ABSTRACT

In this paper, we approach the problem of real-time filtering in the Twitter Microblogging platform. We adapt an effective traditional news filtering technique, which uses a text classifier inspired by Rocchio's relevance feedback algorithm, to build and dynamically update a profile of the user's interests in real-time. In our adaptation, we tackle two challenges that are particularly prevalent in Twitter: sparsity and drift. In particular, sparsity stems from the brevity of tweets, while drift occurs as events related to the topic develop or the interests of the user change. First, to tackle the acute sparsity problem, we apply query expansion to derive terms or related tweets for a richer initialisation of the user interests within the profile. Second, to deal with drift, we modify the user profile to balance between the importance of the short-term interests, i.e. emerging subtopics, and the long-term interests in the overall topic. Moreover, we investigate an event detection method from Twitter and newswire streams to predict times at which drift may happen. Through experiments using the TREC Microblog track 2012, we show that our approach is effective for a number of common filtering metrics such as the user's utility, and that it compares favourably with state-of-the-art news filtering baselines. Our results also uncover the impact of different factors on handling topic drifting.

Categories and Subject Descriptors

[**H.3.3 Information Search and Retrieval**]: Information filtering

Keywords

Real-time filtering; Microblogs; Event detection

1. INTRODUCTION

Social media have grown as massive networks of information publishers and consumers. In these networks, consumers may have difficulties to keep up with the vast amounts of real-time information and publishers have no way to ensure that their content can reach their targeted audience. Information filtering (IF) [5] can help both publishers and consumers by ensuring that only relevant information is delivered to the right audiences. We aim to approach an emerging problem in the area of IF on social media. In particular, we focus on the Twitter microblogging platform, as it is one of the most popular and fastest growing social media platforms. Information filtering in Twitter faces unique challenges that do not necessarily exist in traditional domains such as newswire, which have been extensively studied in the literature, e.g. [2, 18]. Indeed, the nature of tweets makes them dissimilar to news articles for several reasons. Unlike news articles, tweets are a form of user-generated text. A tweet is a very short text, as it is bounded with a 140 character limit, and typically consists of few terms (around 11 terms on average [10]). Moreover, Twitter operates in a real-time fashion where users can immediately access tweets posted by tweeters they are subscribed to (their social space) or in fact any public tweet. More interestingly, users in Twitter may quickly reflect on news and events happening in the real world [12, 22]. Finally, Twitter has been rapidly growing in terms of active users and volume of activity, which can reach hundreds of millions of tweets in a day [25].

In this paper, we study the problem of real-time filtering in Twitter. We devise a solution by building on an effective news filtering technique that is based on the text classification approach of Incremental Rocchio [2]. In particular, we introduce novel adaptations to Incremental Rocchio to deal with the unique challenges in Twitter. The first challenge is the shortness of the documents (tweets), a *sparsity* problem that is less prevalent in the news domain and therefore traditional filtering approaches may not work as effectively on tweets. To deal with the acute sparsity issue, we propose to use a query expansion (QE) approach to enrich the representation of the user's profile (the explicit relevant judgments of the user) during the filtering process. In particular, our QE approach derives additional terms and documents that are relevant and timely during the filtering process. The second challenge is the potential *drift* over time where the interests of the user swing between different aspects (subtopics) of the more general topic or particular aspects of the topic become more popular than other aspects over time. We hypothesise that events in the real world may be the main reason behind this drift. To give the reader a concrete example, let us say the topic of filtering is a football match. A few days before the match, the user will be interested in tweets about which players are going to play or who are going to miss. During the match, the interest drifts towards goals and after the match it drifts towards reactions on the game. To tackle this topic drift issue, we propose to modify the classifier such that it recognises short-term interests (emerging subtopics),

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2505515.2505709.

and balances between the importance of short-term interests and the long-term interests in the overall topic. However, our modification requires correctly identifying what is considered to be the short-term interests in the topic, which can be a challenging task. For example, while filtering, we can consider the explicit user judgments in an arbitrary interval, e.g. the current day, to represent the short-term interests, but the drift may occur more than once during the day. Since events may be the main reason for topic drifts, we explore the automatic detection of such events from either the Twitter stream itself or external newswire streams, to identify potential times of drift to other aspects of the topic and to capture the short-term interests automatically.

Using the real-time filtering task of the Microblog track of TREC 2012, we thoroughly evaluate our adaptations against the standard Incremental Rocchio and a state-of-the-art news filtering technique based on Regularised Logistic Regression [30]. Our empirical results show that the traditional news filtering techniques are not effective in the context of tweet filtering. They also show the power of the adaptations we propose to improve these filtering techniques and mitigate the problems of sparsity and topic drifting in Twitter.

The remainder of the paper is structured as follows. In Section 2, we present related work. Section 3 provides a formal description of the real-time tweet filtering problem and describes the filtering approaches we build on. Section 4 describes the sparsity challenge and our query expansion approach to deal with it, while Section 5 describes the challenge of topic drifting and our solution to it. Section 6 describes our experimental methodology and Section 7 reports and discusses the results of the experiments. Finally, Section 8 summarises our conclusions and future work directions.

2. RELATED WORK

IF and information retrieval (IR) are similar in the sense that they both aim to provide users with access to information and they deal with the same kind of information [5]. However, the fundamental difference is that IF deals with information needs spanning over a long period of time, i.e. a user *profile*, rather than a one-shot *query* that vanishes at the end of a search session. The user profile can be maintained over time and can possibly be improved if the user provides explicit feedback to the system. This is the case of adaptive filtering [2], which is the focus of this paper. Adaptive filtering has been studied extensively in the news domain and several approaches were proposed to tackle adaptive filtering in news, e.g. [6, 18, 30]. Some adaptive news filtering approaches treat the problem as a retrieval task and introduce an adaptive threshold on the retrieval score to make a binary decision. The threshold in this case can be defined as a function of the retrieval scores for documents which are explicitly judged relevant by the user. Examples include Bayesian inference on content representation nodes [6] and thresholding on the Okapi probabilistic model [18]. Other approaches are those which treat filtering as a special case of a single-label text classification, where relevance feedback is then used to update the classifier. An example of such approaches is Incremental Rocchio [2], which uses Rocchio's relevance feedback in the vector space model. Incremental Rocchio has shown to be effective in news filtering and it was among the best performing approaches in the TREC Filtering track [19]. Text classification with logistic regression has also been popular in adaptive news filtering and is considered as the state-of-the-art news filtering approach. For example, Zhang [30] introduces a news filtering method that employs a regularised logistic regression model using a Bayesian framework, which outperformed Incremental Rocchio. Similarly, Yang *et al.* [27] found that a regularised regression model is more robust than Incremental Rocchio when tuning its parameters across different corpora. As a summary, Incremental Rocchio and Regularised Logistic Regression are effective traditional news filtering methods. We aim, in this paper, to investigate whether these methods are effective when applied on tweet filtering. We also build on Incremental Rocchio to tackle specific challenges in tweet filtering.

Despite of the wealth of research in adaptive filtering of news, there has been little work done on adaptive filtering of tweets until TREC has recently introduced the real-time filtering task in the Microblog track 2012. However, since we build on a text classification approach to perform adaptive tweet filtering, it is worth reviewing work on tweet classification to highlight challenges that can be relevant to our problem. For example, Hu *et al.* [11] studied tweet clustering where they used a bag-of-word approach. They addressed the sparsity problem stemming from the shortness of the tweets. Using Wikipedia as an external knowledge source, they extracted useful related phrases that can enrich the cluster representation. Similarly, Sriram *et al.* [23] proposed to use metadata about the author of the tweet to enrich the bag-of-word representation of the tweets. Other work on short user-generated content includes the classification of SMS, short emails and blog comments. Cormack *et al.* [7] studied spam classification in this type of content and pointed out that shortness may have an impact on the classifier's performance. They suggested performing lexical expansion to derive a richer word and character n-gram representation, which can account for the shortness of messages. We also aim to address the sparsity issue but instead of using external knowledge sources as in some of the aforementioned techniques, we apply query expansion that relies on a pseudo-relevance feedback approach to extract terms that are both relevant and timely.

3. ADAPTIVE FILTERING OF TWEETS

As discussed in Section 2, adaptive filtering was previously investigated in the TREC Filtering track [19] within the news domain. In adaptive filtering, and unlike a traditional search query, user information needs reflect a long-term interest and they are represented in a user's profile, which is usually a set of documents considered relevant by the user [5]. Moreover, instead of searching a static document collection, an adaptive filtering system examines a stream of incoming documents over time and decides for each document whether it matches the user's profile such that it is displayed immediately to the user [2]. Generally, the filtering system starts with a user profile and a very small number of positive feedback examples (relevant documents). The profile can then be adapted using the feedback information provided by the user for the displayed documents [5]. The problem we are tackling in this paper is the real-time adaptive filtering of tweets. In the remainder of the section, we first provide a formal description of the real-time tweet filtering problem. We then describe the state-of-the-art news filtering methods we build on for our solution.

3.1 Problem Formulation

We regard the problem of real-time tweet filtering as an instance of the adaptive filtering problem. Given a user u

with an initial information need, i.e. an input query q, at a certain starting time t_s and a small set of positive example tweets prior to t_s, the filtering system should decide whether subsequent tweets posted after t_s are relevant to the user and therefore should be displayed to the user. This should allow the user to stay updated in real-time by browsing relevant tweets for a developing topic. The user can examine these tweets and provide explicit feedback to the filtering system whether they are relevant or not. The filtering system can hence adapt the user's profile, which is defined to be the set of feedback tweets provided by the user. Indeed, Twitter already implements a filtering functionality ("X new tweets") in its search page,[1] however it does not allow the user to provide explicit judgments and, to our knowledge, there is no study published on the effectiveness of their filtering tool.

3.2 Filtering with a Text Classifier

We introduce two effective adaptive filtering methods on news that can be employed on tweets, namely Incremental Rocchio [2], and Regularised Logistic Regression [30].

3.2.1 Incremental Rocchio

This method is based on a classifier that uses the popular Rocchio's relevance feedback approach [21] to build a profile of the user's interests, which is then updated online using the explicit judgements provided by the user [2]. More specifically, at each point of time t, the profile of the user is represented in the term vector space model by a vector $\vec{c}_t$, which is called the *centroid* of the user's interests. The centroid is calculated as follows:

$$\vec{c}_t = \frac{\alpha}{|R_t|} \cdot \sum_{d_i \in R_t} \vec{d_i} - \frac{\beta}{|N_t|} \cdot \sum_{d_i \in N_t} \vec{d_i} \quad (1)$$

where R_t is the set of tweets judged *relevant* by the user so far (at time t), N_t is the set of tweets judged *non-relevant* by the user so far (at time t), α and β are co-efficient parameters for positive and negative feedback documents respectively. For each incoming tweet $\vec{d}$, the cosine similarity is computed between the centroid at the time at which the tweet arrives and the actual tweet $Sim(\vec{c}_t, \vec{d})$. If the cosine value $Sim(\vec{c}_t, \vec{d})$ exceeds a certain threshold η_R, the tweet is predicted relevant and is therefore displayed to the user, otherwise it is not. When the tweet is judged relevant by the user it will be added to the set of relevant tweets for the next time point R_{t+1}, otherwise it will be added to N_{t+1}, and as a result the centroid c_{t+1} will be updated. Our initial experiments have revealed that penalising negative documents does not improve tweet filtering effectiveness and hence we only consider positive feedback documents ($\beta = 0$), i.e the centroid is reduced to:

$$\vec{c}_t = \frac{1}{|R_t|} \cdot \sum_{d_i \in R_t} \vec{d_i} \quad (2)$$

The terms in the vector space are weighted using any appropriate term weighting models such as BM25 [20].

3.2.2 Regularised Logistic Regression

This state-of-the-art filtering method is based on a logistic regression learner with a Gaussian prior [30]. More specifically, logistic regression estimates the posterior probability of an unobserved variable (a topic) given an observed variable (a tweet) using a log linear function:

$$P(y = 1|\vec{x}, \vec{w}) = 1/(1 + e^{-\vec{w}.\vec{d}}) \quad (3)$$

where $\vec{x}$ is a tweet vector in the term vector space, $\vec{w}$ is the vector of regression coefficients, and $y \in \{0, 1\}$ is the boolean topic output variable corresponding to whether the tweet $\vec{x}$ is non-relevant or relevant to the topic.[2] During adaptive filtering, at each point of time t, the profile of the user is represented with the training set of labelled tweets $D_t = \{(\vec{x_1}, y_1), ..., (\vec{x_n}, y_n)\}$. A *regularised* regression model is applied to find the *Maximum a Posteriori* (MAP) estimation of the regression coefficients:

$$\vec{w}_{MAP} = \underset{\vec{w}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} log(1 + e^{-y_i \vec{w} \vec{x}}) + \lambda \|\vec{w} - \vec{\mu}\|^2 \right\} \quad (4)$$

Equation (4) has a unique and numerically stable solution using a stochastic gradient descent (SGD) algorithm [29]. The second term in this objective function is equal to adding a Gaussian prior with the mean $\vec{\mu}$ and variance covariance matrix $1/2\lambda.I$, where I is the identity matrix. Following the work in [27], we fix the mean $\vec{\mu}$ of the Gaussian prior to 0, and tune the variance parameter λ for the the best filtering performance. Once the regression coefficients are estimated, the filtering prediction can be made for each incoming tweet $\vec{x}$ by calculating the posterior probability $P(y|\vec{x}, \vec{w})$. If it exceeds a certain threshold η_L, then the tweet is considered relevant, otherwise it is not. Note that $\vec{w}$ is constantly updated whenever a relevance judgment is available from the explicit feedback provided by the user. As in Incremental Rocchio, the terms in the vector space are weighted using any appropriate term weighting model. The Regularised Logistic Regression has shown to outperform Incremental Rocchio in adaptive news filtering [27, 30]. In our experiments, we will investigate the effectiveness of both the Incremental Rocchio and the Regularised Logistic Regression news filtering approaches when applied on tweets.

In addition, we also discuss in the following sections how we modify Incremental Rocchio to tackle specific challenges in filtering tweets. We choose to devise these modifications on Incremental Rocchio due to its generative nature, which may well suit adaptive filtering in the highly dynamic environment of Twitter, in comparison to the more complex discriminative regression approach that requires a large number of training examples to have a stable performance.

4. HANDLING SPARSITY

The acute sparsity issue in filtering tweets is a unique challenge caused by the shortness of tweets. As discussed before, this problem is less prevalent, for instance, in traditional filtering tasks, such as adaptive filtering of news streams for which the Incremental Rocchio method we build on was originally designed. Sparsity is particularly problematic when we know little about the user's interests or the topic itself, i.e. when the filtering system knows only a small number of documents (tweets) that the user is interested in. As a result, the centroid of the Rocchio's classifier has a small number of terms, which may not be representative of the topic or the user's interests.

[1] https://twitter.com/search/

[2] $\vec{x}$ is identical to $\vec{d}$, but we use the notation $\vec{x}$ instead for uniformity with conventional logistic regression notations.

To tackle the sparsity of the initial centroid representation in incremental Rocchio, our approach is to make use of query expansion (QE), a traditionally successful method for improving the retrieval performance in a number of IR tasks, such as adhoc retrieval [26]. In a *static* document collection, QE automatically derives terms that can be added to a user's original query from a pseudo-relevant set of documents (the initial set of highly ranked documents retrieved for the query). We apply QE as a mechanism to enrich our knowledge of the topic and the user's interests by deriving terms that are both topically relevant and timely. For this, and to initialise our classifier, we apply QE using the user's query and a document collection comprising tweets publicly posted up to the time of issuing the query or triggering an interest in a tweet (by providing an explicit positive feedback).

Formally, at a certain time t_i and for a user with a query q, we use a set of tweets that were posted before t_i. We denote this set of tweets by T_s. We limit T_s to an arbitrary period of time before t_i. We can then apply QE to score terms in the pseudo-relevant set of tweets $T_q \subseteq T_s$. We denote this set of terms by E. In particular, the centroid of the classifier at a certain time t can be computed as follows:

$$\vec{c_t} = \frac{1}{|R_t|} \cdot \sum_{d_i \in R_t} \vec{d_i} + \vec{e} \tag{5}$$

where $\vec{e}$ represents a vector that is comprised of all the terms in the expansion set E weighted with their scores provided by the QE weighting model. This vector can be expressed formally as $\vec{e} = \sum_{e_i \in E} w(e_i, q) \cdot \vec{\hat{e}_i}$, where $\vec{\hat{e}_i}$ is the standard basis vector for dimension (term) e_i. Note that $\vec{e}$ is particularly useful when the set of positive feedback documents R_t is small, e.g. only one tweet, which is the main purpose of adding this component to the centroid.

Furthermore, we investigate adding the entire set of pseudo-relevant tweets to the centroid. In this case, the vector is calculated as follows:

$$\vec{c_t} = \frac{1}{|R_t|} \cdot \sum_{d_i \in R_t} \vec{d_i} + \frac{1}{|T_q|} \cdot \sum_{d_i \in T_q} \vec{d_i} + \vec{e} \tag{6}$$

As for the documents in R_t, each document in T_q is weighted with an appropriate term weighting model.

5. TOPIC DRIFTING

One of the challenges of filtering in general is topic drifting [2]. This can occur because the interest of the user may change or events related to the considered topic develop. In particular, we aim to address the problem of topic drift in adaptive tweet filtering, where the problem is particularly prevalent due to the highly dynamic environment of Twitter [25]. We first discuss this phenomenon with examples from the TREC Microblog track. Then, we propose our methods to deal with drifting and enhance the effectiveness of filtering tweets.

5.1 Illustrative Examples

To illustrate how topic drifting occurs and affects the real-time filtering of tweets, we perform the following analysis. Considering the training topics of the real-time filtering task of the Microblog track 2012, we examine the relevant tweets for each topic that a real-time filtering system should trigger as relevant and present to the user, who eventually provides positive feedback indicating they are actually relevant to the topic. The main purpose is to understand how the topic develops over time. To achieve this, we analysed over time the distribution of terms in those relevant tweets. Figure 1 presents three coloured grids representing the distribution of stemmed terms in relevant tweets for each day in the considered filtering period. The intensity of the colour in each cell represents the frequency of the corresponding term in the relevant tweets during the corresponding day. For each topic, we only consider terms of a medium frequency within the relevant tweets. The reason for discarding the most frequent terms is that they do appear in almost all the relevant tweets. They represent the long-term aspects of the topic and would not give insights on how the topic develops. Moreover, terms of low frequency only appears occasionally and may not be representative of certain aspects of the topic.

By inspection of the grids, it can be observed that there are certain terms which peak on different days. This indicates that there are different aspects (subtopics) of the original topic that become more popular over time, or there are certain events that occurred and drifted the topics into new aspects. To validate this observation, we manually examine the content of the relevant tweets for each topic in question over time as well as the content of any hyperlinked document in these tweets. For example, for the first topic "BBC World Service staff cuts", on the first day (Jan 24), the terms 'online' and 'budget' were important. This is the day when the BBC announced that it was "cutting its online budget" and as a result more tweets containing those terms were publicly posted on Twitter. Later, these terms were rarely used. Moreover, on the third day (Jan 26), we can observe that other terms started to emerge. On that day, two different stories were spread about the topic of BBC cuts. The first one is that the BBC is going to "slash 650 jobs" and the second one is about "cutting five BBC language services". As a result, the terms '650', 'language' and 'service' have become dominant on that day. A similar discussion applies to the second and the third grids corresponding to the other two topics (21 and 41). For topic 21 "Emanuel residency court rulings", Jan 24 corresponds to the day when the news about "Rahm Emanuel being booted off ballot by the Appellate court and becoming ineligible for the mayoral race" emerged. Later on Jan 26, several tweets were posted saying that "the State high court will hear Rahm's appeal". For topic 41 'Obama birth certificate', Jan 25 is the day where the news spread about "a swear of a Hawaii official that Obama birth certificate does not exist". Later on Jan 28, several tweets were posted about "a new Hawaii bill that would make it possible to release Obama's birth records".

As a summary, we can see that certain terms become popular at different times as the topic develops and this may be due to the occurrence of real-world events. To deal with drifting, we propose to make a distinction in the user profile between the short-term interest in emerging subtopics and the long-term interest in the overall topic. In the following, we describe how the Incremental Rocchio approach, introduced in Section 3, can be modified to balance between short-term and long-term interests in order deal with the drifting phenomenon and aid tweet filtering.

5.2 Balancing Short- and Long-term Interests

To deal with the topic drifting illustrated in the previous examples, we suggest to modify the representation of the classifier's centroid defined in Equation (2). Our idea is to dynamically change the centroid over time by introducing

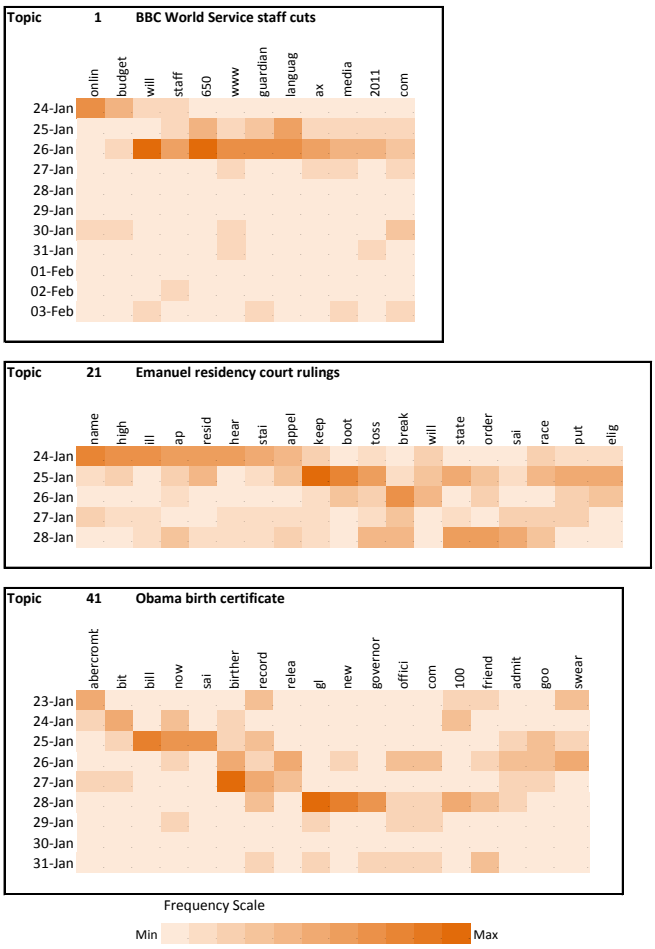

Figure 1: Distribution of medium frequency stemmed terms over time for the relevant tweets of three training topics of the TREC 2012 Microblog track filtering task. The colour intensity in each cell represents the frequency of the corresponding term during that day. We only consider terms of a medium frequency within the relevant tweets of each topic.

a decay factor that balances between short-term and long-term interests. In particular, the profile of the user can be represented by the vector:

$$\vec{c}_t = \frac{1-\sigma}{|L_t|} \cdot \sum_{d_i \in L_t} \vec{d_i} + \frac{\sigma}{|S_t|} \cdot \sum_{d_i \in S_t} \vec{d_i} \quad (7)$$

where $0 \leq \sigma \leq 1$ is a decay factor that determines the balance between short-term and long-term interests, L_t is the set of all the relevant tweets so far representing the long term interests in the overall topic, i.e L_t=R_t, and S_t is the set of the most recent relevant tweets representing the short term interests. Considering the example in the first grid of Figure 1, over time, L_t will contain the most frequent terms of the topic e.g. 'bbc', 'world', 'service' and 'cuts'. On the other hand, S_t will boost over time recent subtopics which are becoming popular. For instance, 'online' and 'budget' on Jan 24 or '650' and 'languag' on Jan 26. However, defining S_t to contain representative tweets of the short-term interests can be a challenging problem. We explore two intuitive adhoc methods for defining S_t:

(A) *Arbitrary adjustments*: We consider S_t to be the most n recent tweets added to R_t, where n is a free parameter.

(B) *Daily adjustments*: We consider S_t to be the tweets in R_t that have been added in the current calendar day, i.e. $S_t = \{d_i : d_i \in R_t \wedge time(d_i) \in today\}$.

Both methods (A) and (B) may still produce inaccurate representation of the short-term interests. For instance, when using daily adjustments (B) to define S_t, our balancing approach may not be capable of handling cases where the topic drifts more than once within the same day. Indeed, in the examples of Figure 1, there are such cases (Jan 26 of topic 1). However, we observe in those examples that events trigger new aspects of the topic. In the next section, we propose to use an approach for event detection from social media and newswire streams to identify points of time for potential topic drift, such that S_t is automatically adjusted rather than using arbitrary or daily adjustments. Indeed, we show, in our experiments, that using the event detection method is better than the adhoc methods (A) and (B) for adjusting S_t in the terms of the achieved filtering quality.

Moreover, our balancing approach has an important parameter, which is the decay factor σ that practically adjusts the emphasis on the short-term interests. In our experiments, we aim to conduct a sensitivity analysis on how the decay factor σ affects the filtering quality.

Finally, the balancing approach is independent of the representation of the centroid. Indeed, it can be combined with the QE approach for tackling sparsity in Section 4, where the centroid is calculated as follows:

$$\vec{c}_t = (1-\sigma) \cdot \left[\frac{1}{|L_t|} \cdot \sum_{d_i \in R_t} \vec{d_i} + \frac{1}{|T_q|} \cdot \sum_{d_i \in T_q} \vec{d_i} + \vec{e} \right] + \frac{\sigma}{|S_t|} \cdot \sum_{d_i \in S_t} \vec{d_i} \quad (8)$$

5.3 Event Detection to Aid Filtering

As discussed in the previous section, identifying a representative tweet set of short-term interests is a challenging problem. Our main objective here is to identify the points of time at which topic drift happens to automatically adjust the set S_t such that it contains relevant tweets, which have been posted after the topic has drifted. Our assumption is that real-world events result in a shift in interest towards new aspects of the topic, and detecting those events helps in identifying the potential drift. We apply a method of event detection from social media streams that has been recently proposed [1]. In this section, we describe this method and how we use it to automatically adjust S_t. The idea is that social media may reflect real-world events, and hence when an event related to a given topic occurs, it is expected to find (i) topically related social posts about it and (ii) increasing microblogging activity [31], causing peaks of tweeting rates during the event (bursts). In other words, the stream of tweets can be used as a source of evidence to detect events. Similarly, news articles reflect on real-world events and the previous assumptions (i) and (ii) hold when considering news articles in a newswire stream. Therefore, a newswire stream can be an another source of evidence to detect events.

Considering the Twitter stream, it is assumed that a certain point of time t_i is characterised by the microblogging activities within a given time frame $(t_i - t_{i-1})$. The microblogging activities are represented with a set of tweets (documents) within the given time frame $(t_i - t_{i-1})$, which is

denoted by D_i. Note that the fixed time frame is defined using an arbitrary sampling rate $\theta; \forall i : t_i - t_{i-1} = \theta$. To decide whether t_i represents a potential time for an occurrence of an event related to the topic, the two components of evidence identified above are quantified, namely the *topically related* tweets and the *increased tweeting* activity. First, the score $S(q, D_i)$ is quantified representing how the tweets D_i are related to the topic q (the query) that the user is interested in. In particular, $S(q, D_i)$ is estimated using the CombSUM voting technique [13], which estimates the final score of the tweet set D_i by aggregating the scores (the votes) of individual tweets within D_i. Second, it is necessary to quantify the change in the tweeting rate about the topic – the volume of tweets over time related to the topic – observed within the time frame $(t_i - t_{i-1})$ when compared to observations over previous time frames, i.e. comparing the score $S(q, D_i)$ retrospectively to the scores $S(q, D_{i-1})$, $S(q, D_{i-2})$, .. , $S(q, D_{i-k})$. This is achieved by applying Grubb's test [8], which determines if the tweeting rate about the topic at the current time t_i is an outlier with respect to the tweeting rates of previous observations in a window (t_{i-k}, t_i) of size k. Grubb's test gives a binary decision for each considered point of time. More specifically, the tweeting rate about the topic at the current time $r_i = S(q, D_i)$ is an outlier if:

$$r_i - \overline{x}_{i,k}/\sigma_g^2 > z \quad (9)$$

where $\overline{x}_{i,k}$ is the mean tweeting rate in the window (t_{i-k}, t_i), σ_g is the standard deviation of the tweeting rates in the window (t_{i-k}, t_i), and z is a fixed threshold.

Of course, we can also apply the event detection approach on an external source of newswire instead of the Twitter stream itself. In this case, each newswire article can be seen as a tweet and each point of time t_i is characterised with the news articles posted in the time frame $(t_i - t_{i-1})$.

Finally, we apply the event detection approach to aid filtering as follows. During the filtering process, periodically at certain times t_i, corresponding to time frames $(t_i - t_{i-1})$ of size θ, we apply the Grubb's test to decide if the current point of time represents an occurrence of an event related to the topic. If this is case, we can reset the recent relevant tweet set S_t. We continue adding tweets judged relevant by the user to S_t until we detect another event using the Grubb's test. The assumption we make here is that a new event erases the current short-term interests of the user. However, those are still considered in the overall long-term component L_t. The process of adjusting S_t with this approach is illustrated in Algorithm 1.

```
Initialise S_{t_s}, L_{t_s};
while filtering tweets do
    d ← next tweet;
    if θ has elapsed then
        if Grubb's test is positive then
            S_t ← φ ;
        end
    end
    if (d is positive) AND (feedback on d is positive)
    then
        S_t ← S_t ∪ d;
        L_t ← L_t ∪ d;
    end
end
```

Algorithm 1: The filtering process with event detection

6. EXPERIMENTAL METHODOLOGY

In this section, we describe our experimental methodology. We first identify the main research questions that the experiments aim to answer in light of the discussion before in Sections 3, 4 and 5. Then we describe our experimental setup. The results of the experiments are further reported and discussed in Section 7.

6.1 Research Questions

Our experiments aim to validate the models we have described in the previous sections to uncover the characteristics of the adaptive tweet filtering problems. In particular, we aim to answer the following research questions:

- **RQ1**: How effective are the traditional adaptive news filtering approaches (Incremental Rocchio and the state-of-the-art Regularised Logistic Regression) when applied to tweets?
- **RQ2**: Are our adaptations for tackling sparsity, using QE as described in Section 4, successful in improving filtering effectiveness?
- **RQ3**: Are our adaptations for tackling topic drift, as described in Section 5, successful in improving filtering effectiveness? In particular, this question can be addressed by answering three further related research questions: (a) Are the adhoc methods for defining the short-term interests sufficient when using our balancing approach to handle topic drift? (b) Is event detection useful in adjusting short term interests current emerging aspects/subtopics to aid filtering? (c) How sensitive is the filtering performance to the decay factor when using the balancing approach?

6.2 Experimental Setup

We use the real-time filtering task of the TREC 2012 Microblog track as our main testbed. The document collection used in this track is the Tweets2011 corpus, where the number of available tweets may change over time as users can delete their tweets or their accounts [24]. We use a filtered version from a list of tweetIDs provided by TREC for the 2012 tasks, which consists of 10,561,763 tweets in the period between 23 Jan and 8 Feb 2011. 49 topics are used in this task, which are split into 10 training and 39 testing topics.

In our experiments, we use Dirichlet language models [28] for weighting the terms in the vector representation as described in Section 3. To speed up the experiments, tweets that do not contain at least one query term are not considered for similarity computation and are regarded as irrelevant (we refer to this heuristic later as h1). We also experiment with relaxing this condition by considering tweets that contain at least one term in either the query or the first positive example (we refer to this heuristic later as h2).

Filtering Thresholds: The filtering threshold for Incremental Rocchio η_R we use is fixed, throughout the filtering process for all topics. It has been tuned for both official filtering measures of the TREC 2012 Microblog track ($T11SU$ and $F\_0.5$), using the ten training topics provided by TREC. The best value was then used on the testing topics. It should be noted that adapting the threshold has been investigated in news filtering [19] and a number of methods were proposed. In this paper, we do not look into this issue and we

leave for future work the exploration of methods for adapting the threshold over time. As for the threshold of the Regularised Logistic Regression approach η_L, an optimal value can be computed for particular filtering metrics directly as the logistic regression methods produce probabilistic scores. In particular, we use the optimal threshold for the $T11SU$ metric. The linear utility measure gives a credit of 2 whenever a relevant document is retrieved and penalises a system by 1 for a false positive [19], and therefore the value of the threshold used is η_L=0.33.

QE setup: For applying QE to extract expansion terms and tweets as described in Section 4, we used the Kullback Leibler divergence weighting model to rank terms in the pseudo-relevant set. The size of the pseudo-relevant set of tweet is chosen to be the one that achieved the best performance in previous experiments on real-time ad-hoc Twitter search [14]. In particular, $|T_q| = 20$, and the number of expansion terms used is $|E| = 10$.

Event detection: We apply the event detection approach using one of two different streams: the Tweets2011 stream (the tweets we are filtering) and a newswire stream we have collected in the same 16 day period of the Tweets2011 dataset. The newswire stream was crawled from major global news sources: BBC, CNN, Google News, New York Times, Guardian, Reuters, The Register and Wired. The crawl resulted in a total of 5,967 news articles (approximately 373 articles per day). The details of applying the event detection described in Section 5.3 are as follows. We use a sampling rate of $\theta = 10$ minutes, i.e. every 10 minutes at time t_i, we group the accumulated tweets or news articles in the last 10 minute time frame and consider them as a single candidate representing t_i. To score individual documents (tweets or news articles) within that candidate, for a topic (query), the weighting models employed were as follows: For tweets, we use the DFReeKLIM weighting model of the Divergence from Randomness framework designed particularly to rank short texts – DFReeKLIM was one of the most effective tweet ranking approaches submitted to the TREC 2011 Microblog track [4]; For the more conventional text of news articles, we use the BM25 weighting model. Regarding the Grubb's test parameters, we tuned the values of the window size of the Grubb's test k and the threshold z for both filtering measures using the training topics. In particular, k is set to 60 (60 * 10 minutes = 10 hours) and z is set to 5 standard deviations.

Finally, to conduct the experiments, we develop a stream processing infrastructure based on the open source Storm[3] framework. Within our infrastructure, we extend the Terrier IR platform [15] with real-time in-memory data structures. Performing the real-time filtering can be described as follows. Tweets are streamed from local disk. Each tweet is then passed to the rest of the Storm infrastructure where it is first preprocessed by removing stopwords and stemming using the English Porter stemmer. Then using the Terrier in-memory extension, the collection statistics are updated. Collections statistics include term frequencies, number of documents (tweets) accumulated, and term document frequencies. The filtering procedure is applied for each topic (when it is active) to make a binary decision (relevant or not) and update the profile of the topic if necessary. Note that the topic is active if the first positive example arrives, as this is considered that starting time of the filtering task. With this setup, we ensure that the task is conducted in a truly real-time manner.

[3]https://github.com/nathanmarz/storm

7. RESULTS AND DISCUSSION

In this section, we report and discuss the results of the various experiments we conducted under the general setup we described in Section 6. Our discussion aims to convey answers for the research questions of Section 6.1

7.1 Baseline News Filtering Approaches

The first four rows of Table 1 report the results of two variants of the standard Incremental Rocchio classification (RC) and the Regularised Logistic Regression (LR) approaches described in Section 3.2. For each combination, the second column specifies precisely how the method is instantiated. As discussed in Section 6.2, we introduce heuristics on considering a tweet for classification. The third column specifies the heuristic used. The table reports the set precision (set_prec), the set recall (set_recl) and the filtering measures (F_0.5, and T11SU). For the LR approaches, we report the best results achieved after tuning λ on the training topics for both filtering measures ($T11SU$ and $F\_0.5$). The best λ is then used for the testing topics and reported in the second column of Table 1. The key finding observed from the first four rows, which answers our research question (RQ1), is that the traditional news filtering approaches are not effective when applied to tweets. This is inferred from the low scores of the filtering measures in comparison with those of the TREC 2012 best run according to the official $T11SU$ measure (the last row of Table 1). Moreover, in the news filtering literature, Yang *et al.* [27] reported a $T11SU$ of 0.5715 for the state-of-the-art LR approach when applied on adaptive news filtering in the TREC 2002 Filtering track, which is much higher than the performances observed here. The scores are particularly low when relaxing the heuristic to (h2). However, in the second row, we observe an improvement in recall but that's on the expense of swamping the user with a very large number of false positives (set_prec is 0.0093). Finally, the state-of-the-art LR approach, which has been shown to outperform RC in the news filtering literature [27, 30], exhibits a poor performance and it is markedly outperformed by the RC approach. LR's poor performance on tweets can in fact be explained by the discriminative nature of LR in comparison to the generative model of Rocchio. LR needs more data to achieve a good performance. In a high dimensional space and with very few sparse learning examples, the LR model is not capable of correctly identifying relevant tweets. In addition, the highly dynamic nature of Twitter makes it harder for the LR model to converge over time to its optimal linear decision.

7.2 Sparsity

The rows (5 to 8) of Table 1 report the results of our adaptation of the RC approach to tackle sparsity using QE as described in Section 4. We observe a significant improvement in the overall filtering effectiveness. In fact, the best performing variant in terms of the two official filtering measures is the one which uses an enriched representation of the classifier's centroid using QE in row 8 of Table 1. It significantly improves the filtering effectiveness using either filtering measures over the original RC baseline approach of rows 3 and 4 in Table 1 according to a paired t-test ($p < 0.05$). Moreover, using the entire pseudo-relevant set instead of only expansion terms results in a better filtering effectiveness overall.

Table 1: Results obtained for the TREC 2012 Microblog track filtering task. ○ denotes a statistically significant increase over all other approaches. • denotes a statistically significant increase over all other variants apart from RC,h1. † denotes a statistically significant increase over all other variants apart from RC+Qe+Te,h1. ‡ denotes a statistically significant increase over all other variants apart from RC+Qe+Te,h2. Statistical significance is estimated with a paired t-test at ($p < 0.05$). Figures in bold correspond to the top performance. Results for the TREC best run are italicised and excluded from comparison and t-tests.

Approach	Properties	heuristic	set_prec	set_recl	F_0.5	T11SU
LR	(λ=0.05, μ=0)	h1	0.2306	0.1303	0.1227	0.2168
LR	(λ=0.05, μ=0)	h2	0.0093	**0.4061**○	0.0113	0.0070
RC	No QE. Equation (2)	h1	0.5508	0.1394	0.1915	0.3427
RC		h2	0.2057	0.1847	0.0904	0.1704
RC+Qe	QE. Using Equation (5)	h1	0.4940	0.1621	0.2095	0.3300
RC+Qe		h2	0.2299	0.2497	0.1032	0.1986
RC+Qe+Te	QE. Using Equation (6)	h1	**0.6127** •	0.1957	0.3361	**0.3985** ‡
RC+Qe+Te		h2	0.4206	0.3370	**0.3435** †	0.3615
TREC 2012 best			*0.6219*	*0.1740*	*0.3338*	*0.4117*

Table 2: Results obtained for the TREC 2012 Microblog track filtering task. Triangles denote increases (▲) or decreases (▼) compared to the baseline in the first row. Double triangles denote that the differences are statistically significant (paired t-test, $p < 0.05$). Figures in bold correspond to the top performance.

Classifier's centroid	parameters	heuristic	set_prec	set_recl	F_0.5	T11SU
QE using Equation (6)		h2	**0.4206**	0.3370	**0.3435**	**0.3615**
QE and handling drift using Equation (8). (A) Arbitrary adjustments of S_t	($\sigma = 0.3$, $n = 1$)	h2	0.3896 ▼	0.3485 ▲	0.3314 ▼	0.3472▼
	($\sigma = 0.3$, $n = 3$)	h2	0.3617 ▼ ▼	0.3514 ▲	0.3086 ▼ ▼	0.3265▼
	($\sigma = 0.7$, $n = 1$)	h2	0.3086 ▼ ▼	0.3445 ▲ ▲	0.2662 ▼ ▼	0.2773 ▼ ▼
	($\sigma = 0.7$, $n = 3$)	h2	0.3044 ▼ ▼	**0.3705** ▲ ▲	0.2620 ▼ ▼	0.2673 ▼ ▼
QE and handling drift using Equation (8). (B) Daily adjustment of S_t	($\sigma = 0.3$)	h2	0.3789 ▼	0.3230 ▼	0.3112 ▼	0.3372 ▼
	($\sigma = 0.7$)	h2	0.2833 ▼ ▼	0.3374 ▼	0.2403 ▼ ▼	0.2557 ▼ ▼

In addition, our adaptation with QE outperforms the best TREC 2012 run using the $F\_0.5$ measure. However, the utility achieved $T11SU$ in row 8 of Table 1 is lower than the best TREC 2012 run that seems to be a conservative approach with a high precision and a low recall unlike ours, where we achieve a better balance. As a summary, and in answer to our research question RQ2, our QE modification tackle the sparsity of the tweets is in fact effective and improves the filtering quality over the standard RC baseline.

7.3 Handling Topic Drift

We conduct a number of experiments to evaluate our adaptation of Incremental Rocchio to handle topic drift during filtering that we introduced in Section 5. As a baseline, we use our best performing variant in the previous experiment (the last row in Table 1), which applies QE to tackle sparsity and applies the heuristic (h2). We then experiment with a number of methods to apply the balancing approach of the centroid specified in Equation (7), while using the same heuristic (h2) of the baseline. As discussed in Section 5.2, defining the tweet set S_t, the set of the relevant tweets representing the short-term interests at a certain time t, can be performed by two adhoc intuitive methods: (A) arbitrary adjustments or (B) daily adjustments. Using the 10 training topics, we tune the arbitrary number of recent tweets n of method (A) and the decay factor σ for both filtering measures ($T11SU$ and $F\_0.5$). The best values are then used on the testing topics. For method (A) of defining S_t (Arbitrary adjustments), the best performance is achieved for the combination ($\sigma = 0.3$, $n = 1$) on the training topics, but nevertheless we report other combinations in Table 2 to illustrate the effect of each parameter. Although there was a slight improvement on the training topics for ($\sigma = 0.3$, $n = 1$), the filtering performance using both filtering measures degrades on the testing topics however only marginally. Also, we can observe an increase in the recall over the baseline, but not statistically significant according to a paired t-test (p<0.05). In fact, with a higher decay of long-term interests, we can achieve a significant increase in recall but on the cost of the precision and therefore a significantly worse filtering effectiveness in both filtering metrics (rows 4 and 5 of Table 2). For method (B) of defining S_t (Daily adjustments), we observe negative results in Table 2. The best performance on the training topics was achieved when ($\sigma = 0.3$) and it yielded marginal improvement but this is not the case on the testing topics. Indeed, in this case, even the recall does not improve and the filtering performance degrades marginally when $\sigma = 0.3$ and significantly when $\sigma = 0.7$. This indicates that using daily intervals to adjust short-term interests does not handle topic drift. In answer to our research question RQ3(a), we conclude that the adhoc methods for capturing short-term interests may not be useful for our balancing approach for topic drift, as it fails to improve the filtering quality when these methods are employed.

7.4 Using Event Detection

We next experiment with our automatic approach for adjusting the tweets set S_t, as described in Section 5.3, to aid our balancing approach of handling topic drift. As in Section 2, we tune the decay factor σ for both filtering measures using the training topics and use the best value on the testing topics. However, we will provide a comprehensive sensitivity analysis of this parameter in the next section. The results are reported in Table 3. When using the Tweets11

stream for event detection (second row of Table 3), as in Section 2, the recall increased on the expense of precision and both filtering measures. However, the main difference is that we see a statistically significant increase in recall over the baseline but not a statistically significant decrease on either filtering metrics according to a paired t-test ($p < 0.05$). This is an interesting finding and it shows the usefulness of our event detection to aid the balancing approach. Shifting the centroid to the short-term interests results in more false positives, which degrades precision. However, event detection helps to better represent the short-term interests over time, thereby reducing the number of false positives, while at the same time identifying more relevant tweets, thereby enhancing the recall. We perform a further analysis where we examine the differences in recall with the baseline (first row of Table 3) across all the testing topics. We plot these differences in Figure 2. We observe that the increase in recall is fairly consistent across all topics. Only on a single occasion, we see a slight decrease in recall, while for all other topics we see either an increase or no difference at all. When using an external newswire stream for detecting events (last row of Table 3), the same pattern is observed. If we compare this approach to the one that uses the Tweets11 stream (the row above), we observe that the differences are marginal. This may indicate that the events detected with either streams have a large overlap, which has been shown in a recent study [17]. As a summary and in answer to research question RQ3(b), we find that the event detection method, using either the Tweets11 stream itself or the external newswire stream, helps our balancing approach to tackle drift. In particular, it results in significantly improving recall while only marginally decreasing the filtering performance.

7.5 Decay Sensitivity Analysis

To answer research question RQ3(c), we conduct a sensitivity analysis of the decay factor for the approach specified in the second row of Table 3. We vary the decay factor σ between 0.0 (only long-term interests; a baseline equivalent to the the first row in Table 3) and 1.0 (only short-term interests) increasing it by 0.1 at a time. Figure 3 plots the changes in the various evaluation measures. As the decay increases, we first observe a marginal increase on all measures apart from F_0.5. Later, we observe that the recall starts to improve but the precision degrades. At σ=0.4, the improvement in recall is statistically significant using a paired t-test (p<0.05) while the decrease in both filtering measures and in precision is statistically insignificant. After this point, recall continues to improve and stabilises between σ=0.6 and σ=0.8. However, precision and both filtering measures fall sharply in this region. At σ=1.0, when the classifier puts all the emphasis on the latest tweets only, we observe a sharp decrease in recall and on filtering measures and a slight increase in precision. This is possibly due to the fact that the S_t set may be empty at certain times (just after resetting due to a detection of an event), which may reduce the false positives picked up by the L_t component of the centroid when σ<1.0. This analysis uncovers a typical problem in IF and IR, where there is a trade-off between recall and precision. More importantly, it shows that achieving a good filtering effectiveness is particularly challenging because it has a higher sensitivity to precision than to recall. An effective filtering technique should strive to achieve the best trade-off between the two measures. Finally, as a summary and in answer to the overall research question RQ3, our approach to handle topic drift works best when aided with event detection to capture potential times of topic drift. It significantly improves recall at the cost of a marginal decrease in the overall filtering performance.

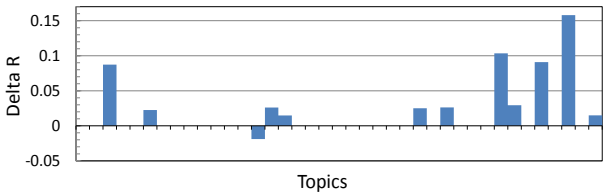

Figure 2: Changes in recall across all the testing topics between the approach in the first row and the one in the second row of Table 3.

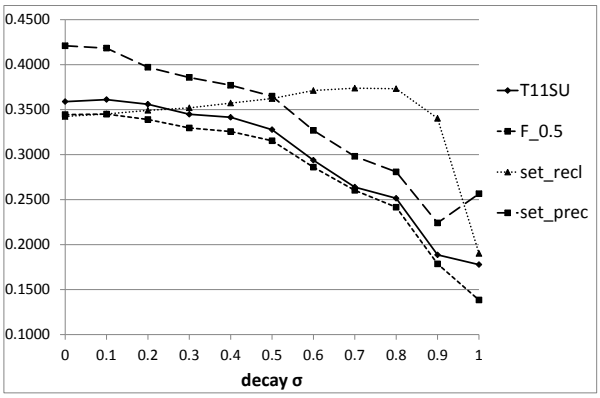

Figure 3: Sensitivity analysis of the decay factor σ when using the balancing approach with adjustment aided by event detection (the approach in the second row of Table 3).

8. CONCLUSIONS AND FUTURE WORK

Real-time filtering of tweets is an emerging information filtering task that can still be tackled using traditional filtering techniques with appropriate adaptations to address the unique challenges prevalent in Twitter. Our thorough evaluation, using a standard TREC collection, shows that traditional state-of-the-art news filtering techniques are not as effective when applied on tweets. However, the modifications we proposed on a traditional news filtering approach are shown to mitigate the acute sparsity issue that is prevalent in Twitter. In particular, our query expansion approach to tackle the sparsity of tweets yields a significant improvement in filtering effectiveness, by deriving a richer representation of the user profile with relevant and timely terms. Moreover, we introduced another adaptation to tackle topic drifting during filtering, which can be amplified by the highly dynamic nature of Twitter. The results show that by using event-detection to balance between short-term and long-term interests, we can significantly improve the filtering recall while only marginally harming the filtering utility.

There is plenty of scope for future work. First, to tackle the sparsity issue, we aim to explore the use of dynamic knowledge resources, such as Wikipedia, in a timely manner to complement our query expansion approach. This may allow the classifier, for example, to capture terms that are not picked up by query expansion, and hence obtain a richer representation of the user's information needs. Secondly, the TREC dataset has the limitation that explicit user judgments are simulated and may not necessarily reflect realistic

Table 3: Results obtained for the TREC 2012 Microblog track filtering task. Triangles denote increases (▲) or decreases (▼) compared to the baseline in the first row. Double triangles denote that the differences are statistically significant (paired t-test, $p < 0.05$). Figures in bold correspond to the top performance.

Classifier's centroid	parameters	heuristic	set_prec	set_recl	F_0.5	T11SU
QE using Equation (6)		h2	**0.4206**	0.3370	**0.3435**	**0.3615**
QE and handling drift using Equation (8) with event detection to adjust S_t	(Tweets11, σ=0.4, k= 10hrs, z= 5)	h2	0.3771 ▼	0.3573 ▲ ▲	0.3256 ▼	0.3415 ▼
	(Newswire, σ=0.4, k= 10hrs, z= 5)	h2	0.3724 ▼	**0.3598** ▲ ▲	0.3198 ▼	0.3351 ▼

feedback. Therefore, we aim to conduct an online evaluation, in the form of A/B testing, where we can obtain explicit feedback from real users.

Acknowledgements

This work has been carried out in the scope of the EC cofunded project SMART (FP7-287583).

9. REFERENCES

[1] M. Albakour, C. Macdonald, I. Ounis. Identifying Local Events by Using Microblogs as Social Sensors. In *Proc. of OAIR*, 2013.

[2] J. Allan. Incremental relevance feedback for information filtering. In *Proc. of SIGIR*, 1996.

[3] G. Amati. *Probability models for IR based on Divergence From Randomness*. PhD thesis, Univ. of Glasgow, 2003.

[4] G. Amati, G. Amodeo, M. Bianchi, G. Marcone, C. Gaibisso, A. Celi, C. De Nicola, and M. Flammini. FUB, IASI-CNR, UNIVAQ at TREC 2011. In *Proc. of TREC*, 2011.

[5] N.J. Belkin, and W.B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[6] J. Callan. Learning while filtering documents. In *Proc. of SIGIR*, 1998.

[7] G. Cormack, J. M. G. Hidalgo, E. P. Sánz, Spam filtering for short messages. In *Proc. of CIKM*, 2007.

[8] F. Grubb. Procedures for detecting outlying observations in samples. *Technometrics*, 11, 1969.

[9] L. Horváth. The maximum likelihood method for testing changes in the parameters of normal observations. *Annals of statistics*, 21(2), 1993.

[10] L. B. Jabeur, L. Tamine, and M. Boughanem. Uprising microblogs: a bayesian network retrieval model for tweet search. In *Proc. of SIGIR*, 2012.

[11] X. Hu, N. Sun, C. Zhang, and T.-S. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *Proc. of CIKM*, 2009.

[12] B. Krishnamurthy, P. Gill, and M. Arlitt. A few chirps about Twitter. In *Proc. of WOSN*, 2008.

[13] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proc. of CIKM*, 2006.

[14] R. M. C. McCreadie, C, Macdonald, R, L. T. Santos and I. Ounis. University of Glasgow at TREC 2011: Experiments with Terrier in Crowdsourcing, Microblog, and Web Tracks. In *Proc. of TREC*, 2011.

[15] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proc. of OSIR at SIGIR*, 2006.

[16] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the TREC-2011 Microblog Track. In *Proc. of TREC*, 2011.

[17] S. Petrovic, M. Osborne, R. McCreadie, C. Macdonald, I. Ounis, and L. Shrimpton. Can Twitter replace Newswire for breaking news? In *Proc. of ICWSM*, 2013.

[18] S. Robertson. Threshold setting and performance optimization in adaptive filtering. *Information Retrieval* 5(2):239–256, 2002.

[19] S. E. Robertson and I. Soboroff. The TREC 2002 Filtering track report. In *Proc. of TREC*, 2002.

[20] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Information Retrieval*, 3(4), 2009.

[21] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System*, 313–323, 1971.

[22] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proc. of WWW*, 2010.

[23] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in Twitter to improve information filtering. In *Proc. of SIGIR*, 2010.

[24] I. Soboroff, D. McCullough, J. Lin, C. Macdonald, I. Ounis and R. McCreadie. Evaluating real-time search over Tweets. In *Proc. of ICWSM*, 2012.

[25] T. Wasserman. Twitter says it has 140 million users. URL: *http://mashable.com/2012/03/21/twitter-has-140-million-users/*, March, 2012.

[26] J. Xu and W.B. Croft Query expansion using local and global document analysis. In *Proc. of SIGIR*, 1996.

[27] Y. Yang, S. Yoo, J. Zhang, B. Kisiel. Robustness of adaptive filtering methods in a cross-benchmark evaluation. In *Proc. of SIGIR*, 2005.

[28] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *Transactions on Information Systems*, 22(2):179–214, 2004.

[29] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proc. of ICML*, 2004.

[30] Y. Zhang. Using bayesian priors to combine classifiers for adaptive filtering. In *Proc. of SIGIR*, 2004.

[31] A. Zubiaga, D. Spina, V. Fresno, and R. Martínez. Classifying trending topics: a typology of conversation triggers on Twitter. In *Proc. of CIKM*, 2011.