Sentiment-Focused Web Crawling

A. Gural Vural
Middle East Technical University
Ankara, Turkey
gural@ceng.metu.edu.tr

B. Barla Cambazoglu
Yahoo! Research
Barcelona, Spain
barla@yahoo-inc.com

Pinar Senkul
Middle East Technical University
Ankara, Turkey
senkul@ceng.metu.edu.tr

ABSTRACT

The sentiments and opinions that are expressed in web pages towards objects, entities, and products constitute an important portion of the textual content available in the Web. Despite the vast interest in sentiment analysis and opinion mining, somewhat surprisingly, the discovery of the sentimental or opinionated web content is mostly ignored. This work aims to fill this gap and address the problem of quickly discovering and fetching the sentimental content present in the Web. To this end, we design a sentiment-focused web crawling framework for faster discovery and retrieval of such content. In particular, we propose different sentiment-focused web crawling strategies that prioritize discovered URLs based on their predicted sentiment scores. Through simulations, these strategies are shown to achieve considerable performance improvement over general-purpose web crawling strategies in discovering sentimental content.

Categories and Subject Descriptors

H.3.3 [**Information Storage Systems**]: Information Retrieval Systems

General Terms

Design, Experimentation, Performance

Keywords

Sentiment analysis, focused web crawling

1. INTRODUCTION

The advent of Web 2.0 has led to an increase in the amount of sentimental content available in the Web, specifically the textual content that involves sentiments and opinions [9, 10]. In the last decade, a large body of research investigated the extraction, classification, retrieval, summarization, and presentation of the sentimental content obtained from the Web. Interestingly, however, the discovery of such content has not received much attention from the research community, despite the fact that the discovery is the first step that enables any other type of processing. It is evident that a fairly large amount of sentimental content is easily accessible through certain web APIs and RSS feeds. Nevertheless, a much larger amount remains inaccessible through this kind of passive content discovery techniques since the sentimental content is spread across millions of web sites. Therefore, conventional active content discovery techniques need to be adopted, i.e., the content has to be crawled from the Web.

In this work, our focus is on the quick discovery of the sentimental content available in the Web. In this context, the timely discovery is important as most sentiments or opinions quickly lose their value if they are not immediately discovered. To facilitate the quick discovery of sentimental content, we develop a sentiment-focused web crawling framework, where the goal is to prioritize the crawling of sentimental content with respect to non-sentimental content. Within our framework, we propose different techniques to estimate the sentimentality of an “unseen” web page and guide the crawling process through these estimates.

The following are the contributions of the paper.

- To best of our knowledge, we make the first sentiment-focused web crawler design, using the state-of-the-art page processing and sentiment analysis tools.
- We propose alternative techniques for predicting the sentimentality of web pages and their prioritization by the crawler.
- We evaluate our techniques through simulations that are conducted over a subset of the publicly available ClueWeb09-B web page collection,[1] allowing the reproducibility of our findings. The experimental results indicate considerable improvement in early discovery of sentimental content with respect to the baseline general-purpose crawlers.
- We conduct a user study to select a set of parameters that enable us to create a ground-truth for the sentimentality of the web pages in the ClueWeb09 data.

The rest of the paper is organized as follows. We present the problem and the proposed framework in Section 2. Section 3 includes the results of the user study that is conducted to create a ground-truth for our web page sample. Section 4 provides the details of our experimental setup. The experimental results are presented in Section 5. Section 6 contains a brief survey of the related work. Finally, the paper is concluded in Section 7.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

[1] ClueWeb09 – TREC 2009 “Category B” dataset, `http://lemurproject.org/clueweb09.php`.

2. SENTIMENT-FOCUSED CRAWLING

In the sentiment-focused web crawling problem, the goal is to maximize the total sentimentality of the crawled pages at the early stages of the crawling. In this work, we adopt a simple greedy approach where pages are crawled in non-increasing order of their estimated sentimentality values. In this approach, at each iteration, the crawler downloads the page that has the highest sentimentality value among the pages in its download queue. The approach is based on the assumptions that i) the sentiment of a page can be accurately estimated to a certain degree without having its content, only by using some auxiliary information, and ii) the pages that are linked by sentimental pages are also likely to be sentimental. The findings of our work indicate that these two assumptions are reasonable (see Section 5).

We design a sentiment-focused web crawling framework that involves four main components: page retrieval, storage, text processing, and sentiment-based URL prioritization components. These components are described below.

Page retrieval component. This component, which forms the heart of our framework, is responsible for fetching the web pages from the Web and storing them in a database in the form of HTML files. Many implementation issues (e.g., handling of duplicates, DNS caching, multi-threading, politeness policies) are omitted here for the brevity of the presentation. Most of these issues can be handled as in a general-purpose crawling framework.

Storage component. The storage component involves different databases, mainly for storing the web pages, extracted links, and various features obtained from the pages.

Text processing component. Once a page is fetched from the Web, it is passed through the text processors in this component. First, the tags in the page are removed to obtain the textual content of the page. The parser used in this step may affect the obtained content. In our work, we consider two different parser alternatives: Html Parser[2] and BoilerPipe.[3] Among the available extraction options of BoilerPipe, we use the "CanolaExtractor" since it obtained the best extraction performance over a large number of pages. The obtained content is then split into sentences and words using the Stanford NLP library.[4] In the mean time, the URLs that are linked by the page are extracted. The data generated by the above-mentioned text processors are used to generate some features associated with the textual content of the page as well as the URLs and the anchor text extracted from the page. An important feature is the sentiment score of the page. To compute this score, we rely on the SentiStrength software [14],[5] which is a lexicon-based sentiment analysis tool. This tool generates a negative and a positive sentiment score for a given piece of text. We compute the sentiment score of a sentence in the page by adding the absolute values of the negative and positive scores associated with the sentence [10]. The sentimentality of a page is simply computed as the average of the sentimentality scores of all sentences in the page. As an alternative to this approach, we also compute the sentiment score of a page by taking an average over the absolute sentiment scores of the individual words in the page. As the lexicon of sentimental words, we use the default word list provided by the SentiStrength tool. As an alternative, we also consider a smaller lexicon that contains only the emotional adjectives.[6] All of these alternatives are evaluated in a user study (see Section 3) to identify the best design choices for our framework.

Table 1: The features used by the learning model

Type	Feature description
Referring page	Average page size (w/ HTML tags) Average page size (w/o HTML tags) Number of DOM objects Number of outgoing links Number of pictures Number of self links Number of sentences Number of words Number of unique words Ratio of links to page size Ratio of number of links to page size Average sentence length Average sentiment score of content Maximum sentiment score of sentences σ for sentiment score of content Average sentiment score of keywords Average sentiment score of titles
Anchor text	Term count Average sentiment score
Page URL	Sentiment score

Sentiment-based URL prioritization component. This component forms the brain of our sentiment-focused web crawler. For every discovered URL whose content is not yet downloaded, a sentiment score is estimated. The URLs are maintained in a download queue in non-increasing order of these estimates. At each iteration, the URL with the largest estimated sentiment score, i.e., the one with the highest likelihood of being sentimental, is passed to the page retrieval component as the next URL to be downloaded.

We evaluate three alternative techniques for predicting the sentiment score of a web page whose URL is discovered but the textual content is not yet downloaded.

- **Based on referring anchor text.** A sentiment score is computed for every anchor text extracted from the downloaded pages. The sentiment score of a page is estimated by the average of the sentiment scores of the anchor text on the links referring to the page.
- **Based on referring page content.** The sentiment score of a page is estimated by the average of the sentiment scores of the pages that refer to the page.
- **Based on machine learning.** A machine learning model is built using some features extracted from the previously downloaded pages. An instance in the model corresponds to a web page. The prediction target is the actual sentiment score of the page. The complete list of the features used by the model is given in Table 1. The model is periodically rebuilt using all of the pages downloaded so far and is used to predict the sentiment scores of unseen pages.

[2] HTML Parser (version 2.0), http://htmlparser.sourceforge.net.

[3] BoilerPipe (version 1.2.0), http://code.google.com/p/boilerpipe.

[4] Stanford Core NLP library (version 1.2.0), http://nlp.stanford.edu/software/corenlp.shtml.

[5] SentiStrength, http://sentistrength.wlv.ac.uk.

[6] The adjective list is compiled by the authors by filtering non-adjective terms from the original SentiStrength lexicon.

Table 2: The degree of agreement among the judges

		Overlap (O)				Kappa (κ)			
Judge	S	J1	J2	J3	GT	J1	J2	J3	GT
J1	0.23	1.00	0.87	0.87	0.89	1.00	0.64	0.59	0.63
J2	0.24	0.87	1.00	0.85	0.88	0.64	1.00	0.55	0.61
J3	0.18	0.87	0.85	1.00	0.94	0.59	0.55	1.00	0.78
Avg.	0.22	0.91	0.91	0.91	0.91	0.74	0.73	0.71	0.67

Table 3: The ranking quality achieved by different parameter combinations over 500 randomly sampled pages

Metric	Rand	HP-SS-All	HP-WS-All	BP-SS-All	BP-WS-All	HP-SS-Adj	HP-WS-Adj	BP-SS-Adj	BP-WS-Adj
P@10	0.12	0.40	0.30	0.40	0.40	0.50	**0.60**	0.30	0.40
P@50	0.12	0.30	0.40	0.36	0.36	0.26	0.40	0.32	**0.44**
P@100	0.12	0.29	0.30	0.32	**0.35**	0.29	0.31	0.33	**0.35**
AP	0.13	0.31	0.33	0.36	0.37	0.32	0.37	0.35	**0.41**
DCG	8.47	10.36	11.05	10.89	10.82	11.15	11.50	11.27	**11.76**

3. USER STUDY

Evaluating the performance of different focused crawling techniques in fetching sentimental content requires knowing the actual sentiment scores of downloaded pages. In our case, unfortunately, there are no ground-truth sentiment scores available for our web page collection. In order to create a ground-truth, we use the estimated sentiment scores as substitute for the actual sentiment scores.

To be able to generate the score estimates, we need to identify the best combination of parameters that yield the page sentiment scores as accurately as possible. In our implementation of the sentiment-focused crawling framework, we have three sets of parameters to select from (see Section 2). First, we need to decide whether the textual content of a page is extracted using Html Parser (HP) or BoilerPipe (BP). Second, the sentiment scores can be computed based on the sentence scores (SS) or word scores (WS). Third, as the lexicon for SentiStrength, we can use all sentimental words (All) or only the sentimental adjectives (Adj). Consequently, there are eight possible parameter combinations to be considered: HP-SS-All, HP-SS-Adj, HP-WS-All, HP-WS-Adj, BP-SS-All, BP-SS-Adj, BP-WS-All, and BP-WS-Adj.

To identify the best performing parameter combination, we conduct a small-scale user study. We randomly sample 500 pages from our collection (see Section 4 for the details of the page sample used in our work). Every web page is classified by three judges (J1, J2, and J3), each judge individually assigning the labels "sentimental" or "not sentimental" to a page. We also define a ground truth (GT) for the judged pages: a page is labeled as sentimental only if all three judges agree on the sentimentality of the page; otherwise, the page is labeled as being not sentimental.

To quantify the agreement between different judges, we use the overlap metric and Cohen's kappa. Table 2 displays the agreement between different pairs of judges as well as their agreement with the above-mentioned ground-truth. The fraction of sentimental pages is identified as 0.23, 0.24, and 0.18 by the three judges (the S column in the table), who labeled about 22% of the pages as sentimental, on average. According to the table, we observe high agreement between the judges, the overlap (O) in their decisions reaching above 85%. The kappa values (κ) also indicate substantial agreement although their interpretation is relatively difficult.

The high inter-judge agreement indicates that the labels obtained through the user study form a sufficiently reliable basis to evaluate the performance of the parameter combinations mentioned before. We next compute the sentiment scores for our sample pages using the eight possible parameter combinations, each yielding a ranking where the pages are sorted in non-increasing order of their estimated sentiment scores. As the evaluation metrics, we use the average precision (AP) and discounted cumulative gain (DCG) metrics, as well as the precision values obtained at ranks 10, 50, and 100. As the baseline, we assume a random ranking, where the pages are randomly sorted. In this case, the metrics reflect the average of one million trials, each started with a different random seed.

Table 3 provides the computed metrics for different rankings. According to the results, all of our rankings perform considerably better than the random ranking. Overall, the BP-WS-Adj combination is the best parameter combination that yields the most accurate ranking. In the rest of our work, we use the sentiment scores estimated according to the BP-WS-Adj combination as the ground-truth sentiment scores for our web page collection.

4. EXPERIMENTAL SETUP

Web page collection. As our web page collection, we use a random sample obtained from the publicly available ClueWeb09-B web page collection, which contains about 50 million English web pages. Our sample contains 1,185,385 web pages. About half of the pages in the sample (53.3%) fall into the spam category. About 15.8% do not link to any other page and about 26.1% do not receive a link from any other page. On average, a page contains 696.0 words (247.1 unique words) and 16.0 sentences. There are 63.1 outgoing and 14.9 incoming links per page, on average. The average page size is 30,556 and 3,228 bytes before and after parsing the HTML tags, respectively.

Each page in our sample is associated with three different scores: sentiment score, spam score, and PageRank. The sentiment score values are computed using the BP-WS-Adj parameter combination (see Section 3). The spam scores are computed by Waterloo Spam Rankings using a simple content-based classifier [7].[7] Among the four sets of spam

[7] Waterloo Spam Rankings, http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/.

scores, we use the "Fusion" subset, where the lower spam scores indicate a larger likelihood for the page content to involve spam. As the PageRank values, we use the values that are already available in the ClueWeb09-B collection.

Software and hardware. For performance evaluation, we simulate the framework described in Section 2. The simulator is implemented in Java. We store the web pages and extracted features in a MySQL database (version 5.1.61). We use a 16-core computer with 48GB of RAM.

Crawlers. We evaluate three different sentiment-focused web crawlers, each adopting one of the sentiment prediction strategies described in Section 2: based on anchor text (`S-AT`), page content (`S-PC`), or machine learning (`S-ML`). We compare these strategies against three different general-purpose crawling baselines: random (`B-RA`), indegree-based (`B-ID`), and breadth-first (`B-BF`). The random crawler arbitrarily picks a URL from the download queue at each iteration. The indegree-based crawler always visits first the pages having the largest indegree. The breadth-first crawler visits the pages in breadth-first search order. In addition to these baselines, we also evaluate three different oracle crawlers, each prioritizing the URLs according to a different metric: sentiment score (`O-SE`), spam score (`O-SP`), and PageRank (`O-PR`). The PageRank crawler always prefers visiting the URLs with the highest PageRank values while the spam crawler prefers those pages with the highest spam scores (i.e., less likely to be spam). We note that, although these oracles have perfect knowledge of the individual scores of pages, their choices are limited to the pages that are available in the download queue, i.e., they are not hypothetical oracles that can arbitrarily fetch any web page without following the link structure among the pages.

Machine learning model. For the sentiment-focused crawler that relies on machine learning, we rebuild the predictive model over all crawled pages at regular intervals (after crawling every 1,000 pages). As the learner, we use the LibSVM software [5] with the regression mode.[8]

5. EXPERIMENTAL RESULTS

In our experiments, we observe the increase in the total sentiment score of the downloaded pages as the crawling proceeds according to different crawling strategies. Figs. 1(a) and 1(b) show two different scenarios where spam filtering is not present or present, respectively. In case of spam filtering, we simply ignore the downloaded page if it is detected as spam, i.e., the links inside the page are not added to the download queue. As suggested by the creators of the spam scores, we assume that pages whose spam score is less than 70 are spam pages.

According to Fig. 1(a), as expected, the oracle crawler that prioritizes pages according to their actual sentiment scores (`O-SE`) achieves the best performance. This crawler can accumulate about two-third of the sentimentality available in the web page sample after crawling only less than half of the accessible pages. This finding justifies the second assumption we made in Section 2, i.e., the links in sentimental pages are likely to lead to other sentimental pages. The proposed sentiment-focused crawling techniques also perform quite well, `S-PC` and `S-ML` performing slightly better than `S-AT` but similar to each other. Even better, the performance gap between the proposed techniques and the oracle is not very large. This justifies the first assumption made in Section 2, i.e., the predicted sentiment scores are good substitutes for the actual sentiment scores. In general, the remaining strategies show relatively inferior performance.

As demonstrated in Fig. 1(b), the performance of different strategies is affected by spam filtering. As before, `O-SE` is the best performing strategy, which discovers a very large portion of the sentimental pages by crawling only half of the accessible pages. The `S-PC` strategy performs well at the early stages of the crawling while `S-AT` is better at the later stages. This means that a hybrid strategy between the two may be adopted. Indeed, the `S-ML` strategy, which uses the sentiment scores of both strategies as features, works like such a hybrid strategy. Interestingly, the oracle that prioritizes the pages by their PageRank values also performs well, especially at early stages. The `B-BF` and `B-ID` baselines as well as the `O-SP` oracle, which prioritizes the pages by their spam scores, perform worse than the random strategy `B-RA`, which demonstrates an almost linear, steady performance.

6. RELATED WORK

To the best of our knowledge, there is no prior work on sentiment-focused web crawling. Hence, herein, we provide a brief survey of the related work in the context of focused web crawling and sentiment analysis.

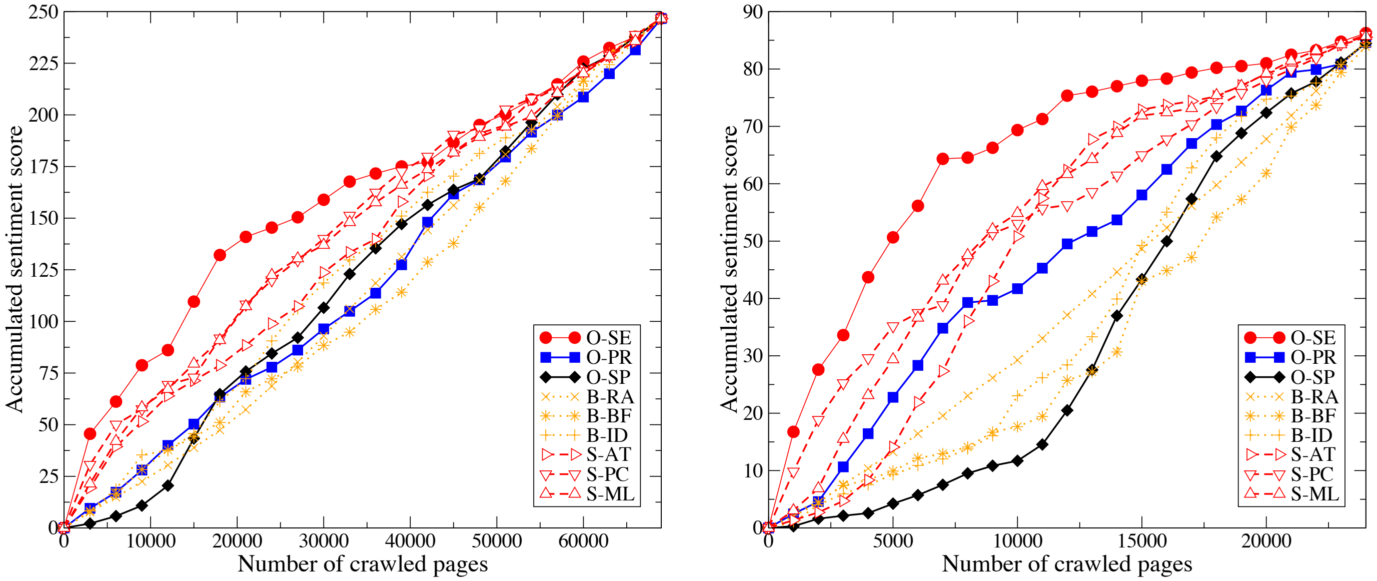
A focused web crawler aims to collect web pages that are related to a predefined set of topics. The topics are defined through a set of sample documents and the relevance of a discovered but not yet crawled page is estimated. Therefore, it is necessary to derive a powerful model for the topic out of the provided samples [3, 4]. To this end, earlier works employ different techniques such as Hidden Markov Model, Conditional Random Fields [11], and Support Vector Machines. To improve the learned model, ontology [8] and link semantics are also exploited. There are also works that concentrate on the context rather than focusing on a particular topic (e.g., discovering location-specific documents [2]).

With the increase in the amount of subjective content, such as blogs and reviews, the interest in discovering the sentimental value of web pages has also increased. Sentiment analysis basically aims to discover whether a given text is subjective or not [13]. In [12], the sentimentality and polarity are particularly sought for a given subject. The basic techniques employed are generally based on linguistic approaches [12] or machine learning techniques [15]. In the linguistic approaches, the similarity and dependency between the observed words and prior sentiment scores of a defined set of words are used. Some works concentrate on domain-specific sentiment analysis [6]. Although most of the research concentrate on analysis of English text, there are efforts on analysis of other languages [1] or multiple languages [15].

7. CONCLUSIONS

We presented a framework for sentiment-focused web crawling and proposed different strategies for predicting the amount of sentimentality present in a web page whose textual content is not available. The findings of our work are as follows i) sentiment-focused web crawling is feasible as the link structure of sentimental web pages allow quick discovery of new sentimental pages, ii) sentiment scores of web

[8] LibSVM (version 3.1.12), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.



(a) Without spam filtering.

(b) With spam filtering.

Figure 1: The amount of sentimentality accumulated while pages are crawled by different crawling techniques.

pages can be predicted to a certain degree without having the textual content of the page, and iii) sentiment-focused web crawling beats the traditional crawling alternatives in terms of the early discovery and retrieval of sentimental content from the Web.

A future research issue is the identification of new features that can further improve the sentiment prediction performance. Another possibility is to apply the proposed techniques to similar focused crawlers, e.g., early discovery of web content that involves positive polarity. Such content may be valuable for certain "niche" search engines, such as those targeting children.

8. ACKNOWLEDGMENTS

This work is supported by grant number TUBITAK-112E002, TUBITAK.

9. REFERENCES

[1] A. Abbasi, H. Chen, and A. Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):12:1–12:34, 2008.

[2] D. Ahlers and S. Boll. Adaptive geospatially focused crawling. In *Proc. 18th ACM Int'l Conf. Information and Knowledge Management*, pages 445–454, 2009.

[3] I. S. Altingovde and O. Ulusoy. Exploiting interclass rules for focused crawling. *IEEE Intelligent Systems*, 19(6):66–73, Nov. 2004.

[4] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11-16):1623–1640, 1999.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[6] Y. Choi, Y. Kim, and S.-H. Myaeng. Domain-specific sentiment analysis using contextual feature generation. In *Proc. 1st Int'l CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*, pages 37–44, 2009.

[7] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *CoRR*, abs/1004.5168, 2010.

[8] M. Ehrig and A. Maedche. Ontology-focused crawling of web documents. In *Proc. 2003 ACM Symp. Applied Computing*, pages 1174–1178, 2003.

[9] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-scale sentiment analysis for news and blogs. In *Proc. Int'l Conf. Weblogs and Social Media*, 2007.

[10] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu. A large-scale sentiment analysis for Yahoo! Answers. In *Proc. 5th ACM Int'l Conf. Web Search and Data Mining*, pages 633–642, 2012.

[11] H. Liu, E. Milios, and J. Janssen. Probabilistic models for focused web crawling. In *Proc. 6th ACM Int'l Workshop on Web Information and Data Management*, pages 16–22, 2004.

[12] T. Nasukawa and J. Yi. Sentiment analysis: capturing favorability using natural language processing. In *Proc. 2nd Int'l Conf. Knowledge Capture*, pages 70–77, 2003.

[13] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2:1–135, 2008.

[14] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, 2010.

[15] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proc. 20th ACM Int'l Conf. Information and Knowledge Management*, pages 1031–1040, 2011.