Expert-Driven Topical Classification of Short Message Streams

Krishna Y. Kamath
Department of Computer Science
Texas A&M University
College Station, Texas 77843
Email: kykamath@cs.tamu.edu

James Caverlee
Department of Computer Science
Texas A&M University
College Station, Texas 77843
Email: caverlee@cse.tamu.edu

***Abstract*—We study the problem of expert-driven topical classification of short messages in time-evolving streams like Facebook status updates, Twitter messages, and SMS communication. While high-level topics in these streams may be fixed (e.g., Sports, News), the content associated with these topics is typically less static, reflecting temporal change in interest as these streams evolve (e.g., tweets about the Olympics wane, while tweets about the World Cup rise in popularity). Coupled with this rapid concept drift, short messages themselves provide little contextual information and result in sparse features for effective classification. With these challenges in mind, we present an expert-driven framework for time-aware topical classification framework of short messages. Three of the salient features of the framework are (i) a novel expert-centric classifier; (ii) a sliding-window training for adaptive topical classification; and (iii) a suite of enrichment-based methods (lexical, link, collocation) for overcoming feature sparsity in short messages.**

I. Introduction

The past few years have seen the rise of massive-scale social messaging systems that support the rapid exchange of short messages, near instantaneous global reach, and unprecedented leveraging of massive-scale interpersonal connections. Prominent examples include the popular Twitter micro-blogging service, which boasts 90 million tweets per day (as of September 2010) [1], and Facebook, which supports the sharing of brief status updates with friends and acquaintances. These short status updates, as well as short text comments, forum postings, and other manifestations of emerging text-based participatory sensing streams provide new opportunities in the areas of web search, advertising, personalized information services, recommendation engines, and so on. Some recent examples include Blogscope, a system to detect events on blogosphere [2], a real-time system to detect earth quakes using twitter [15], methods to measure public opinions [12], and a system to predict future ratings for YouTube videos based on comment and rating history [17].

One of the key challenges for making sense of these high-volume short message streams is in organizing these *unstructured* social streams into *structured* categories of interest. For example, several recent efforts have begun the study of Twitter message classification in the context of information filtering [19], news aggregation [16] and business specific mining [21]. In many of these cases, however, mapping from unstructured social streams to structured categories of interest may lead to errors and poor quality identification of relevant messages due to a number of key challenges:

- The rapid evolution of social streams, so that important keywords associated with a concept one day may not correspond to the same concepts the next day. To illustrate, Figure 1 shows how the prevalence of the keyword "healthcare" varied on Twitter across several categories during the healthcare debate (details described later in the paper). Note that during the month of March (weeks 9 to 12) the Senate was debating the healthcare bill leading to many mentions of "healthcare" in politics; at other times, "healthcare" was associated with business-related messages and of course, health-related messages.
- The inherent error-laden and lack of context in many messaging systems that restrict the number of characters (140 characters, in the case of Twitter). As an example, consider the message – "Almst over da Flu..stayin in all weeknd" – which contains shortened words and misspellings.
- A mismatch between the language in use by participants and the language expected by the classification framewrok (e.g., the use of emergent hashtags, colloquialisms) as in an example tweet describing an earthquake "Ahh!! :S tremble. Walls cracking!! #timetoleave".

Together, this coupling of rapid concept drift, lack of contextual information, and sparse feature representation present strong challenges to effective and ongoing topical classification of short message streams. With these challenges in mind, we present an expert-driven framework for time-aware topical classification framework of short messages. The key insight driving the framework is the reliance on category-specific *experts*, whose streams themselves may serve as prototypes for learning generalized categorical models for robust stream classification. We show how these expert streams may seed classification, and we propose a sliding-window training approach for adaptive topical classification. Additionally, we explore techniques for augmenting short messages using feature-based, link-based and collocation expansion. Through experimental study over Twitter, we find good performance of the proposed method for ongoing expert-driven topical classification of short message streams.

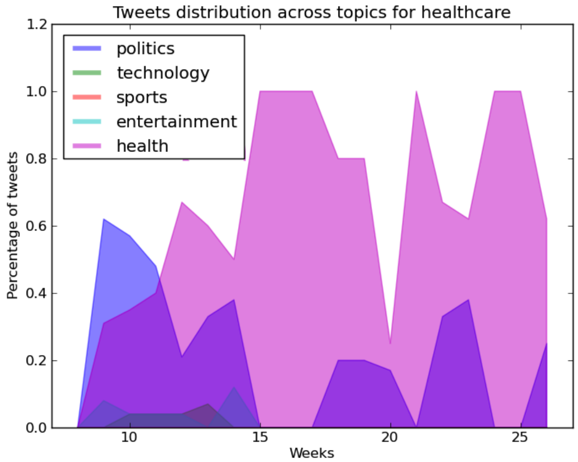


Fig. 1. Prevalence of the term 'healthcare' across domains from March 2010 to July 2010.

II. Problem Statement and Setup

In this section we present the overall framework of our study of expert-driven topical classification over short message streams. We begin with a discussion of the problem, and then introduce the data and baseline classifier used in the rest of the paper.

A. Problem Statement

While a domain model may identify an arbitrarily complex concept hierarchy, we focus in this paper on a simple one-level hierarchy corresponding to general high-level topical categories. We selected four high-level categories for this study that are generally well-represented in current popular social messaging systems: *politics*, *technology*, *sports* and *entertainment*. For each category, the system takes as input a set of *expert* accounts and their messages. These experts are intended to be representatives of the category, although not all of their messages may actually belong to a single category. For example, a sports-themed account may intersperse entertainment and politics messages in their stream of mostly sports-related messages. In practice we will only be able to identify a small number of expert accounts relative to the large body of actual accounts in a system. Given a set of categories and a list of expert accounts, we seek to identify messages over time that map to these categories. We refer to this as the problem of *expert-driven topical classification of short messages in time-evolving streams*.

B. Data

For this study, we require a collection of time-stamped short messages from across a number of different categories. While there are large benchmark collections of Web pages, email messages, and other longer-form documents, we are unaware of any existing topically-segmented short message collections. Hence, we collect a "ground-truth" domain-specific Twitter stream by identifying prominent accounts for the 4 domains

TABLE I
Data Distribution Per Domain

Domain	Total Messages	Messages per day
Politics	30,658	143
Technology	21,880	102
Sports	67,782	316
Entertainment	38,477	179

– technology, entertainment, politics and sports – using a snowball sampling approach described in [20]. The output of this snowball sampling method is for each category an ordered list of accounts, ordered by their significance within that category (the details are omitted here, but explained more fully in [20]). Then for each domain we select the top 1,250 accounts and use the "follow" parameter of the filter method from Streaming API to generate a domain specific stream of tweets. Based on this method, we collected a total of 209,046 messages between March and April 2011. The breakdown per domain is shown in Table I.

C. Topical Classification with MaxEnt

Given a message from a social messaging system, we aim to automatically determine its appropriate category through an analysis of the text in the message itself. While many text classifiers are possible (e.g., Naive Bayes, Support Vector Machines), we focus in this paper on maximum entropy (MaxEnt) classification [11], which has been shown to efficiently model domains in which information is sparse (as in the case of short messages). MaxEnt is based on the maximum entropy principle [3] and has been widely used for text classification [13]. We will now describe the maximum entropy principle in terms of text classification.

Consider a document (short message) d that belongs to class y in a training set of labeled documents. Generally, in text classification, terms in the documents are represented as features. So, let x be a term in d. Then we can define a feature function $f(x,y)$ as an indicator random variable.

$$f(x,y) = \begin{cases} 1 & If\ x\ is\ in\ document\ of\ class\ y \\ 0 & Otherwise \end{cases}$$

From the training set we can calculate the empirical probability distribution $\tilde{p}(x,y)$ of observing x in documents of class y. Using this we can determine the empirical expected value of f.

$$\tilde{p}(f) = \sum_{x,y} \tilde{p}(x,y) f(x,y)$$

When the ideal classification model $p(y|x)$ is known, we can use the empirical distribution of x, $\tilde{p}(x)$ (calculated from the training set), to determine the expected value of f as:

$$p(f) = \sum_{x,y} \tilde{p}(x) p(y|x) f(x,y)$$

Now, given a set of feature functions $F = \{f_1, f_2, \ldots, f_n\}$, one for every term, and the space of all probability distributions P we can define $C \subset P$, as the set of distributions which give the same expected value of f as the empirical value of f obtained from the training set.

$$C \equiv \{p \in P|\ p(f_i) = \tilde{p}(f_i) \quad for\ i \in \{1, 2, \ldots n\}\}$$

Of all the models (distributions) in C, we have to pick the model that gives the most uniform distribution. Hence, we can use conditional entropy $H(p)$ to optimize the solution.

$$H(p) \equiv -\sum_{x,y} \tilde{p}(x)p(y|x)logp(y|x)$$

The maximum entropy principle states to pick the the distribution $p_\star \in C$ that yields the maximum entropy $H(x)$:

$$p_\star = \arg\max_{p \in C} H(x)$$

For text classification, $p_\star$ gives us the model from which the probability that document d, which contains a term x, belongs to a class y can be calculated using $p(y|x)$. In this way, we can assign short messages to one of the four categories.

III. Overall Expert-Driven Approach

Toward bridging the gap between unstructured social messaging streams and structured categories of interest, we must first identify a set of candidate expert accounts associated with each category – these expert accounts serve as prototypes of what we expect to see from a particular category. While the particular expert-selection criteria may vary across domains and application setting, we adopt a baseline approach where we select as *experts* the top-125 accounts in each domain as ordered by the snowball sampling approach described in Section II-B.

A. Sliding Window Training

Given an appropriate selection of expert streams, to abate the effects of rapid concept drift we propose to train a classifier over a *sliding window* to capture the day-to-day and hour-to-hour changes in the concepts associated with a particular category. Using a fixed period of days, we could monitor all messages posted by the expert accounts, build a classification model based on these messages, and then classify all new messages based on this model. For example, if today is the 11^{th} day of March, then we could build a classifier over the prior eight days of messages posted by the pre-seeded expert accounts (from 3^{rd} March to 10^{th} March) and apply this new model to all messages encountered. Moving to the following day, the classification models could be updated with the sliding window (now from 4^{th} March to 11^{th} March), and so on and so on. In this way, the classification decisions are based primarily on concepts that are recently reflected in the social messaging system, rather than being tied to immutable keywords.

Of course, there are a number of open questions: (i) What is the best size of a sliding window? Choosing a very small window may perform well on bursty events within a category (e.g., a particular football game within the domain "sports"), but more poorly on longer-lived themes. (ii) Is there enough feature density (i.e. keywords) in each expert stream to produce robust topical classifiers? (iii) How can this feature sparsity be overcome in a lightweight manner?

IV. Short Message Enrichment

Even with a dynamic sliding window classifier in place, we still face one of the key challenges to content-based classification of short messages – the problem of limited features found in these messages. Whereas traditional web page and document classification tasks have typically focused on feature selection for reducing the many available word-based features to identify a smaller set of highly-valuable distinguishing features, in short message classification we take an alternate approach to enrich the sparse messages with additional features. Concretely, we explore three general approaches for overcoming feature sparsity in short messages: (i) lexical-based, in which features in short messages are increased by applying lexical feature expansion techniques based on the content within the message; (ii) external-based, in which externally-derived features like part-of-speech and URL features extracted from links embedded in short messages are used to augment the feature representation; and (iii) collocation-based, in which the terms in a message are associated with related terms (collocations) from other messages, and these related terms are added as features to the original message.

A. Lexical-Based Enrichment

To overcome the sparsity of feature set in short message classification we can use lexical feature expansion techniques. We use bigrams, trigrams and orthogonal sparse bigrams to increase features. The details of these techniques are given below:

- *Character n-grams*: Using this technique n consecutive characters in the message are used as features.
- *Word n-grams*: Similar to character n-grams, in this technique n consecutive words in the original message are used as features.
- *Orthogonal Sparse Word Bigrams:* Following Cormack [4], this technique generates as features every pair of words that are separated by 3 or fewer words.

B. External-Based Enrichment

In this approach we overcome feature sparsity in short messages by augmenting each message with features extracted from an external resource. Specifically, we consider two approaches: link-based and part-of-speech-based.

- *Link-based*: Short messages often contain URLs in them and in many cases an individual URL linking to a webpage contains information that describes the page. We call this information collected from the raw URLs the link meta information.

  For example, consider the URL:

  http://www.nytimes.com/2010/07/27/sports/football/27concussion.html?ref=sports

By just reading the URL we can understand that the page is a sports page about football that talks about concussions. We can extract the terms sports and football from the URL and enrich the short message with it. For URLs that are shortened using service like bit.ly, goo.gl etc., we expand the actual link pointed by the shortened URL and extract meta information from the long-form URL.

- *Part-of-speech*: In a given short message, identifying nouns can give us a good understanding of the message topic. So, in our analysis we tagged terms in a message with their corresponding part-of-speech (POS). We used the POS tagging feature in NLTK Python toolkit [9] and filtered words which were not tagged as nouns.

C. Collocation-Based Enrichment

The third expansion technique considers words that are associated with the words in a message. We identify associated words by examining collocations from across other expert accounts within a category and from what we refer to as "affiliate" accounts (described more fully in the experiments section). A collocation is "an expression consisting of two or more words that correspond to some conventional way of saying things" [10]. Examples of collocations are *kobe bryant, boston celtics*, etc. Intuitively, a short message may refer to some aspect of a concept (e.g., "kobe"), but due to the space limitation may not include other related terms (e.g., "bryant", "lakers"). By identifying collocations, we can enrich a single message with additional terms, but perhaps at the cost of introducing noise terms.

Concretely, we limit ourselves in this paper to collocations consisting of two words only. To identify collocations we first need an association measure between words. Association measures, are mathematical formulae, used to measure the closeness between the words of a phrase. This measure is used to rank the pair of words. The measure is based on the count of occurrences of words and co-occurrences between pairs of words. There are various association measures starting from plain frequency of occurrence, to measures based on information theory like mutual information and heuristic methods.

In [10], the authors have illustrated the problems with association measures that use frequency or variance to determine collocations. They also show mutual information is not very suitable to identify collocations. Hence, to determine collocations in this paper we will be using two asymptotic hypothesis test methods: *Pearson's chi-squared test* (χ^2) and *Dunning's log-likelihood ratio test*. Generally it is observed that the log-likelihood test is more useful in determining collocations on sparse data compared to the χ^2 test.

V. Experimental Study

In this section, we present a comparative study of the time-aware topical classification framework for short messages. We use the dataset of 209,046 messages across four categories, collected during March-April, 2011. Using the top-125 accounts per domain as the seed experts, we test the developed topical classifiers over a test set consisting of the *bottom* 125 accounts per domain (out of 1,250), meaning that these test accounts are only loosely-related to the categories of interest and non-overlapping with the expert accounts.

A. Metrics

To evaluate the quality of a topical classifier over short message streams, we consider a variation of the area under the Receiver Operating Characteristic (ROC) curve called the *M-value*.

M-value: The area under Receiver Operating Characteristic (ROC) curve is a widely-used metric to measure the performance of classifiers. But, it is appropriate only in binary classification problems and hence cannot be directly applied to multi-class classification problems. So, in this paper, since we are dealing with a multi-class classification problem, we use a metric which is an generalization of the ROC metric used in binary classification. We use the popular *M-value* metric proposed by Hand and Till [6], that extends the area under the curve definition to the case of more than two classes by averaging pairwise comparisons.

Given a set of classes $C = \{c_1, c_2 \dots c_k |\ k > 2\}$ and a document d in the test set, the classification algorithm gives us an estimate of the probability of the document belonging to any class c, $P(c|d)\ \forall c \in C$. Given this we can calculate $\hat{A}(i|j)\ \forall i,j \in C$. $\hat{A}(i|j)$ is defined as the probability that a randomly drawn member of class j will have a lower estimated probability of belonging to class i than a randomly drawn member of class i. Similarly, we can calculate the value of $\hat{A}(j|i)$ as well. Note that in case of binary classifiers $\hat{A}(0|1) = \hat{A}(1|0)$, while this is not true in the case of multi-class classifiers, i.e. $\hat{A}(0|1) \neq \hat{A}(1|0)$. We then calculate the separability between any two classes as $\hat{A}(i,j) = [\hat{A}(i|j) + \hat{A}(j|i)]/2$.

The overall separability for all the classes – the M-value – is given by the average of all the values of $\hat{A}(i,j)$:

$$M = \frac{1}{\binom{|C|}{2}} \sum_{i<j} \hat{A}(i,j)$$

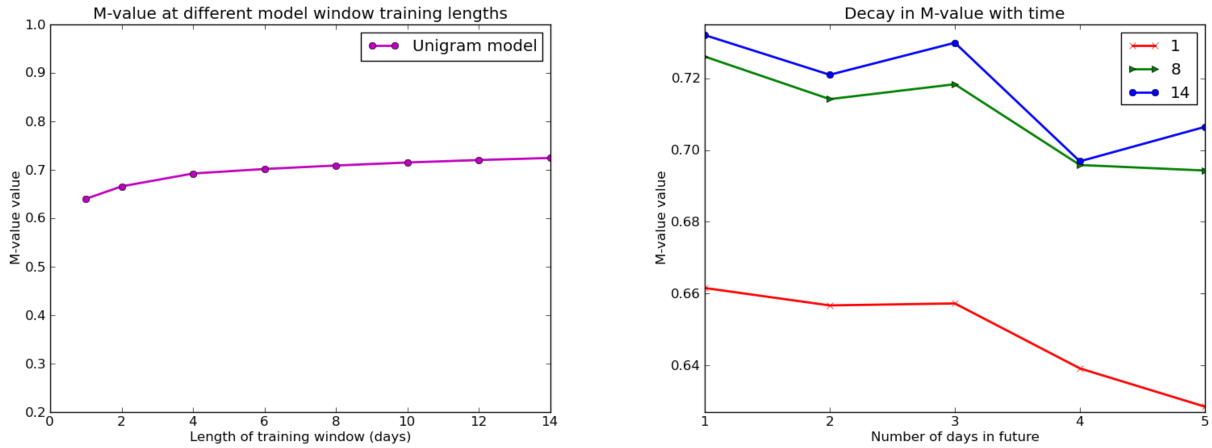
A higher M-value indicates a "better" classifier.

B. Sliding Window Length

We begin the experimental study by examining how the size of the training window impacts the quality of categorization. We first try different window lengths and observe the length at which the M-value is maximum. We then use the models trained on different window lengths to see how they perform over time.

Different Window Lengths: The sliding window approach we advocate requires that we identify a set of expert accounts to serve as our prototypes for each category. For fairness, we train on the messages in the gold set for days leading up to but not including the test day of messages.

We trained a MaxEnt classifiers, using unigram features, on different training-windows to determine the optimum length. In Figure 2(a), we show the performance of the classifiers that were trained on a window length from 1 to 14 days. We



(a) M-value at different training-window sizes.

(b) Diminishing M-value of the classifiers with time at different model lengths.

Fig. 2. Comparing Training Window Sizes and Classifier Effectiveness Over Time.

observe that the M-value is lowest with only a single day of training; this indicates that the concepts introduced on a single day are not representative of the overall theme. The curve flattens around the 8^{th} day, indicating that about a week's worth of messages are necessary to capture the main concepts. We also notice that classifier that is trained for around 8 days yields almost the same accuracy as a classifier trained for 14 days, indicating that longer window sizes do not necessarily lead to large gain in classification accuracy.

Classifier Decay: We next investigate how long after a classifier has been built it is still effective. We know that as new messages are observed and newer concepts introduced the accuracy of an older classifier decreases. We refer to this decrease in M-value of a classifier with time as classifier decay. A good classifier should decay relatively slowly, meaning that the essential characteristics of a category have been learned. To analyze classifier decay, we took classifiers that were trained on 1, 8 and 14 day windows, and used them to classify tweets. This is shown in Figure 2(b). We observe that the classifier trained on a 14 days window decays slowly.

The difference in decay can be attributed to the features that these different classifiers learn from the training set. A classifier trained for 14 days learns concepts that are spread over a longer duration of time while the 8 day classifier picks up concepts that occur for a shorter time. For example, a 14 day classifier may learn features related to MLB games, an event that happens over months, and not learn relatively short events like individual games during March madness, which happens on a single day. But the 8 day classifier is able to learn these events of shorter durations.

C. Short Message Enrichment

Based on the results in the previous section, we next evaluate the several approaches to short message enrichment where each classifier has been trained over an 8 day window. We begin by testing the performance of lexical feature expansion, as shown in Table II. First, we can see that the unigram gives the best performance of all the lexical approaches. Hence, from now on for all the experiments we will be unigrams as features for classification.

To test the performance of the classifiers with collocation-based expansion, we append the messages in the training set with the collocations discovered using χ^2 and Dunning's log-likelihood. We use two different collections of messages to identify collocations: (i) Messages from "experts": Top 125 accounts per domain (500 accounts); and (ii) Messages from "affiliates": Top 375 accounts per domain (1500 accounts). The affiliates are accounts outside of the top 125 accounts. The hypothesis is that by enriching messages with collocations from affiliate accounts, we may identify extra category-specific collocation terms that cannot be obtained from experts. For all four cases, the performance of the classifiers is shown in Table III. Interestingly, we note that the performance is improved by using collocations obtained from affiliates.

We next test the two external enrichment approaches – part-of-speech tagging and link expansion. We see in Table IV that the noun-based approach results in a smaller M-value than the unigram approach, indicating that the key distinguishing features for topical classification are most likely to be unigrams. Also, when introducing link-based information in spite of additional features we don't see any improvement in performance. This is quite encouraging, since extracting nouns and link information is expensive in a real-time application, and the result that unigrams can yield the best performance can motivate efficient classification algorithms.

VI. Related Work

Short text classification in the context of spam filtering has been discussed in several papers. Most of the work deals with short text in the form of mobile communications (SMS), blog comments, email summaries, etc. Cormack et.al. [4] examined

TABLE II
LEXICAL FEATURE EXPANSION

Description	M-value
Unigrams	0.71
Character bigrams	0.67
Bigrams	0.49
Orthogonal sparse bigrams	0.54

TABLE III
COLLOCATION-BASED EXPANSION

Description	M-value
χ^2 (experts)	0.70
χ^2 (affiliates)	**0.74**
Dunning's (experts)	0.69
Dunning's (affiliates)	0.69

TABLE IV
LINK AND POS-BASED EXPANSION

Description	M-value
Unigrams	0.70
Nouns	0.66
Unigrams + link	0.71
Nouns + link	0.66

several lexical feature expansion techniques for classifying short messages as spam or not. In [18] the authors model the style in which short texts are written to filter spam. They utilize features like the length of the short text and part-of-speech n-grams to build classifiers that identify spam. Other techniques like bayesian filtering [5] and the use of an external dataset [13] have also been found suitable for short text classification. In [13], the authors use an external large source of words like Wikipedia to compensate for the lack of features in short text. Though all of these papers concentrate on the problem of sparsity in short texts, there is no notion of time associated with the classifier as in our study.

Concept mining over temporal streams of data is used in areas like news classification [7] and email spam filtering [8]. One of the common approaches is the use of an ensemble of classifiers to track concept drift. In [14] the authors use an ensemble of decision tree classifiers trained on sequential data chunks. They then select appropriate classifiers depending on the data they are trying to classify. In [8] the authors develop an ensemble of classifiers to identify recurring concepts in an online stream of data for email filtering. They describe a system to sort out recurring concepts and then train the classifiers to learn these concepts. Katakis et al. in [7] develop a system which manages concept drift in news articles to provide a personalized new dissemination system. Differing from the ensemble approach they develop an instance of an incremental classifier based on naive bayes that updates every time it receives a new news article. These techniques are generally used in domains where the data is online but not sparse.

VII. CONCLUSION

In this paper we studied the problem of culling messages from time-evolving short message streams that correspond to pre-defined areas of interest. We have proposed and evaluated an expert-driven sliding window approach for classifier training in order to capture the day-to-day and hour-to-hour changes in the concepts associated with a particular category. We explored three general approaches for overcoming feature sparsity in short messages: (i) lexical-based; (ii) link-based; and (iii) collocation-based. We are encouraged by our initial results. As future work we are interested to adapt the time-sensitive classification framework to finer-grained time slices (e.g., hours, minutes) and to investigate per-account classification rather then per-message classification.

REFERENCES

[1] Techcruch article. http://techcrunch.com/2010/09/14/twitter-seeing-90-million-tweets-per-day/.

[2] N. Bansal and N. Koudas. Blogscope: spatio-temporal analysis of the blogosphere. WWW '07, pages 1269–1270, New York, NY, USA, 2007. ACM.

[3] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to nlp. *Comput. Linguist.*, 22(1):39–71, 1996.

[4] G. V. Cormack, J. M. Gómez Hidalgo, and E. P. Sánz. Spam filtering for short messages. In *CIKM '07*, pages 313–320, New York, NY, USA, 2007. ACM.

[5] J. M. Gómez Hidalgo, G. C. Bringas, E. P. Sánz, and F. C. García. Content based sms spam filtering. In *DocEng '06*, pages 107–114, New York, NY, USA, 2006. ACM.

[6] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Mach. Learn.*, 45(2):171–186, 2001.

[7] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades, and I. Vlahavas. An adaptive personalized news dissemination system. *J. Intell. Inf. Syst.*, 32(2):191–212, 2009.

[8] I. Katakis, G. Tsoumakas, and I. Vlahavas. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowl. Inf. Syst.*, 22(3):371–391, 2010.

[9] E. Loper and S. Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching NLP and Computational Linguistics.*, 2002.

[10] C. D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1 edition, 1999.

[11] K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification. In *IJCAI-99*, pages 61–67, 1999.

[12] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *ICWSM 2010*, 2010.

[13] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW '08*, pages 91–100, New York, NY, USA, 2008. ACM.

[14] S. Ramamurthy and R. Bhatnagar. Tracking recurrent concept drift in streaming data using ensemble classifiers. In *ICMLA '07: Proceedings of the Sixth ICML*, pages 404–409, Washington, DC, USA, 2007. IEEE Computer Society.

[15] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[16] J. Sankaranarayanan, B. E. Teitler, M. D. Lieberman, H. Samet, and J. Sperling. Twitterstand: News in tweets.

[17] S. Siersdorfer, S. Chelaru, W. Nejdl, and J. San Pedro. How useful are your comments?: analyzing and predicting youtube comments and comment ratings. In *WWW '10*, pages 891–900, New York, NY, USA, 2010. ACM.

[18] D.-N. Sohn, J.-T. Lee, and H.-C. Rim. The contribution of stylistic information to content-based mobile spam filtering. In *ACL-IJCNLP '09*, pages 321–324, Morristown, NJ, USA, 2009. ACL.

[19] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *SIGIR '10*, pages 841–842, New York, NY, USA, 2010. ACM.

[20] S. Wu, J. M. Hofman, D. J. Watts, and W. A. Mason. Who says what to whom on twitter. *WWW 2011*, 2011.

[21] S. R. Yerva, Z. Mikls, and K. Aberer. It was easy, when apples and blackberries were only fruits. 2010.