Evaluating Methods for Summarizing Twitter Posts

Gary Beverungen
St. Mary's College of Maryland
16800 Point Lookout Rd.
St. Mary's City, MD
gebeverungen@smcm.edu

Jugal Kalita
University of Colorado at Colorado Springs
1420 Austin Bluffs Pkwy
Colorado Springs, CO
kalita@eas.uccs.edu

ABSTRACT

Microblogs like Twitter[1] are becoming increasingly popular and serve as a source of ample data on breaking news, public opinion, etc. However, it can be hard to find relevant, meaningful information from the enormous amount of activity on a microblog. Previous work has explored the use of clustering algorithms to create multi-post summaries as a way of understanding the vast amount of microblog activity. Clustering of microblog data is notoriously difficult because of non-standard orthography, noisiness, limited sets of features, and ambiguity as to the correct number of clusters. We examine several methods of making standard natural language processing techniques more amenable to the domain of Twitter including normalization, term expansion, improved feature selection, noise reduction, and estimation of the number of natural clusters in a set of posts. We show that these techniques can be used to improve the quality of extractive summaries of Twitter posts, providing valuable tools for understanding and utilizing microblog data.

Keywords

microblogs, normalization, extractive summarization, term expansion, clustering

1. INTRODUCTION

Microblogging is a relatively new form of communication, providing both new opportunities and new challenges for Natural Language Processing (NLP). Microblogs such as Twitter may have as many as 2.5 million posts per day about a variety of topics and from a diverse set of users. One could mine this data to discover public opinion [13], breaking news, sentiment analysis [14], or even predict the stock market [4]. Clearly there is the potential for vast amounts of useful data to be found from microblog posts. Unfortunately, standard approaches to NLP often fail in the domain of microblog posts, and it is not clear which techniques for extracting and utilizing microblog data are most useful. Clearly it is necessary to determine which tools will be most helpful in making use of microblog data. We explore the use of clustering as a means of detecting important subtopics in sets of Twitter posts and selecting posts which are representative of the activity on that topic.

Several obstacles stand in the way of processing microblog posts such as those on Twitter. First, Twitter posts are highly non-standard. While most standard NLP techniques were developed for long, structured, grammatical text, Twitter is short, colloquial, and ungrammatical. Users frequently misspell words either unintentionally (*teh, waht*) or intentionally, by expanding words, abbreviating words, or using lexical/numeric substitutions (*loooovveeeee, rly, c u l8r*). Twitter posts also frequently contain other non-standard tokens such as acronyms (*lol, smh*), hash tags (*#beatcancer, #iusuallylieabout*), user tags (*@nina1983*), or Twitter specific terminology indicating "re-tweeted" posts (*RT*) and trending topics (*TT*). This poses a problem for NLP techniques, since two posts with alternate spellings of some word may not be considered related, when in fact they are. While normalization of Twitter posts remains a difficult problem, progress has been made by those like Kaufmann and Kalita [12] and Han and Baldwin [10]. We use the techniques developed in Kaufmann and Kalita to normalize Twitter posts and hopefully improve the effectiveness of other NLP techniques.

Second, Twitter posts are very short, no more than 140 characters and typically not more than ten words or so. Unfortunately, this means that Twitter posts are feature sparse, and that comparisons between posts will be difficult. This is especially problematic for clustering, which is highly sensitive to the features chosen for comparison. This problem could be alleviated by expanding terms in the twitter posts to include relevant similar terms (essentially adding additional features), selecting more descriptive features (e.g. just named entities), using n-grams instead of unigrams, or some combination of the three.

Additionally, even with good features Twitter posts could be hard to cluster. One challenge of clustering in general is determining how many clusters are "inherent" in the data set. Most clustering algorithms require the number of clusters to be specified ahead of time, but it is not always obvious what that number should be. Choosing incorrectly can lead to suboptimal clustering of data, splitting coherent clusters into multiple clusters or grouping distinct clusters into one.

[1] http://www.twitter.com

WSDM'11, February 9–12, 2011, Hong Kong, China.

Luckily, several methods for determining the "correct" number of clusters exist, including those by Tibshirani et al.[18] and Ben-Hur et al. [2]. Ideally, these methods would allow us to determine the number of salient subtopics and generate summaries that more accurately reflect the posts.

However, even if we can identify relevant subtopics in a set of Twitter posts, there will still be many posts which do not fit well into any subtopic or cluster. Undoubtedly, there will be many posts that are largely unrelated to other posts in the data set. These outliers may have a negative impact on the ability of the clustering algorithms to correctly identify subtopics, and hurt summarization overall. Thus it may be beneficial to try to remove outliers and other noisy posts from the data set before using other NLP techniques.

This paper presents preliminary attempts at making the domain of microblog posts more ammenable to NLP techniques by combining techniques for tackling each of the challenges described above.

2. PREVIOUS WORK

2.1 Normalizing Posts

As mentioned, microblog posts are notoriously hard to process computationally, containing frequent misspellings, unfamiliar named entities, OOV words, improvised abbreviations, slang, and novel lexography. Converting the post to standard English would improve processing. Research has used the noisy channel model, wherein normalizing noisy text T to a standard form S by assuming T is an error of S. The most likely S is found by finding the probability of T being an error of S time the probability of S occurring. In other words, $S_{max} = argmax(P(S|T)) = argmax(P(T|S)P(S))$. Various approaches have been made to characterize the error model, $P(T|S)$, including edit distance [5] and letter transformation [7]. Machine translation may be able to assist with normalization [12]. Part-of-speech (POS) parsers created for Twitter [9] might be able to give additional information about ambiguous words. We utilize the normalization tool developed by Kaufmann et. al. [12].

2.2 Clustering

Previous work by Sharifi et al. [17, 16] has explored the topic of multi-post extractive microblog summarization. They explored frequency based, graph based, and cluster based methods of selecting multiple posts that conveyed information about a given topic without being redundant. They found that ROUGE-N scores and human evaluation did not provide an obvious choice of one summarizer over another [1]. In fact, most multi-post summarizers did not perform significantly differently from a simple Most Recent summarizer. However, clustering algorithms could be improved in a number of ways, as described here.

2.2.1 Determining the Number of Clusters

The clustering algorithms used in Sharifi et al. [1] were fairly basic, and there remains room for improvement. One limitation of the clustering algorithms used in Sharifi et al. [17] is that they generate a specific number of clusters that must be determined before running the algorithm. This means that an arbitrary number of clusters must be chosen for a set of microblog posts, regardless of the actual distribution of posts. Although Sharifi et al. determined that most users thought four clusters was appropriate, the optimal number of clusters likely varies from topic to topic. Ben-Hur et al. have used a stability based method for determining the number of clusters in data, building off of the idea that a good clustering should be stable, consistent, and robust to noise. [2] Alternatively, Tibshirani et al. have used the gap statistic, a measure of within cluster dispersion of a clustering compared to an expected value, to determine the correct number of clusters. [18] Using these methods, we may be able cluster microblog posts in a way that more accurately reflects relevant sub-topics.

2.2.2 Choosing Features

In addition to using different clustering techniques, it may be possible to improve results by improving the way in which posts are compared. It may be the case that simple word level similarity doesn't capture what humans perceive to be the important aspects of sentence similarity. Hatzivassiloglou et al. have shown that including information about the NP heads, named entities, events, and other information included in the sentences, it is possible to improve the quality of clusters [11]. Alternatively, limiting features to certain parts of speech (Nouns, Verbs) may significantly cut down on the extraneous features in the posts. Using a Part-of-Speech (POS) tagger, like the one developed by Gimpel et al. would allow us to do just that. [9] Additionally, some authors have looked at expanding posts by adding highly related terms, thus overcoming the feature sparsity of Twitter posts. Perez-Tellez et al. use pointwise mutual information to determine which words are most similar to ones already in the post. [15] Chen et al. use a similar technique, but also use inforation from Wikipedia to expand posts. [6] By improving the feature vector to more accurately reflect perceived similarity, we may be able to improve the effectiveness of clustering, and thus, the quality of the resulting summary.

3. METHODS

3.1 Data

Our data set includes 50 topics selected from Twitter's list of Trending Topics. For each topic, posts are selected by taking 1500 posts from the Twitter API and processing them as follows:

1. Convert HTML encoded characters to ASCII.
2. Discard any posts that aren't in English. (Defined as containing at least 40% English Words.)
3. Discard a post if there has already been another post by the same user.
4. Discard a post if it is spam.
5. Reduce number of posts by taking the most recent 100.

3.2 Normalization

This process can is described in greater detail in [**?**]Sharifi-MS.

For normalization, we utilize the normalized developed by Kaufmann and Kalita. [12] Their method uses a combination of lexical normalization, syntactic disambiguation, and statistical machine translation to convert Twitter posts from their noisy, non-standard, ungrammatical form to something resembling more standard texts.

3.3 Clustering

Previously, Sharifi et al. have experimented with several types of clustering algorithms. [1] They found that, of the algorithms they tested, bisecting k-means was the most effective at producing summaries. Thus, that is the clustering we will be using for our purposes. For the remainder of this paper, when we refer to clustering a set of Twitter posts, we a referring to bisecting k-means clustering.

3.4 Determining the Optimal Number of Clusters

3.4.1 Non-counting Method

As a control, and to facilitate comparing our results with those in Sharifi et al. [1], we implement a trivial cluster counter which determines that there should be four clusters regardless of the data. Four clusters was chosen to imitate the method of clustering in Sharifi et al.

3.4.2 Stability Based Method

In order to determine the number of "inherent" clusters in a set of Twitter posts, we chose to implement Ben-Hur et al.'s stability based method. [2] The crux of the algorithm is that a good clustering of data should be relatively stable and robust to noise. While a suboptimal clustering may be clustered differently every time the clustering algorithm is run, a good clustering should produce roughly the same result every time. Additionally, even if a small amount of the data is removed, as long as all the clusters are adequately represented, a good clustering should still be able to find the "correct" clutsering. Thus, but clustering random subsamples of a data set and comparing the similarity of the clusterings for different numbers of clusters, we should be able to get a good estimate for which number of clusters is the most stable, and thus most apt to fit the data. A description of the implementation of the algorithm is as follows:

```
Given: Data Set ← $X, Float \leftarrow f$ for k = $2tok_{max}$ do
  for $i = 1$ to num iterations do
    Sub1 = subsample(X, f)
    Sub2 = subsample(X, f)
    Cluster1 = cluster(Sub1)
    Cluster1 = cluster(Sub2)
    Similarity(k,i) = Sim(Cluster1, Cluster2)
  end for
end for
```

: Algorithm for determining the average similarity of two samples of the data set for each number of clusters, k

The similarity function returns a measure of the similarity of two clusterings. To compute said similarity, first we construct an nn matrix, where n is the number of posts in the clusterings and each entry in the matrix, (i,j) is defined as:

1: if posts i and j are in the same cluster

0: otherwise.

Once we have obtained matricies for each clustering, M_1 and M_2 respectively, we let N_{ij} be the number of entries in which M_1 and M_2 have the values i and j, respectively. The similarity measure is then defined as the Jaccard coefficient:

$$\frac{N_{11}}{N_{01} + N_{10} + N_{11}}. \quad (1)$$

Thus, after running the algorithm described above, we have a list of i similarities between clusterings for each possible number of clusters k. The k we ultimately choose is the one for which the average of each of the i similarities is the greatest.

3.4.3 Gap Statistic

As an alternative to the Stability Based method we implement the Gap Statistic method described in Tibshirani et al. [18] The gap statistic makes use of the measure of within cluster dispersion. For each cluster C_r in a clustering, $n_r = |C_r|$ and $D_r = \Sigma_{i,i' \in C_r} d_{ii'}$ where $d_{ii'}$ is the squared Euclidean distance between posts i and i'. Within cluster dispersion for a clustering of k clusters is then

$$W_k = \Sigma_{r=1}^{k} \frac{1}{2n_r} D_r. \quad (2)$$

Conventional wisdom has it that when there is a sharp decrease in the within cluster dispersion, the correct number of clusters has been found. Tibshirani et al. calculate an expected withing cluster dispersion using a null reference data set, and determine how far below that value the real data falls. As our null reference, we generate a set of posts each with length equal to the average length of posts in the real data set. Each word in each null reference post is chosen at random, uniformly across all words seen in the real data set. For each value of k, we cluster the null reference set b times and compute the withing cluster dispersion, W_k^*, for each clustering, as well as average within cluster dispersion, $W_{k\,avg}^*$, and the standard deviation of the dispersions, $W_{k\,stddev}^*$. We then cluster the real data set and calulate the within cluster dispersion, W_k. The gap statistic is defined as

$$Gap(k) = W_{k\,avg}^* - W_k \quad (3)$$

and the chosen value of k is the smallest value for which the following inequality holds

$$Gap(k) \geq Gap(k+1) - W_{k\,stddev}^*. \quad (4)$$

3.5 Feature Selection

3.5.1 Term Expansion

In order to overcome the small size of microblog posts, we expand the post to include terms similar to other terms in the post. We follow the methods described in Tellez-Perez et al. [15] Given a set of posts, $X = p_1, p_2, \ldots p_n$, we find the Pointwise Mutual Information (PMI) for each pair of terms, $t_i, t_j \in p_n$ found in the posts

$$PMI(t_i, t_j) = \frac{P(t_i, t_j)}{P(t_i)P(t_j)} \quad (5)$$

For each term t_i in a post p_n, Tellez-Perez et al. find the PMI between that term and each other term found in X. Any term t_j for which $PMI(t_i, t_j)$ is greater than some threshold value is added to the post p_n. This process is repeated for each post in X. While Tellez-Perez et al. set the threshold value manually, we found that the PMI between a pair of posts varied greatly depending on the set of posts X. Depending on the topic, the average PMI and the variation in PMI could vary greatly. Thus, we set it as the one standard of deviation above the average PMI of all pairs of posts in X.

3.5.2 N-Grams

In effort to pick up on the more nuanced relationships between terms in a post, we used n-grams as features instead of simple unigrams. Posts that contain the same words in the same order are more likely to be related than posts that simply have the same words contained somewhere in the post. By comparing posts using n-grams instead of or in addition to unigrams we may be able to get a more nuanced measure of similarity between posts. For a post p with terms $t_1, t_2, \dots t_i \in p$ we define the n-grams of p as

$$n_1 = (t_1, t_2, \dots t_n), n_2 = (t_2, t_3, \dots t_{n+1}), \dots n_{i-n} = (t_{i-n+1}, \dots t_{i-1}, t_i).$$

We experiment with using unigrams, bigrams, and trigrams, and a combination of unigrams, bigrams, and trigrams.

3.6 Noise Reduction

As another method of improving the validity of cluster, we investigate noise reduction. The goal of noise reduction is to remove posts that do not fit well into any cluster. This can be determined in a number of ways, but for our purposes we define this as any post for which the average distance to another post is more than one standard of deviation more than the population average. Average distance from post i to another post is defined as:

$$D_i =$$

$1n\Sigma_{j \in S} d(i,j)(6)$where S is the set of posts to be summarized, $n = |S|$, and d(x,y) is the squared Euclidean distance between posts i and j. This value is calculated for every post in S, and any post for which the value is one standard of deviation above the average value is removed.

3.7 Evaluation

3.7.1 Cluster Validity

The end goal of each of these methods is to produce good clusters of Twitter data and thus descriptive summaries of each Twitter topic. Therefore, we will evaluate the effectiveness of each technique in terms of the cluster quality and quality of the overall summary. To measure the validity of a particular clustering we use a modified Dunn's Index, as described in Bezdek Pal. [3] Dunn's Index is the ratio of the minimum between cluster distance to the maximum within cluster dispersion. The assumption is that a good clustering will produce well separated clusters and clusters that are densely packed. While Bezdek Pal offer several definitions of cluster distance and dispersion, we use the following definitions, which they cite as among the most effective. Intercluster distance $D_{is,t}$ of two clusters, S and T, is defined as

$$D_{is,t} = \frac{1}{|S||T|} \Sigma_{x \in S,y} d(x, y) \tag{7}$$

where d(x,y) is the squared Euclidean distance between posts x and y. Within cluster dispersion, D_{wS}, of a cluster, S, is defined as

$$D_{wS} = 2(\frac{\Sigma_{x \in S} d(x, \mu_s)}{|S|}) \tag{8}$$

where μ_s is the mean of the feature vectors of the posts in S. Thus, D_w is essentially twice the average distance of a post in S from the mean of S. Given a set of clusters $X = C_1, C_2, \dots C_n$ the modified Dunn's Index is then

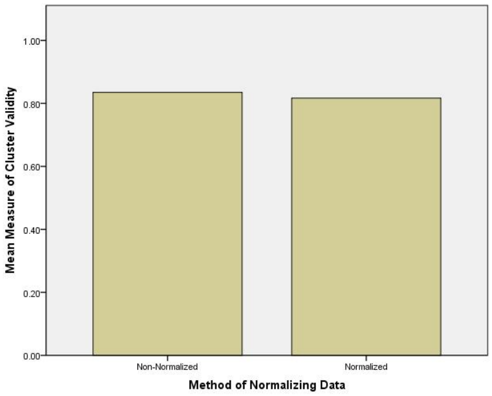
$$DI = \frac{argmin_{s,t \in X}(D_{is,t})}{argmax_{s \in X}(D_{ws})}. \tag{9}$$

4. RESULTS

To determine the effectiveness of each strategy, we perform an ANOVA and test for main effects for each strategy (normalization method, cluster counting method, noise reduction method, and feature selection). Thus, we perform the clustering process for each of the 50 data sets for every combination of methods (2 normalization methods 3 cluster counting methods 2 noise reduction methods 5 feature selection methods = 60 total methods of clustering). SPSS version 19 was used to perform the ANOVA with $\alpha = .05andpost-hoctestsperformedusingTukey'sHSD.Theresultsbelowshowthemai$

4.1 Normalization

Normalization had a small but statistically significant effect on overall cluster validity. Surprisingly, normalizing the posts led to decreased clustering performance. Normalized posts had an average cluster validity of .817, whereas the non-normalized posts had a validity of .835, as seen in Figure 4.1. Thus, we note that normalizing posts does not drastically impact the clustering of Twitter data.



Graph.jpg

Figure 1: The mean cluster validity for normalized and non-normalized posts. The effect is small but significant.

4.2 Cluster Counting Method

We found that the Gap Statistic method of evaluating the number of clusters significantly outperformed both the baseline and stability based methods. Surprisingly, the stability

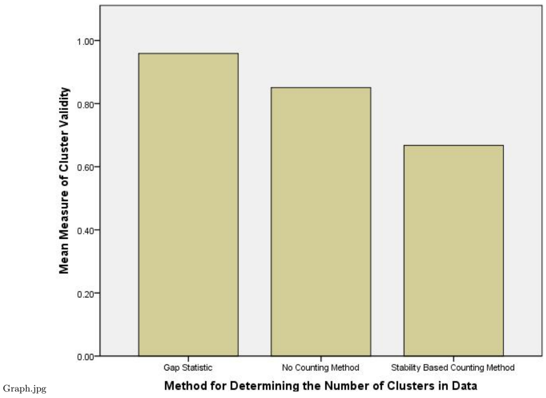


Figure 2: The mean cluster validity for each of the different methods of counting the correct number of clusters. There were significant differences between each method of counting clusters.

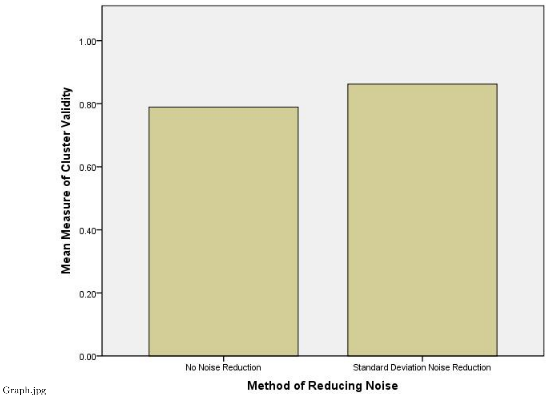


Figure 3: The mean cluster validity with and without noise reduction. Noise reduction significantly improved results.

based method produced the worst clusters. Upon further analysis, we found that the stability based method tended to favor larger numnbers of clusters, drastically decreasing the intercluster distance and increasing Dunn's Index. The gap statistic produced an average cluster validity of .959, the stability based method .668, and the baseline method .851, with significant differences between each method, as shown in Figure 4.2

4.3 Noise Reduction

The noise reduction method described above significantly improved cluster validity scores. The noise reduction method had an average cluster validity score of .862 as opposed to the a cluster validity of .789 for clusters with no noise reduction. These results can be seen in figure 4.3.

4.4 Feature Selection

Two feature selection methods showed significant improvement over the rest, bigram feature selection and trigram feature selection. A table with the means for each feature selection method can be found in table 4.4. Bigrams performed significantly better than either unigrams, the combination of unigrams, bigrams and trigrams, and term expansion techniques. Likewise, Trigrams significantly outperformed unigrams, the combination of unigrams, bigrams and trigrams, and term expansion techniques. There was no significant difference between bigrams and trigrams, nor were there significant differences between any of the three remaining techniques. A graph of the results can be seen in Figure 4.4.

Feature Selection Method	Average Cluster Validity
Unigrams	.792
Bigrams	.869
Trigrams	.868
Combination	.806
Term Expansion	.793

Table 1: The mean cluster validities for each method of feature selection.

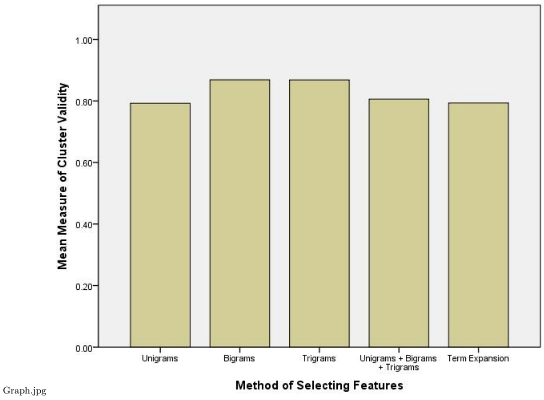


Figure 4: The mean cluster validity for each method of feature selection. Both bigrams and trigrams performed significantly better than trigrams, combination, and term expansion. There were no other significant differences.

5. FUTURE WORK

Once we have obtained a multi-post extractive summary from a set of microblog posts, there remains the question of how to order the posts in a way that maximizes the coherence of the overall summary. We have done some preliminary analysis of the feasibility of ordering posts, but found that, when humans were asked to manually order posts selected for a summary, the inter-rater correspondence was only slightly more than the value expected by random ordering. However, the assumption that there is a single "correct" ordering is perhaps unfounded. There may be several plausible, coherent orderings for a set of posts. More research is needed to determine a good method for measuring the coherency of a summary and finding good orderings.

We have focused exclusively on k-means clustering, specifically bisecting k-means. However, other types of clustering algorithms exist. Heirarchical or density based algorithms could just as easily be used to find structure in microblog data. Additionally, density based clustering has the advantage of being fairly robust to noise and obviates the need to choose a number of clusters. However, density based clustering has its own challenges and parameter that need to be fine tuned. Since all of the mentioned clustering algorithms still depend on good features and similarity measures, most of the work in this paper could still apply, but further investigation is necessary to determine how efffective these other clustering algorithms are at summarizing microblog data.

With term expansion, we have used only the pointwise mutual information (PMI) technique, but other methods of expanding the number of features in a post exist. Future work could look at adding WordNet synonyms and/or hypernyms to the posts to increase the number of features. Additionally, some authors have looked at using linked web content to find more information about a particular post. [8] Lastly, we have looked at term expansion for the addition of unigrams based on the PMI with other unigrams. However, this need not be the case. If a particular n-gram has a high measure of PMI with any other n-gram, there is grounds for adding it as well. Given the success of bigrams and trigrams in generating good clustering, it might be worthwhile to look into term expansion with bigrams and trigrams as well.

6. CONCLUSION

In this paper, we have explored several means of mitigating the difficulty of processing microblog posts. We have examined methods of normalizing posts as a way of reducing noise, extracting descriptive features from microblog posts, and improving the effectiveness of existing clustering techniques. As a result, we have generated relatively descriptive summaries of particular topics in microblogs. Furthermore, the techniques we have examined here could be used to make many other NLP techniques more effective in the microblog domain.

Acknowledgement

The research reported in this document has been funded partially by NSF grants CNS-0958576 and CNS-0851783.

7. REFERENCES

[1] David Inouye Beaux Sharifi and Jugal Kalita. Extractive summarization of twitter microblogs. *Under Revision for ACM Transactions for Speech and Language Processing*, 2011.

[2] Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[3] J. C. Bezdek and N. R. Pal. Some new indexes of cluster validity. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 28(3):301–315, January 1998.

[4] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *CoRR*, abs/1010.3003, 2010.

[5] Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. pages 286–293, 2000.

[6] Qing Chen, Timothy Shipper, and Latifur Khan. Tweets mining using wikipedia and impurity cluster measurement. In *ISI'10*, pages 141–143, 2010.

[7] Bingqing Wang Fei Liu, Fuliang Weng and Yang Liu. Insertion, deletion, or substitution? normalizing text messages without pre-categorization nor supervision. To be published in the proceedings of the Association of Computational Linguistics, 2011.

[8] Yang Liu Fei Liu and Fuliang Weng. Why is "sxsw" trending? exploring multiple text sources for twitter topic summarization. To be published in the proceedings of the Association of Computational Linguistics, 2011.

[9] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *ACL (Short Papers)*, pages 42–47. The Association for Computer Linguistics, 2011.

[10] Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a # twitter. 2011.

[11] Vasileios Hatzivassiloglou, Luis Gravano, and Ankineedu Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *SIGIR*, pages 224–231, 2000.

[12] Joseph Kaufmann and Jugal Kalita. Syntactic normalization of twitter messages. pages 149–158, December 2010.

[13] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.

[14] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[15] Fernando Perez-Tellez, David Pinto, John Cardiff, and Paolo Rosso. On the difficulty of clustering company tweets. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, SMUC '10, pages 95–102, New York, NY, USA, 2010. ACM.

[16] B. Sharifi, M.-A. Hutton, and J.K. Kalita. Experiments in microblog summarization. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 49 –56, aug. 2010.

[17] Beaux Sharifi. Automatic microblog classification and summarization. Master's thesis, University of Colorado at Colorado Springs, 2010.

[18] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the gap statistic. 63:411–423, 2000.