Suggesting Hashtags on Twitter

Allie Mazzia, James Juett
Computer Science & Engineering, University of Michigan

Abstract

As micro-blogging sites, like Twitter, continue to grow in popularity, we are presented with the problem of how to effectively categorize and search for posts. Looking specifically at Twitter, we see that users may categorize their posts using *hashtags*, and any word or phrase may be used as the category. Attempting to search for tweets about Facebook, a user would need to try many different hashtags, like *#Facebook, #FB, #Facebook.com,* or *#Zuckerberg*. To combat this, we propose, implement and evaluate a tool for suggesting relevant hashtags to a user, given a tweet. Initial analyses suggest our dataset is rich enough to extract informative distributions of words for many hashtags that will facilitate a naive Bayes model for hashtag recommendation given a query post.

1. Introduction

With 175 million registered users who post a collective 95 million times per day, Twitter has become one of the largest and most popular microblogging websites. To cope with the volume of information shared daily, Twitter has introduced *hashtags*, keywords prefaced with "#", to help users categorize and search for tweets.

Inspection of the public Twitter feed shows that not every tweet includes a hashtag. To help Twitter users more easily incorporate hashtags into their posts, we propose an automatic hashtag suggestion tool that, when given a tweet, will return a short list of relevant hashtags as suggestions.

2. Method

2.1 Data Collection

The first step in creating our tool is data collection. Twitter provides a convenient API[1] that allows us to easily monitor and download the Twitter feeds of a number of users. We have monitored the feeds of 5000 Twitter users over the course of almost two months, collecting their tweets and any hashtags used. Our data set contained over 1.3 million tweets and is discussed in more detail in section 4.

2.2 Pre-processing

A potential obstacle in our learning algorithm is that hashtags are generally noisy labels for classifying tweets into coherent categories. This arises from the fact Twitter users are not restricted in how they use hashtags. In addition to categorizing a Twitter post, hashtags may be used to offer more general commentary on a tweet. For example, users may include hashtags like "#lol" when sharing something funny or "#nowplaying" to share music they like. Another pattern of hashtag usage on Twitter surrounds the phenomenon of *micro-memes* as defined by Huang et al. In this case, a hashtag is used to participate in the micro-meme conversation rather than to label the content of a tweet.

These patterns and others, such as re-tweets, can significantly reduce the coherence of tweets for a particular hashtag, so it is crucial to identify whether a hashtag is likely to be associated with coherent tweets. To this end, we have developed a pre-processing schema that will eliminate both the above patterns and words that are likely to impede the detection of relevant hashtags.

Pre-processing begins with removing internet slang, stopwords, punctuation, links and re-tweets. To remove internet slang, we use a large slang-to-English dictionary[2]. Using this dictionary, we remove all slang and replace it with its English equivalent. We remove stopwords in a similar manner, using a list of 544 common words and removing each instance. To remove re-tweets and links, we use simple pattern-matching, since re-tweets and links follow common patterns (i.e. "RT @original_poster" or "http://www.umich.edu"). We then manually inspected our data set, removing all tweets that were in a foreign language, or did not include English words (some tweets were

[1] http://dev.twitter.com/doc

[2] http://www.noslang.com/dictionary

gibberish).

After we processed the data set into solely English tweets without punctuation, stopwords or links, we removed hashtags classified as "micro-memes." Micro-memes present some difficulty, since their format does not differ from that of other tweets. We base our algorithm for eliminating micro-memes on a property identified in Huang et al., that micro-memes are often tweets containing text about many unrelated topics, that all conform to the same "theme" of the micro-meme. Thus, if several tweets containing the same hashtag are dissimilar, we can infer that they are likely both micro-memes. To measure the content similarity of two tweets, we use the standard cosine similarity over the bag-of-words vector representation of the tweet content, as described in Lee et al. Using the similarity value, we can compute the average content similarity over all tweets containing the same hashtag, discarding those with low values. Currently, our tool precomputes the similarity values for all hashtags, and allows the user to specify the minimum similarity value that a hashtag must possess in order to be suggested.

Finally, we noticed that our data set contained a number of tweets from spam accounts that skewed our model towards words present in the spam tweets. An example of this is an account that published several hundred tweets with the hashtag "#jobs", all of which contained the word "webdesign". This will produce some irrelvant recommendations for regular users tweeting about web design but will also have the more subtle effect of overwhelming the association between other words like "apple" or "ipad" and the potentially relevant hashtag "#jobs". To combat this, we capped the number of effective tweets that a particular user could contribute to a hashtag's distribution in our model. If an account contributed more than 10 tweets to a particular hashtag, then each of its tweets was assigned a fractional weight so that their effective weight was no more than 10 tweets. Overall, this encourages our algorithm to learn from the past tweets of many users rather than one extremely prolific user.

2.3 Determining Relevance

We use a naive Bayes model to determine the relevance of hashtags to an individual tweet. Although the content of a tweet is arbitrary, we only consider a certain subset of possible words as a vocabulary. It would be possible to use a vocabulary based on standard word usage, but the prevalence of Internet-specific abbreviations and slang in tweets suggest a vocabulary based on the most common words in actual twitter posts will be more useful. From preliminary analysis of our data set, we have chosen the top 50,000 words used as our vocabulary. Consequentially, the features of a tweet can be represented as a vector x where $x_i = 0$ or 1 to indicate the presence or absence of the i_{th} dictionary word. Additionally, since users may preface *any* word with a hash symbol to create a hashtag, we only consider a subset of possible hashtags for recommendations. We discuss the selection of this subset below.

Using Bayes' rule, we can determine the posterior probability of Ci given the features of a tweet $x_i...x_n$:

$$p(C_i|x_1,...,x_n) = p(C_i)p(x_1|C_i)...p(x_n|C_i) / p(x_1...x_n)$$

In the simplest case, the prior probability $p(C_i)$ may be assumed the same for all C_i to give a maximum likelihood approach. The term $p(x_1...x_n)$ essentially serves as a normalizing constant and can be ignored when selecting the most probable hashtags because it does not depend on the hashtag C_i. Finally, using the naive Bayes assumption that the features x_i are conditionally independent given C_i, we can write the likelihood $p(x_1,...,x_n|C_i)$ as the product $p(x_1|C_i)...p(x_n|C_i)$. Each of these conditional probabilities is based on the frequency with which each word appears in tweets with hashtag C_i, given user type *u*.

2.4 Additional Methods Investigated

We also investigated extensions to the simplest naive Bayes model. Since many words in our dictionary never appear in a tweet for a particular hashtag, we could use Laplace smoothing to correct for these zero probabilities. However, simple Laplace smoothing causes an interesting problem because our data is fairly sparse for individual hashtags. If a hypothetical hashtag "#fastCar" occurs in only 5 tweets, none of which contain the word "cake", Laplace smoothing assigns cake a probability of ⅙ given the hashtag "#fastCar". This is clearly too high and will in fact be higher than for more appropriate hashtags such as "#yum" which might have many tweets but only contain "cake" in 10% of them. To avoid this problem, we smooth by using a word's frequency in the overall vocabulary from our dataset.

A common technique in information retrieval and text processing tasks is to use the term frequency-inverse document frequency weight (TF-IDF) to influence a word's importance. Because Twitter posts are extremely

short, a single word appearing multiple times is most likely anomalous, but the IDF weight can still offer valuable information about the specificity of a word. If a word in a tweet is very rare, it is likely that word has only been included because of its semantic importance rather than by mere coincidence. This reasoning led us to also try a variant of our simple naive Bayes model in which the likelihood term is given instead by:

$$p(x_1,...,x_n|C_i) = p(x_1|C_i)^{(1-t_1)}...p(x_n|C_i)^{(1-t_n)}, \text{ where } t_j \text{ is the IDF weight of word } x_j$$

In this model, the contribution of very common words will shrink to be negligible. In the experiments that follow, we used term frequency data[3] from the American National Corpus (Ide and Suderman, 2006).

We can also extend the simple naive Bayes model by taking advantage of the popularity of each hashtag. We can derive a prior probability $p(C_i)$ from the number of times a particular hashtag is used compare to the total number of tweets. In essence, this extension captures the notion that if we are to suggest a rare hashtag, we need more evidence that the tweet matches that hashtag in terms of content than if we were to suggest a commonly used hashtag.

We planned to compare our Naive Bayes recommendation algorithm with an alternative recommendation algorithm based on supervised latent Dirichlet analysis (sLDA). However, our data set contained over 1 million tweets, and we were unable to complete the analysis due to space and time restrictions on our machines. Analysis of a corpus (a test set) containing 3000 documents (tweets) from 30 classes (hashtags) using a C++ implementation of sLDA[4] took 10 hours to run on one author's laptop. We are currently investigating ways to reduce the computation time and space required, but as of this writing we have not yet had success. One advantage of the naive Bayes model is that the assumptions involved are strong enough to make inference easily feasible for such a massive data set.

3. Related Work

To our knowledge, there has been no published work on recommending hashtags to Twitter users on anywhere near the scale we propose. However, there has been extensive analysis of *social tagging systems* (Marlow et al. 2006) and categorization. In addition to work in the aforementioned areas, our proposed algorithm also draws on elements of text categorization and keyword extraction.

Hashtags as used on Twitter can be seen as a keyword or label for a post. A key difference, however, is that many keyword extraction systems only extract candidate keywords from those words or phrases already present in the text. We seek to *recommend* relevant hashtags not limited to words already present in a Twitter post.

There has been a significant amount of research on keyword extraction from documents, and specifically keyword extraction on the web. Yih et al. address the problem of finding advertising keywords on web pages. Their system incorporates traditional text mining features such as TF-IDF (Term Frequency-Inverse Document Frequency) and linguistic features, but also web-unique features like a document's URL, meta attributes, and even search query logs.

Li et al. discuss a method for extracting keywords from *social snippets*, which include micro-blogs like Twitter posts. They discuss a number of features and measure the relative importance of each feature in their keyword selection algorithm. Specifcally, they propose the term frequency component of TF-IDF may not be particularly useful because term repetition is uncommon Twitter posts. We build upon their approach by incorporating additional features and by working from a dictionary of possible hashtags (keywords).

Phan et al. present a general framework for classification tasks involving short and sparse text segments. The key component of their framework is collecting a domain relevant "universal dataset" from which hidden topics can be derived and applied to the classification task. Because Twitter posts are so short, a valuable area of future work may be to apply similar techniques to reduce the sparseness of our data.

Sriram and Fuhry present a classification system for Twitter posts into a small set of predefined categories such as news, events, opinions, deals, and private messages. While hashtag recommendation can be seen as a classification problem, our work differs in that we plan to use a large, dynamic set of hashtags as possible categories/ recommendations. Their approach attempts to overcome some of the difficulties in short text classification by

[3] http://www.americannationalcorpus.org/SecondRelease/data/ANC-written-lemma.txt

[4] http://www.cs.princeton.edu/~chongw/slda/

considering author features as well as text-based features. Esparza et al. categorize Twitter posts into five categories based on tweet content: *movies, books, music, apps* and *games*. They manually categorized each tweet, then trained a classifier to predict categories for new tweets. Our work is similar in that we aim to categorize tweets, but we intend to do so using hashtag suggestion and for a larger number of categories. Other attempts at automatic categorization of Twitter data have looked at categorizing a topic by aggregating a number of posts (Sharifi et al. 2010).

Other classification systems have also been applied to Twitter. Go et al. investigate sentiment classification in Twitter posts using a distant supervision technique in which common emoticons serve as noisy labels for training data. While their work does not directly involve hashtags, they describe several pertinent methods for text pre-processing and feature reduction.

Chu et al. describe a system for classifying Twitter users as either humans, bots, or "cyborgs." They incorporate entropy analysis into their approach, but with respect to the delay between posts given a particular user rather than the entropy of the post contents given a hashtag.

4. Results

4.1 Dataset Analysis

Our dataset contains tweets from 5000 Twitter users over the course of almost two months. This includes 1,318,323 tweets. Of these tweets, approximately 15.9% contain hashtags and 4.5% contain multiple hashtags.

Our data contains 321,412 uses of 58,739 distinct hashtags. We counted the number of tweets referencing each hashtag and found that a majority of hashtags only appear in one tweet. On the other hand, there are hundreds of hashtags appearing in at least 100 tweets, but this is reduced to 56 hashtags when the contributions from "spam" accounts are removed. Hashtag frequency between these two extremes is shown in figure 1.

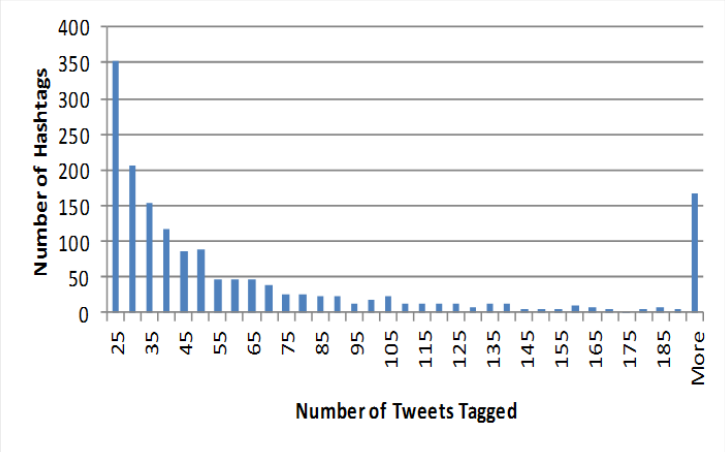

Figure 1

Our investigation indicates that certain hashtags are more likely amenable to our recommendation algorithm than others. Some hashtags, like "#win" are associated with a wide variety of words in referencing tweets. For these hashtags, it appears there is low content similarity between associated tweets. Thus, the word distribution for such a hashtag will probably not indicate a high likelihood for a new query tweet and will not be selected for recommendation by our naive Bayes model. For efficiency's sake and because specificity is usually more important than sensitivity in our recommendations, we allow users of our interactive system (discussed below) to filter out this type of hashtag by adjusting the content similarity parameter.

4.2 Test Implementation

We have developed a test implementation of our algorithm in that provides interactive suggestion of hashtags for a given tweet. Our experience suggests that such a system can in fact suggest relevant hashtags and that a simple heuristic confidence threhsold is able to somewhat prevent irrelevant suggestions from being made, even if no

relevant ones can be found. Our algorithm gives particularly good results for popular topics involving specific people, places, or ideas.

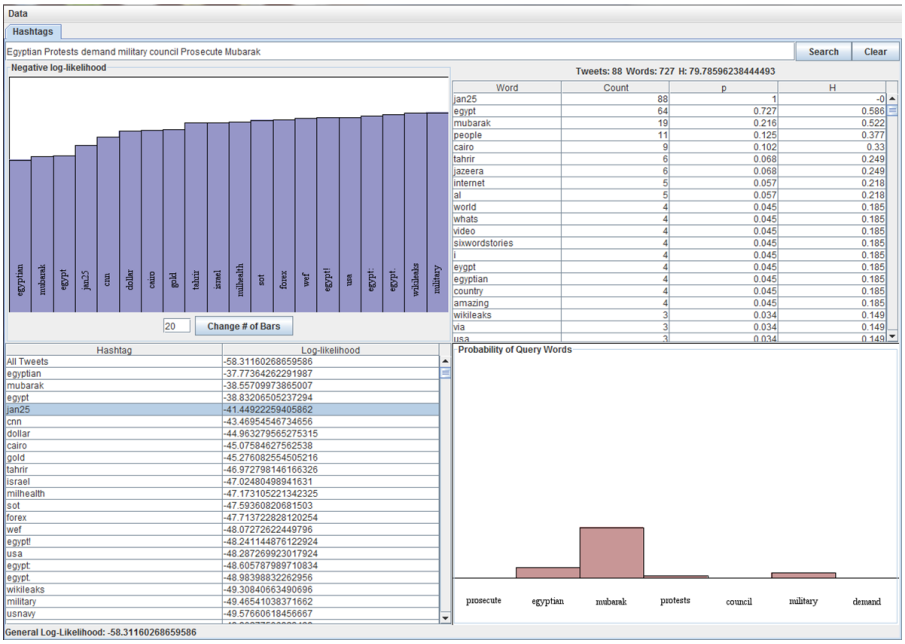

Figure 2: A screenshot of our test implementation. "#jan25" (the beginning of the 2011 Egyptian Revolution) is shown as a relevant hashtag suggestion for a query tweet taken from the public Twitter feed.

4.3 Empirical Evaluation

Evaluating the effectiveness of a relevance-based recommendation system is a non-trivial task because of the inherent difficulty of determining whether or not a suggested hashtag is relevant to the tweet in question. Our ultimate goal is to define relevance as recognized by humans, so a possible course of evaluation is to conduct a user-study to evaluate the effectiveness. Users would be presented with a tweet without any hashtags and a list of possibly-relevant hashtags recommended by our algorithm. Users would then rank the relevancy of hashtags recommended. From this ranking, we could evaluate how meaningful our algorithms' suggestions are to real users.

Though we were unable to conduct a user study during the course of the semester, we did evaluate our recommendation algorithm through hold-out cross validation. We trained our recommendation algorithm using 70% of our data set and used the remaining 30% of our data as a test set. Because we found that much of the test set ended up being tweets from spam accounts, we added the restriction that each user could only contribute 1 tweet to the test set. This decreases performance on cross validation because it is much easier to suggest "relevant" hashtags to a spammer, but we feel it holds more truly to our real-world goals. We removed and stored the hashtags from each tweet and ran our recommendation algorithm on the test set without any hashtags, attempting to replace the missing tags. Throughout the tests, we varied the minimum number of tweets required for a hashtag to be considered valid in order to compare our algorithm's performance at different levels of generality.

We allow our algorithm to produce a ranked list of the top-20 recommended hashtags. For each tweet, we record the rank of the actual hashtag in the list of recommended tags. The best possible rank is 0 and if the hashtag is not present in the list, it is given rank 20. Our reasoning for allowing only this amount of suggestions is that a user is unlikely to read through more than 20 suggestions. In figure 3, we present our results for cross-validation experiments with the minimum tweets required for consideration (min. tweets value) set at 5, 15, 30, 50 and 100. For each min. tweets value and each algorithm described in sections 2.3 and 2.4), we present the mean rank of the original hashtag, median rank of the original hashtag, standard deviation and the percentage of tweets for which the original hashtag was included in the top-20 suggestion list.

	At least 5	At least 15	At least 30	At least 50	At least 100 Tweets

	tweets (3132 hashtags)	tweets (660 hashtags)	tweets (208 hashtags)	Tweets (140 hashtags)	(56 hashtags)
Naive Bayes	mean rank: **14.7** median rank: **20** st. dev.: **8.3** suggested: **32%**	mean rank: **13.6** median rank: **20** st. dev.: **8.6** suggested: **40%**	mean rank: **11.9** median rank: **19** st. dev.: **8.9** suggested: **50%**	mean rank: **10.8** median rank: **11** st. dev.: **8.9** suggested: **58%**	mean rank: **8.6** median rank: **5** st. dev.: **8.4** suggested: **72%**
IDF Term Weighting	mean rank: **15.7** median rank: **20** st. dev.: **7.6** suggested: **27%**	mean rank: **14.1** median rank: **20** st. dev: **8.3** suggested: **37%**	mean rank: **12.7** median rank: **20** st. dev.: **8.7** suggested: **47%**	mean rank: **11.25** median rank: **12** st. dev.: **8.7** suggested: **56%**	mean rank: **9.1** median rank: **6** st. dev.: **8.4** suggested: **70%**
Usage Priors	mean rank: **13.5** median rank: **20** st. dev.: **8.6** suggested: **39%**	mean rank: **12.2** median rank: **20** st. dev.: **8.9** suggested: **48%**	mean rank: **10.4** median rank: **9** st. dev.: **8.9** suggested: **59%**	mean rank: **9.7** median rank: **7** st. dev.: **8.8** suggested: **64%**	mean rank: **7.4** median rank: **3** st. dev.: **8.1** suggested: **78%**
IDF/Priors Combined	mean rank: **14.8** median rank: **20** st. dev.: **8.0** suggested: **32%**	mean rank: **13.4** median rank: **20** st. dev.: **8.5** suggested: **42%**	mean rank: **12.2** median rank: **19** st. dev.: **8.7** suggested: **50%**	mean rank: **11.2** median rank: **12** st. dev.: **8.6** suggested: **57%**	mean rank: **8.1** median rank: **5** st. dev.: **8.1** suggested: **77%**

Figure 3: Cross validation results with testing data limited to 1 tweet per user

4.4 Discussion

As we increase the set of hashtags under consideration our algorithm loses accuracy, as expected. The algorithm simply has more options to choose from and has more opportunities to make mistakes. It is also much more likely that a less common tweet will be incoherent or ill-defined, and thus no algorithm will be able to perform as well when less common hashtags are possible suggestions. Our cross validation experiments will also give less favorable results for less common hashtags because we are only checking the rank our algorithm assigns to the original hashtag. There is a greater chance the algorithm may not suggest specifically the hashtag we are checking for, even if the majority of suggestions are in fact relevant.

Incorporating usage priors resulted in the best mean rank and suggestion percentage, but it seems that IDF weighting consistently decreased our algorithm's performance. This is an interesting and unexpected result. One possible reason that IDF weights do not help is that there is no hard requirement for the rarest words in a tweet to be indicative of its semantic content, especially because the language of Twitter is subject to length constraints. It is also possible that our testing methodology is not favorable for this kind of weighting. We do not include the hashtag itself as a word fed into our suggestion algorithm, and it may be possible that this was by far the most important word in the tweet.

As mentioned previously, an issue with our cross validation experiment is that we are attempting to replace hashtags that have been removed from the tweets in our test set, and in so doing are only considering the rank of the original hashtag. However, we do not consider the case that our algorithm may recommend a more specific or more appropriate hashtag than the tag that was originally given to the tweet. Another aspect of our algorithm's performance not covered by this test is degree to which irrelevant hashtags are suggested by our algorithm. We have tried a heuristic approach to determine a confidence threshold by comparing the likelihood of a tweet under suggested hashtags with the likelihood of the same tweet under our overall vocabulary, but it is difficult to find a way to evaluate the effectiveness of this or any other approach. If we had a computational means to test which hashtags are relevant/irrelevant, our test would also be our solution! Thus, evaluating the proportion of relevant hashtags requires costly human intervention.

Despite these limitations, we believe these results serve as an initial indication of the feasibility of hashtag suggestion and as a justification of our approach. Our experimental data shows our algorithm is able to perform

quite well (78% recovery of the original hashtag) on a subset of the most popular hashtags. In a real-world system, the computational efficiency of our algorithm would also allow for this subset to adapt to the changing trends of Twitter. Again, these tests only check for suggestion of one specific hashtag for each tweet. Our experience with our interactive test implementation has shown it is often the case our algorithm is able to produce many relevant suggestions.

5. Future Work

As results from our static recommendation tool have been promising, we plan to investigate the applicability of our algorithm in a real-world setting. This will involve analyzing the efficiency of our algorithm and the feasibility of real-time updates. We also plan to test our algorithm to determine how long the learned distributions remain valid.

In our real-time system, the "trending topics" on Twitter could also be taken into consideration when determining hashtag relevance. This could be done by adjusting the prior probabilities for each hashtag on the assumption that trending topics should be given preference. Another option would be to present trending topic suggestions separately from normal suggestions to avoid overwhelming regular hashtag suggestions. Either of these strategies would require more frequent updating of the model, but this could be done as a global adjustment and need not be recomputed for individual tweets.

6. Conclusion

As micro-blogging sites, like Twitter, continue to grow in popularity, we are presented with the problem of how to effectively categorize and search for posts. To cope with the volume of information shared daily, Twitter has introduced *hashtags*, keywords prefaced with "#", to help users categorize and search for tweets. To help Twitter users more easily incorporate hashtags into their posts, we have proposed an automatic hashtag suggestion tool that, when given a tweet, will return a short list of relevant hashtags as suggestions.

From our current work, we are hopeful about the success of our hashtag suggestion tool. From the results of our hold-out cross-validation experiments, we believe that the tool could be used with success. We believe a larger training set is would be important for real-world use in order to increase the number of hashtags we can consider recommending while still maintaining a high level of accuracy. Overall, this work makes the following contributions:

- An application of a naive Bayes model to a novel nomain - hashtag suggestion for microblogging sites
- An investigation of the benefits of IDF weighting and usage priors as extensions to naive Bayes in this domain
- A description of preprocessing techniques (filtering tweet content, removing spam tweets) to make Twitter data amenable to machine learning techniques

7. References

Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. 2010. Who is tweeting on Twitter: human, bot, or cyborg?. In *Proceedings of the 26th Annual Computer Security Applications Conference* (ACSAC '10). ACM, New York, NY, USA, 21-30.

S. G. Esparza, M. P. O'Mahony and B. Smyth. 2010. Towards Tagging and Categorisation for Micro-blogs. In *Proceeding of the 21st National Conference on Artificial Intelligence and Cognitive Science* (AICS '10).

A. Go, R. Bhayani and L. Huang. 2009. Twitter Sentiment Classification using Distant Supervision. http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf.

J. Huang, K. M. Thornton, and E. N. Efthimiadis. 2010. Conversational tagging in twitter. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia* (HT '10). ACM, New York, NY, USA, 173-178.

Ide, N., Suderman, K. 2006. Integrating Linguistic References: The American National Corpus Model. In *Proceedings of the Fifth Language Resources and Evaluation Conference (LREC)*, Genoa, Italy.

K. Lee, J. Caverlee, and S. Webb. 2010. Uncovering social spammers: social honeypots + machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '10). ACM, New York, NY, USA, 435-442.

Z. Li, D. Zhou, Y. Juan, and J. Han. 2010. Keyword extraction for social snippets. In *Proceedings of the 19th international conference on World wide web* (WWW '10). ACM, New York, NY, USA, 1143-1144.

C. Marlow, M. Naaman, d. boyd, and M. Davis. 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia* (HYPERTEXT '06). ACM, New York, NY, USA, 31-40.

X. Phan, L. Nguyen, and S. Horiguchi. 2008. Learning to classify short and sparse text \& web with hidden topics from large-scale data collections. In *Proceeding of the 17th International conference on World Wide Web* (WWW '08). ACM, New York, NY, USA, 91-100.

B. Sharifi, M. Hutton and J. Kalita. 2010. Automatic Summarization of Twitter Topics. *National Workshop on Design and Analysis of Algorithms* (NWDAA '10).

B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceeding of the 33rd International ACM SIGIR conference on Research and development in information retrieval*(SIGIR '10). ACM, New York, NY, USA, 841-842.

W. Wu, B. Zhang, and M. Ostendorf. 2010. Automatic generation of personalized annotation tags for Twitter users. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (HLT '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 689-692.

W. Yih, J. Goodman, and V. R. Carvalho. 2006. Finding advertising keywords on web pages. In *Proceedings of the 15th International conference on World Wide Web* (WWW '06). ACM, New York, NY, USA, 213-222.