LiveTweet: Microblog Retrieval Based on Interestingness and an Adaptation of the Vector Space Model

Arifah Che Alhadi, Thomas Gottron, Jérôme Kunegis, and Nasir Naveed

Institute for Web Science and Technologies
University of Koblenz-Landau
{alhadi, gottron, kunegis, naveed}@uni-koblenz.de

Abstract This paper presents the Institute for Web Science and Technology's contribution to the TREC2011 Microblog Track. The goal of the Microblog Track is to address the user's information need in which a user wishes to see not only the most recent but also the most interesting and relevant information to a query in Twitter. In this paper we present the LiveTweet system, submitted by the Institute for Web Science and Technologies (WeST) from the University of Koblenz-Landau. The system addresses two issues of microblog media: sparsity and its effect on document length normalization, as well as the problem of assessing content quality. We provide the following approaches to overcome these issues: ignoring length normalization and using *interestingness* as a static quality measure to find the most recent and interesting tweets related to a given query topic. The results in similar settings have shown that deliberately ignoring length normalization yields better retrieval results in general and that interestingness improves retrieval for underspecified queries.

1 Introduction

In microblog environments, documents are very short. In the largest microblogging service Twitter[1] the length of messages is restricted to 140 characters. While the brevity of messages has been cited as a major reason for Twitter's success, it poses at the same time problems for text retrieval. In this contribution we look into two particularities of information retrieval on microblogs: sparsity and document quality. Sparsity is inherent to microblog documents, as it reflects the technical constraints on message length. The quality of a document with respect to its ability to satisfy an information need originates from the different purposes and environments in which microblog messages are generated [3].

In our previous work on microblog retrieval [11] we motivated from a theoretical and analytical point of view that document length normalization introduces an unmotivated bias towards short documents. We further introduced interestingness as a static quality measure for microblog messages and showed that both

[1] http://twitter.com/

approaches improve retrieval performance in the sense of providing more relevant and generally interesting messages in the search results.

2 Related Work

Twitter is on the focus of a large number of research papers. Thus, in this section we concentrate on work covering the design or adaption of retrieval models as well as static quality metrics, such as influence, interestingness or user status.

The current form of Twitter's own search function does perform optimal, when it comes to rank interesting tweets at the top of the result list. As a remedy, Massoudi et al. [6] presented an approach incorporating query expansion and quality indicators (e.g. emoticons, post length, shouting, capitalization, hyperlinks, reposts, followers, and recency) into a retrieval model for searching microblog posts for a given topic of interest.

Some researchers [9,5,8] mentioned that, in addition to content features, the social status strongly correlates with the likelihood of a tweet to be retweeted and, thus, to have a wider reach. Nagmoti et al. [9] considered social network properties of the authors (e.g. the number of followers and followees or the number of posted tweets) to rank microblogs posts. They also used the tweet length and the presence of a URL as a quality measure of interestingness and informativeness of the information shared with other.

Hong et al. [5] measure the popularity of messages based on the number of retweet and use machine learning technique to predict how often new messages will be retweeted. The content of messages, temporal information, metadata of messages and users, and the user's social graph were used as features in predicting whether messages will get retweeted. Kwak et al. [8] used three different measures to identify influential users on Twitter. They ranked the users by the number of followers, PageRank and number of retweets. As a result, they found that the ranking of the users based on the number of followers and PageRank are very similar, while rankings based on the number of retweeted messages is different.

Cha et al. [2] also used three different measure; number of followers, number of mentions and number of retweets. They disagree that social network features such as a large numbers of followers is correlated with the likelihood of a user's messages to get retweeted. Hence, the social status is not sufficient as a static quality measure to indicate authors that will provide interesting information to their followers.

These current studies have shown that both the context and content of a tweet correlate with its retweetability. Within the limitations implied by the constraints of the TREC Microblog corpus and the task setup, our work covers both aspects. We consider the context in the form of identifying the author and use low-level and high-level content-features to introduce the probability of tweet to be retweeted as a static quality measure.

3 Retrieval Model for Microblogs

Our LiveTweet system is based on the classical vector space model, which we adapt in two ways to cope with the particularities of Microblog documents. First, we address the issue of sparsity and its effect on length normalization. Then, we address content quality in the sense of what causes a tweet to be of interest to a wider audience.

3.1 Length Normalization

We mentioned that microblog messages contain few terms in general, and very rarely contain a term more than once. Term frequencies are typically used as a parameter in the estimation of term importance within a given document. In short texts however, this essential feature does not discriminate much between different documents, as it is nearly a binary value. In fact, a brief analysis across several Twitter datasets shows that about 85% of all tweets contain any term at most once. A quite subtle impact of this sparsity lies in the length normalization typically used in classical retrieval settings.

Modern retrieval systems, in particular those based on the vector space model, use length normalization when retrieving documents from standard text corpora. Length normalization is introduced in the retrieval process to overcome undue benefit gained by longer documents that are explained by the *verbosity hypothesis* and the *scope hypothesis*.

Our analysis in [11] showed that neither the verbosity nor the scope hypothesis hold true for Twitter. We found that there is at best a very weak correlation between document length and redundancy which negates the verbosity hypothesis. At the same time the scope hypothesis is also negated by the fact that a single tweet typically addresses only one or at most two topics.

As neither the verbosity nor the scope hypothesis seem to apply to Twitter, we infer that length normalization for Twitter messages is not necessary. On the contrary, it is even counterproductive, as it introduces an unjustified bias favoring short messages over long ones. Accordingly, we do not use length normalization in the LiveTweet system.

3.2 Interestingness as a Static Quality Measure

Content quality is an attribute of a document which is independent of the query terms. In a retrieval setting and in particular in the context of Twitter, *interestingness* is one essential notion of content quality. We consider a tweet to be interesting – and therefore of good quality – if it is retweeted. However, whether a particular tweet is retweeted depends heavily on context, such as the user's position in the social graph or the time of day the tweet is posted. Generally, a tweet of a user with few or only passive followers is less likely to be retweeted. Similarly, tweets posted in the night tend to get retweeted less. Despite this, neither of these contextual pieces of data has any influence on the content or the quality of a tweet, and will thus introduce a contextual bias into any quality

measure. Therefore, we ignore the this context information in our analysis of tweets.

We follow our approach from [10] to determine interestingness via the probability for a tweet to be retweeted. There, only features based on the tweets themselves were considered, as described in the following.

Features

Exclamation and question marks. We use the presence of exclamation marks "!" and question marks "?" at the end of tweets as two binary features. Exclamation marks are used to mark strong emotional statements in personal communication or to mark interjections and exclamations in general text. Question marks indicate questions in all types, and are by their nature intended to elicit responses. While there might be alternative uses of both symbols, we considered this feature as it can indicate interesting tweets, which are therefore likely to be retweeted.

URLs, usernames and hashtags. Without further differentiation we consider the presence of particular items typical for tweets. These are the presence of a URL, the mention of a username or a hashtag. On Twitter, usernames and hashtags can be identified by their specific syntax using the pattern `@username` and `#hashtag`. We use the string `http`: to identify URLs. This gives us three binary features. Usernames are used in Twitter to refer to other users directly, either for addressing a user of for talking about him. Hashtags, or simply tags, are used to mark specific topics. They can be either inline in the messages or appended after the message itself. URLs are universally used to indicate the location of the full text being talked about.

Terms. The most obvious content feature in text is the contained terms. We extract terms and normalize them using case folding and the Porter stemmer [13]. Then, for each message M we compute the odds of it being a retweet. Assuming independence between the occurrences of terms and employing Bayes' theorem the odds value can be brought into a form that is easier to handle:

$$O(\text{retweet} \mid M) = \frac{P(\text{retweet} \mid M)}{P(\text{non-retweet} \mid M)} = O(\text{retweet}) \cdot \prod_{t \in M} \frac{P(t \mid \text{retweet})}{P(t \mid \text{non-retweet})}$$

where $O(\text{retweet})$ are the a priori odds of a retweet, and the product ranges over the ratios of the probabilities of each contained term to occur in a retweeted or a non-retweeted message. To estimate these probabilities we use maximum likelihood estimation and Laplacian smoothing to handle unseen terms.

Positive and negative terms. We look for positive and negative words from the short predefined list given in Table 1. Terms expressing positive and negative feelings have previously been found to influence social interaction in Twitter [12], and we conjecture them to also play a role in making a tweet interesting or uninteresting.

Emoticons. Emoticons are short character sequences representing emotions. We parse the tweets to find positive emoticons such as the smiley `:-)` and negative emoticons such as `:-(`, giving two binary features. Table 1 gives the complete list.

Table 1. Terms and emoticons expressing positive and negative emotions in Twitter messages.

	Positive	Negative
Terms	great like excellent rock on	bad sad fail eww
Emoticons	`:-) :) ;-)`	`:-( :(`

Sentiments. Many tweets are personal and express sentiments. To detect the sentiments expressed in a tweet, we follow previous Twitter research and select a simple dictionary-based approach [7]. We use the Affective Norms of English Words (ANEW) dictionary [1], which gives for 1,030 English words numerical values that capture valence (pleasure vs. displeasure), arousal (excitement vs. calmness) and dominance (weakness vs. strength).

User. We deviate from the feature set used in our previous work [10] on this one point: We additionally consider the usernames of authors to detect the most retweeted users. This means we looked at the authors of tweets on one hand and the usernames following the `RT` in retweets.

Learning Interestingness Based on the features introduced in the previous section we train an incremental Naive Bayes model to obtain for an individual tweet the probability of retweet. In line with our findings in [10], we interpret this probability as the quality of a microblog message. If the probability of retweet is high, the message is seen as interesting for a wider audience and, therefore, of better quality in a general retrieval scenario.

The model is incremental with respect to the temporal order of the tweets in the dataset. This means that for a tweet at time t_i, we use the tweets up to time t_{i-1} to train our Naive Bayes classifier. We then apply this classifier to determine the likelihood of the tweet at time t_i to be retweeted and assign this value as a static quality measure to the tweet. Then we include this tweet's features and the information whether it actually is a retweet into the classifiers knowledge base to update the prediction model for the next upcoming tweet at time t_{i+1}.

4 LiveTweet: System Setup

Given the limitation of the task to English tweets, we first use a language detection module to filter out all non-English tweets. The module is implemented using a dedicated language detection mechanism optimized for short texts [4]. We

manually create a gold standard for English and non-English tweets on a small subset of 1,000 tweets from the given TREC corpus. After removing URLs, usernames and hashtags as well as reducing excessive repetitions of single characters (e.g., mapping *cooooooool* to *cool*), we obtain a suitable accuracy of 96.9% at separating English from non-English tweets.

After filtering out the non-English tweets we compute the interestingness value of a tweet as defined in Section 3. Technically, our incremental Naive Bayes system assumes the presence and absence of features as results of a Bernoulli experiment with different a posteriori probabilities given we are observing an interesting (i.e., retweeted) or an uninteresting (i.e., not retweeted) tweet. As incorporating sentiment detection requires external knowledge in the form of a dictionary annotated with sentiments, we operated the system once without sentiment features (run `WESTfilter`) and once with sentiment features (run `WESTfilter`). In rest of this paper we only discuss the main `WESTfilter` run, as adding sentiment features did not show any significant improvements for most queries.

The actual LiveTweet retrieval system is based on a standard Lucene[2] system using the vector space retrieval model. We adapted the term weighting scheme of Lucene to neglect document length. The tweets are stored in the LiveTweet index along with the interestingness values that has been computed before.

In order to incorporate interestingness as static quality measure at retrieval time, we investigated two approaches: one based on filtering out non-interesting tweets, while maintaining a given ranking and one in which we additionally reranked the entries in a given result set according to their interestingness. For the purpose of filtering tweets of low interest we look at the relevant entries using a classical vector space model without length normalization. In this result set we look at the distribution of the interestingness values and identify a turning point in this distribution. We observed a general tendency of interestingness to decline fast after the most interesting tweet. Then, interestingness seems first to level out before starting again to drop more and more drastically. This turning point between the slowing and increasing decline in interestingness serves as a dynamic cutoff point (threshold t) in our system. The remaining tweets are ranked according to their interestingness value.

Summarizing our approaches, we submitted the LiveTweet system in four different settings which are as follows:

WESTfilter: retrieving and ranking tweets by our modified VSM and then filtering out tweets having an interestingness less than the threshold t.

WESTfilext: retrieving and ranking tweets by our modified VSM and then filtering out tweets having interestingness less than the threshhold t, but incorporated the sentiment of a tweet for computing its interestingness value.

WESTrelint: retrieval by the modified VSM, filtering out tweets having an interestingness less than the threshold t and finally re-ranking the tweets by their interestingness score.

[2] `http://lucene.apache.org/`

WESTrlext: retrieval by the modified VSM, filtering out tweets having an interestingness less than the threshold t and finally re-ranking the tweets by their interestingness score. Again, here we incorporated the sentiment of a tweet for computing its interestingness value.

5 Results

The official metric used by TREC for evaluating the effectiveness of systems in the retrieval scenario was P@30 in a tweet-ordered ranking. However, participating groups were encouraged to analyze their systems using other measures as well. In particular, TREC provided four possible scenarios for evaluation:

allrel The official evaluation scenario corresponds to a filtering task on a stream of incoming messages. Thus, the ranking of messages is provided by the time at which the tweets in the result set were produced. New tweets are ranked higher, older tweets are ranked lower. The actual challenge for the retrieval system is to filter out all irrelevant tweets from the incoming stream.

highrel For a subset of the topics, the relevance judgments distinguished between relevant and highly relevant tweets. While otherwise equivalent to **allrel** the **highrel** evaluation scenario considered only the highly relevant tweets as actually relevant.

by-score Different from the two previous scenarios, here the task is evaluated as a classical retrieval scenario. This means, that for each topic, the system can actually provide a ranking of the relevant tweets. As in classic TREC evaluation for such a setting, the ranking is imposed by the ordering the documents according to the relevance score provided by the system. This scenario corresponds more to the setting LiveTweet has been designed for. As relevance score we used the interestingness score for the tweets.

by-rank Additionally the TREC Mircoblog guidelines allowed to provide a ranking which diverged from the actual order imposed by the score. We used this freedom to use the ranking of a VSM for the tweets combined with a filter retaining only highly interesting messages. This means, the ranking is imposed by a classical retrieval model, but some tweets were discarded from the result set.

We used MAP, nDCG, P@5, P@10, P@20, P@30, R-Prec and bpref for allrel, highrel, by-score and by-rank to compute the performance of all four variants of the LiveTweet system. As stated above, for some topics the relevance of tweets was judged on a graded scale. This graded relevance judgments distinguish between the allrel and highrel evaluation scenario and we also used it to compute nDCG.

Figure 1 shows the performance of LiveTweet for different measures using the allrel evaluation scenario. In allrel, we do not see significant difference in the performance as the runs based on filtering and re-ranking provide the same resultset and the ranking is implied by the timestamps of the tweets. So, in allrel it is only of interest to compare between using or not using external knowledge.

While there is a small decline in the performance when introducing external knowledge, it is not of statistical significance.

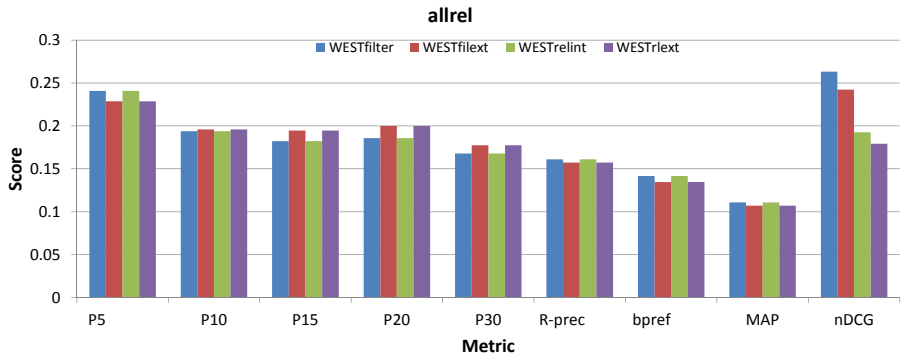

Figure 1. Mean average precision (MAP) of LiveTweet under allrel evaluation.

Figure 2 shows the performance of LiveTweet for different measures using by-rank evaluation scenario, which corresponds more to a retrieval scenario. We see again that the performance of the runs using or not using external knowledge does not differ significantly from each other across all the evaluation measures. Thus, the runs that are actually of interest for comparison are WESTfilter and WESTrelint. The best performance is achieved by the WESTfilter across all measures. Here, the observed improvements in performance are statistically significant.

Table 2 summarizes the results and provides information about significance of the improvements for allrel and by-rank between WESTfilter−WESTfilext and WESTfilter−WESTrelint. From the results we conclude that interestingness is more suitable to be incorporated as a filter function; re-ranking the results according to interestingness demonstrated a poorer performance.

Figure 3 finally shows the performance of LiveTweet variants over individual query topics under the by-rank evaluation scenario. Looking at individual topics gives additional insights, when considering the length of the actual query.

It has been observed in [14] that Web queries have an average length of 3.08 words, while on microblogs such as Twitter the average query has only 1.64 words. In Table 3 we provide an overview of the frequency distribution of the query topics in the TREC Microblog track with respect to the length of the query measured in words. This distribution is slightly in favor of longer queries which are more representative for general Web search but seem to be less typical for Twitter search.

Figure 4 and Figure 5 show the MAP performance of LiveTweet System with respect to number of terms in a query using by-rank and allrel evaluation scenarios respectively. In all of the four variants of the system we see a negative

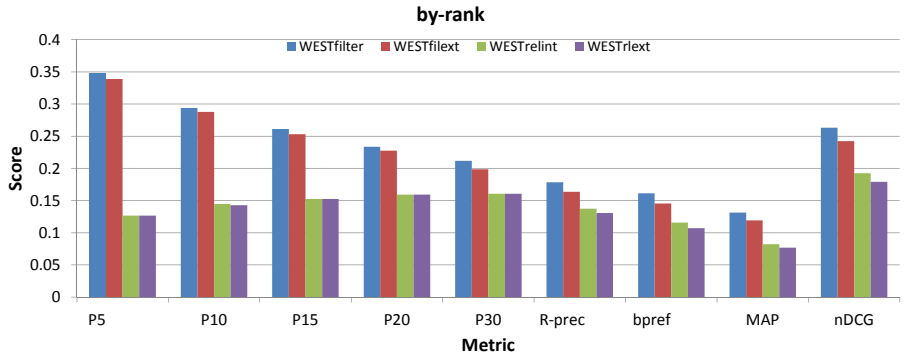

Figure 2. Mean average precision (MAP) of LiveTweet for by-rank.

Table 2. Significance test

	allrel			by-rank		
	WESTfilter	WESTfilext	Significance[a]	WESTfilter	WESTrelint	Significance[a]
P5	0.2408	0.2285	–	0.3469	0.1265	***
P10	0.1939	0.1959	–	0.2939	0.1449	***
P15	0.1823	0.1946	–	0.2612	0.1524	***
P20	0.1857	0.2	–	0.2337	0.1591	**
P30	0.168	0.1775	–	0.2116	0.1605	**
MAP	0.1109	0.1071	–	0.1312	0.0822	***
bpref	0.1416	0.1347	–	0.1612	0.1159	***

[a] – not significant, * significant at 5%, ** significant at 1%, *** significant at 0.1%

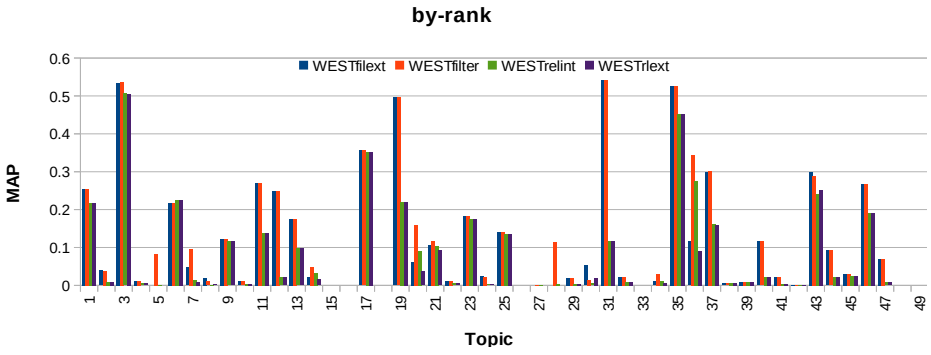

Figure 3. Mean average precision (MAP) of LiveTweet over individual topics for by-rank.

Table 3. Lengthwise distribution of query topics against their frequency

Query Length	1	2	3	4	5	6	7
Frequency	1	7	20	14	6	0	1

correlation between the query length and MAP performance of the system. As indicated in [11], using interestingness is particularly useful for short queries, as they are typical for Twitter [14]. We checked the correlation between the mean average precision and the length of the queries measured by the number of terms. We observe a strong negative correlation of -0.967 which hints in the direction that, as observed in previous work, the model used in LiveTweet actually performs better on short queries.

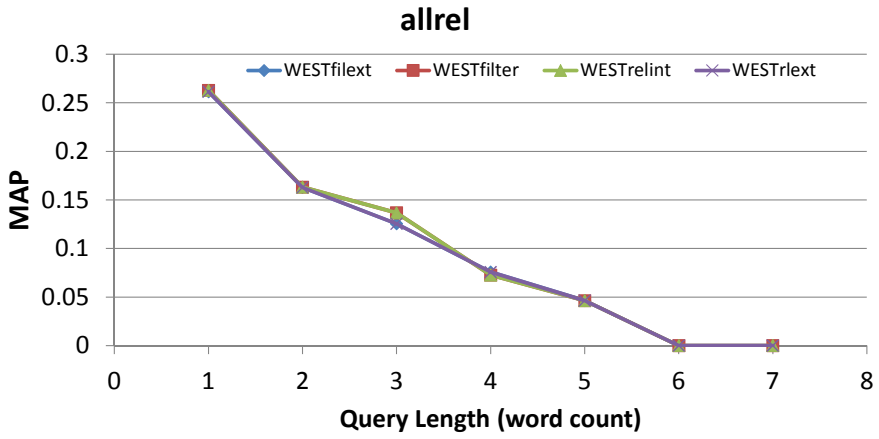

Figure 4. Mean average precision (MAP) of LiveTweet against query length for allrel.

6 Conclusion and Summary

In our TREC Microblog contribution, we focused on the sparse nature of microblogs and content quality. The system presented in this paper is heavily based on previous work and is implemented on top of a standard retrieval engine. We used interestingness as static quality measure and used the interestingness score to filter and re-rank tweets retrieved by modified vector space model.

As a next step, we are currently incorporating further static quality measures to the process. These include the social context of a user, the global social network structure and the freshness of results. Also, we plan to apply learning-

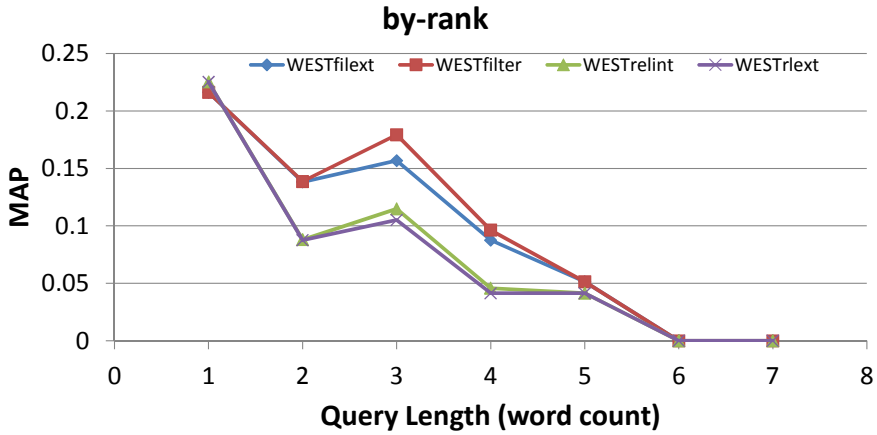

Figure 5. Mean average precision (MAP) of LiveTweet against query length for by-rank.

to-rank methods to identify the appropriate weighting of different features for computing a combined retrieval value.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST and grant agreement no. 248512, WeGov. This work was supported in part by HEC, Govt. of Pakistan in collaboration with DAAD, Germany. This work was also partly funded by the German Research Foundation (DFG) under the Multipla project (grant 38457858).

References

1. M. M. Bradley and P. J. Lang. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical report, The Center for Research in Psychophysiology, University of Florida, 1999.
2. M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in Twitter: the million follower fallacy. In *Proc. Int. Conf. on Weblogs and Social Media*, pages 10–17, 2010.
3. A. Che Alhadi, S. Staab, and T. Gottron. Exploring user purpose writing single tweets. In *Proc. Web Science Conf.*, 2011.
4. T. Gottron and N. Lipka. A comparison of language identification approaches on short, query-style texts. In *ECIR '10: Proceedings of the 32nd European Conference on Information Retrieval*, pages 611–614, Mar. 2010.
5. L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in Twitter. In *Proc. Int. World Wide Web Conf.*, pages 57–58, 2011.

6. M. Karam, T. Manos, d. R. Marten, and W. Wouter. Incorporating query expansion and quality indicators in searching microblog posts. In *Proc. European Conf. on Information Retrieval*, 2011.
7. E. Kim, S. Gilbert, M. J. Edwards, and E. Graeff. Detecting sadness in 140 characters: Sentiment analysis and mourning Michael Jackson on Twitter. Technical report, Web Ecology Project, Aug 2009.
8. H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proc. Int. World Wide Web Conf.*, pages 591–600, 2010.
9. R. Nagmoti, A. Teredesai, and M. De Cock. Ranking approaches for microblog search. In *Proc. Int. Conf. on Web Intelligence and Intelligent Agent Technology*, pages 153–157, 2010.
10. N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proc. Web Science Conf.*, 2011.
11. N. Naveed, T. Gottron, J. Kunegis, and A. Che Alhadi. Searching microblogs: Coping with sparsity and document quality. In *Proc. 20th ACM Conf. on Information and Knowledge Management*, 2011.
12. A. Pepe, H. Mao, and J. Bollen. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *CoRR*, abs/0911.1583, 2009.
13. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
14. J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and Web search. In *Proc. Int. Conf. on Web Search and Data Mining*, pages 35–44, 2011.