Alleviating Data Sparsity for Twitter Sentiment Analysis

Hassan Saif
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
h.saif@open.ac.uk

Yulan He
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
y.he@open.ac.uk

Harith Alani
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA, UK
h.alani@open.ac.uk

ABSTRACT

Twitter has brought much attention recently as a hot research topic in the domain of sentiment analysis. Training sentiment classifiers from tweets data often faces the data sparsity problem partly due to the large variety of short and irregular forms introduced to tweets because of the 140-character limit. In this work we propose using two different sets of features to alleviate the data sparseness problem. One is the semantic feature set where we extract semantically hidden concepts from tweets and then incorporate them into classifier training through interpolation. Another is the sentiment-topic feature set where we extract latent topics and the associated topic sentiment from tweets, then augment the original feature space with these sentiment-topics. Experimental results on the Stanford Twitter Sentiment Dataset show that both feature sets outperform the baseline model using unigrams only. Moreover, using semantic features rivals the previously reported best result. Using sentiment-topic features achieves 86.3% sentiment classification accuracy, which outperforms existing approaches.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*

General Terms

Algorithms, Experimentation

Keywords

Microblogs, Sentiment Analysis, Opinion Mining, Twitter, Semantic Smoothing, Data Sparsity

1. INTRODUCTION

Few years after the explosion of Web 2.0, microblogs and social networks are now considered as one of the most popular forms of communication. Through platforms like Twitter and Facebook, tons of information, which reflect people's opinions and attitudes, are published and shared among users everyday. Monitoring and analysing opinions from social media provides enormous opportunities for both public and private sectors. for private sectors, it has been observed [21, 22] that the reputation of a certain product or company is highly affected by rumours and negative opinions published and shared among users on social networks. Understanding this observation, companies realize that monitoring and detecting public opinions from microblogs leads to building better relationships with their customers, better understanding of their customers' needs and better response to changes in the market. For public sectors, recent studies [3, 9] show that there is a strong correlation between activities on social networks and the outcomes of certain political issues. For example, Twitter and Facebook were used to organise demonstrations and build solidarity during Arab Spring of civil uprising in Egypt, Tunisia, and currently in Syria. One week before Egyptian president's resignation the total rate of tweets about political change in Egypt increased ten-fold. In Syria, the amount of online content produced by opposition groups in Facebook increased dramatically.

Twitter, which is considered now as one of the most popular microblogging services, has attracted much attention recently as a hot research topic in sentiment analysis. Previous work on twitter sentiment analysis [5, 13, 2] rely on noisy labels or distant supervision, for example, by taking emoticons as the indication of tweet sentiment, to train supervised classifiers. Other work explore feature engineering in combination of machine learning methods to improve sentiment classification accuracy on tweets [1, 10]. None of the work explicitly addressed the data sparsity problem which is one of the major challenges facing when dealing with tweets data.

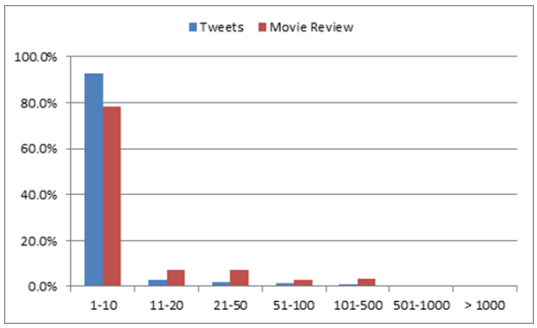


Figure 1: Word frequency statistics.

Figure 1 compares the word frequency statistics of the tweets data we used in our experiments and the movie review data[1]. X-axis shows the word frequency interval, e.g., words occur up to 10 times

[1] http://www.cs.cornell.edu/People/pabo/movie-review-data/

Published as part of the #MSM2012 Workshop proceedings, available online as CEUR Vol-838, at: http://ceur-ws.org/Vol-838
#MSM2012, April 16, 2012, Lyon, France.

(1-10), more than 10 times but up to 20 times (10-20), etc. Y-axis shows the percentage of words falls within certain word frequency interval. It can be observed that the tweets data are sparser than the movie review data since the former contain more infrequent words, with 93% of the words in the tweets data occurring less than 10 times (cf. 78% in the movie review data).

One possible way to alleviate data sparseness is through word clustering such that words contributing similarly to sentiment classification are grouped together. In this paper, we propose two approaches to realise word clustering, one is through semantic smoothing [17], the other is through automatic sentiment-topics extraction. Semantic smoothing extracts semantically hidden concepts from tweets and then incorporates them into supervised classifier training by interpolation. An inspiring example for using semantic smoothing is shown in Figure 2 where the left box lists entities appeared in the training set together with their occurrence probabilities in positive and negative tweets. For example, the entities "*iPad*", "*iPod*" and "*Mac Book Pro*" appeared more often in tweets of positive polarity and they are all mapped to the semantic concept "*Product/Apple*". As a result, the tweet from the test set "*Finally, I got my iPhone. What a product!*" is more likely to have a positive polarity because it contains the entity "*iPhone*" which is also mapped to the concept "*Product/Apple*".

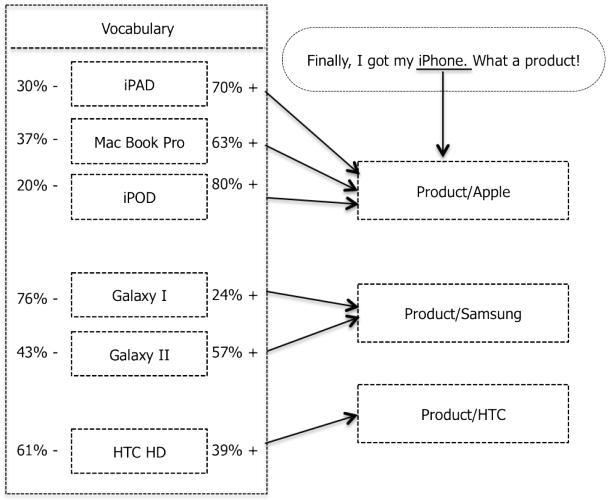


Figure 2: Incorporating semantic concepts for sentiment classification.

We propose a semantic interpolation method to incorporate semantic concepts into sentiment classifier training where we interpolate the original unigram language model in the Naïve Bayes (NB) classifier with the generative model of words given semantic concepts. We show on the Stanford Twitter Sentiment Data [5] that simply replaces words with their corresponding semantic concepts reduces the vocabulary size by nearly 20%. However, the sentiment classification accuracy drops by 4% compared to the baseline NB model trained on unigrams solely. With the interpolation method, the sentiment classification accuracy improves upon the baseline model by nearly 4%.

Our second approach for automatic word clustering is through sentiment-topics extraction using the previously proposed joint sentiment-topic (JST) model [11]. The JST model extracts latent topics and the associated topic sentiment from the tweets data which are subsequently added into the original feature space for supervised classifier training. Our experimental results show that NB learned from these features outperforms the baseline model trained on unigrams only and achieves the state-of-the-art result on the original test set of the Stanford Twitter Sentiment Data.

The rest of the paper is organised as follows. Section 2 outlines existing work on sentiment analysis with focus on twitter sentiment analysis. Section 3 describes the data used in our experiments. Section 4 presents our proposed semantic smoothing method. Section 5 describes how we incorporate sentiment-topics extracted from the JST model into sentiment classifier training. Experimental results are discussed in Section 6. Finally, we conclude our work and outline future directions in Section 7.

2. RELATED WORK

Much work has been done in the field of sentiment analysis. Most of the work follows two basic approaches. The first approach assumes that semantic orientation of a document is an averaged sum of the semantic orientations of its words and phrases. The pioneer work is the point-wise mutual information approach proposed in Turney [20]. Also work such as [6, 8, 19, 16] are good examples of this lexical-based approach. The second approach [15, 14, 4, 23, 12] addresses the problem as a text classification task where classifiers are built using one of the machine learning methods and trained on a dataset using features such as unigrams, bigrams, part-of-speech (POS) tags, etc. The vast majority of work in sentiment analysis mainly focuses on the domains of movie reviews, product reviews and blogs.

Twitter sentiment analysis is considered as a much harder problem than sentiment analysis on conventional text such as review documents, mainly due to the short length of tweet messages, the frequent use of informal and irregular words, and the rapid evolution of language in Twitter. Annotated tweets data are impractical to obtain. A large amount of work have been conducted on twitter sentiment analysis using noisy labels (also called distant supervision). For example, Go et al. [5] used emoticons such as ":-)" and ":(" to label tweets as positive or negative and train standard classifiers such as Naïve Bayes (NB), Maximum Entropy (MaxEnt), and Support Vector Machines (SVMs) to detect the sentiments of tweets. The best result of 83% was reported by MaxEnt using a combination of unigrams and bigrams. Barbosa and Feng [2] collected their training data from three different Twitter sentiment detection websites which mainly use some pre-built sentiment lexicons to label each tweet as positive or negative. Using SVMs trained from these noisy labeled data, they obtained 81.3% in sentiment classification accuracy.

While the aforementioned approaches did not detect neutral sentiment, Pak and Paroubek [13] additionally collected neutral tweets from Twitter accounts of various newspapers and magazines and trained a three-class NB classifier which is able to detect neutral tweets in addition to positive and negative tweets. Their NB was trained with a combination of n-grams and POS features.

Speriosu et al. [18] argued that using noisy sentiment labels may hinder the performance of sentiment classifiers. They proposed exploiting the Twitter follower graph to improve sentiment classification and constructed a graph that has users, tweets, word unigrams, word bigrams, hashtags, and emoticons as its nodes which are connected based on the link existence among them (e.g., users are connected to tweets they created; tweets are connected to word uni-

grams that they contain etc.). They then applied a label propagation method where sentiment labels were propagated from a small set of of nodes seeded with some initial label information throughout the graph. They claimed that their label propagation method outperforms MaxEnt trained from noisy labels and obtained an accuracy of 84.7% on the subset of the twitter sentiment test set from [5].

There have also been some work in exploring feature engineering to improve the performance of sentiment classification on tweets. Agarwal et al. [1] studied using the feature based model and the tree kernel based model for sentiment classification. They explored a total of 50 different feature types and showed that both the feature based and tree kernel based models perform similarly and they outperform the unigram baseline.

Kouloumpis et al. [10] compared various features including n-gram features, lexicon features based on the existence of polarity words from the MPQA subjectivity lexicon[2], POS features, and microblogging features capturing the presence of emoticons, abbreviations, and intensifiers (e.g., all-caps and character repetitions). They found that micoblogging features are most useful in sentiment classification.

3. TWITTER SENTIMENT CORPUS

In the work conducted in this paper, we used the Stanford Twitter Sentiment Data[3] which was collected between the 6th of April and the 25th of June 2009 [5]. The training set consists of 1.6 million tweets with the same number of positive and negative tweets labelled using emoticons. For example, a tweet is labelled as positive if it contains :), :-), :), :D, or =) and is labelled as negative if it has :(, :-(, or : (, etc. The original test set consists of 177 negative and 182 positive manually annotated tweets. In contrast to the training set which was collected based on specific emoticons, the test set was collected by searching Twitter API with specific queries including products' names, companies and people.

We built our training set by randomly selecting 60,000 balanced tweets from the original training set in the Stanford Twitter Sentiment Data. Since the original test set only contains a total of 359 tweets which is relatively small, we enlarge this set by manually annotating more tweets. To simplify and speed up the annotation efforts, we have built Tweenator[4], a web-based sentiment annotation tool that allows users to easily assign a sentiment label to tweet messages, i.e. assign a negative, positive or neutral label to a certain tweet with regards to its contextual polarity. Using Tweenator, 12 different users have annotated additional 641 tweets from the original remaining training data. Our final test set contains 1,000 tweet messages with 527 negative and 473 positive.

It is worth mentioning that users who participated in the annotation process have reported that using the annotation interface of Tweenator, as shown in Figure 3-a, they were able to annotate 10 tweet messages in 2 to 3 minutes approximately.

Recently, we have added two new modules to Tweenator by implementing our work that will be described in Section 4. The first module (see Figure 3-b) provides a free-form sentiment detection, which allows users to detect the polarity of their textual entries. The second module is the opinionated tweet message retrieval tool (see Figure 3-c) that allows to retrieve negative/positive tweets towards a specific search term. For example, a user can retrieve opinionated tweet messages about the search term "*Nike*".

4. SEMANTIC FEATURES

Twitter is an open social environment where there are no restrictions on what users can tweet about. Therefore, a huge number of infrequent named entities, such as people, organization, products, etc., can be found in tweet messages. These infrequent entities make the data very sparse and hence hinder the sentiment classification performance. Nevertheless, many of these named entities are semantically related. For example, the entities "*iPad*" and "*iPhone*" can be mapped to the same semantic concept "*Product/Apple*". Inspired by this observation, we propose using semantic features to alleviate the sparsity problem from tweets data. We first extract named entities from tweets and map them to their corresponding semantic concepts. We then incorporate these semantic concepts into NB classifier training.

4.1 Semantic Concept Extraction

We investigated three third-party services to extract entities from tweets data, Zemanta,[5] OpenCalais,[6] and AlchemyAPI.[7] A quick and manual comparison of a randomly selected 100 tweet messages with the extracted entities and their corresponding semantic concepts showed that AlchemyAPI performs better than the others in terms of the quality and the quantity of the extracted entities. Hence, we used AlchemyAPI for the extraction of semantic concepts in our paper.

Using AlchemyAPI, we extracted a total of 15,139 entities from the training set, which are mapped to 30 distinct concepts and extracted 329 entities from the test set, which are mapped to 18 distinct concepts. Table 1 shows the top five extracted concepts from the training data with the number of entities associated with them.

Concept	Number of Entities
Person	4954
Company	2815
City	1575
Country	961
Organisation	614

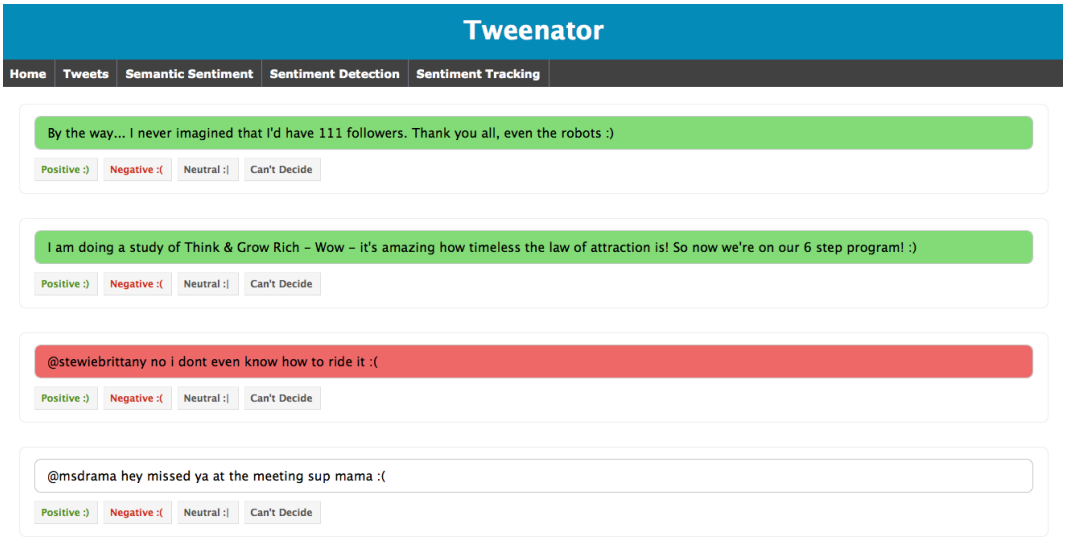
Table 1: Top 5 concepts with the number of their associated entities.

4.2 Incorporating Semantic Concepts into NB Training

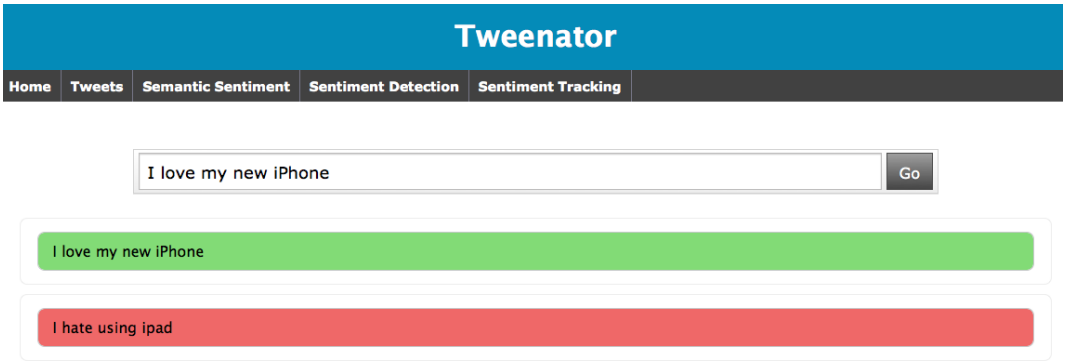
The extracted semantic concepts can be incorporated into sentiment classifier training in a naive way where entities are simply replaced by their mapped semantic concepts in the tweets data. For example, all the entities such as "*iPhone*", "*iPad*", and "*iPod*" are replaced by the semantic concept "*Product/Apple*". A more principled way to incorporate semantic concepts is through interpolation. Here, we propose interpolating the unigram language model with the generative model of words given semantic concepts in NB training.

In NB, the assignment of a sentiment class c to a given tweet $\mathbf{w}$ can

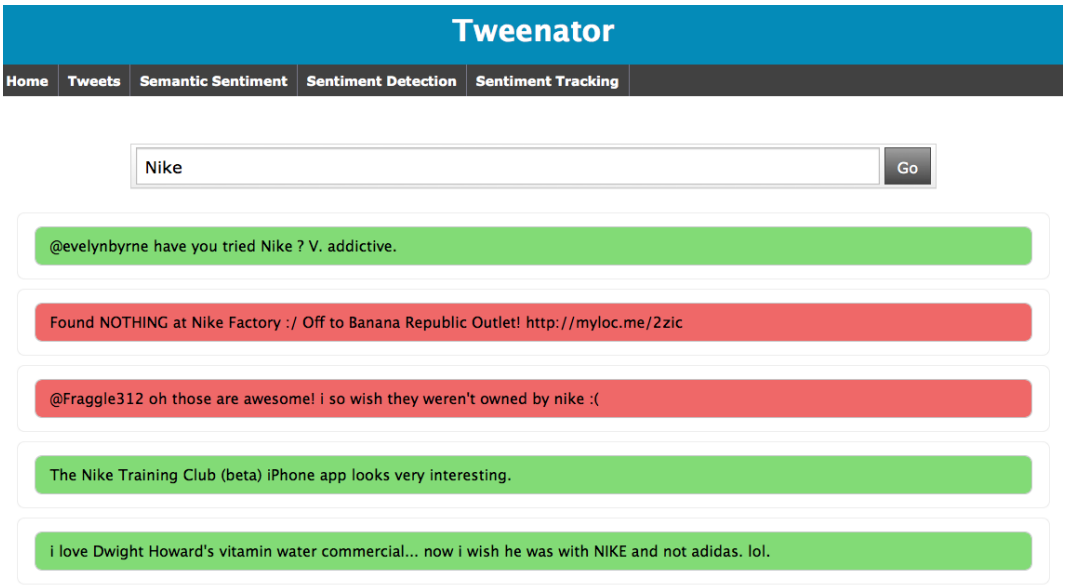
[2] http://www.cs.pitt.edu/mpqa/
[3] http://twittersentiment.appspot.com/
[4] http://atkmi.com/tweenator/
[5] http://www.zemanta.com/
[6] http://www.opencalais.com/
[7] http://www.alchemyapi.com/



(a) Sentiment Annotation Interface.



(b) Free-Form Sentiment Detector Interface.



(c) Opinionated Tweet Message Retrieval Interface.

Figure 3: Tweenator: Web based Sentiment Annotation Tool for Twitter

be computed as:

$$\hat{c} = \arg\max_{c \in \mathcal{C}} P(c|\mathbf{w}) = \arg\max_{c \in \mathcal{C}} P(c) \prod_{1 \le i \le N_{\mathbf{w}}} P(w_i|c), \quad (1)$$

where $N_{\mathbf{w}}$ is the total number of words in tweet $\mathbf{w}$, $P(c)$ is the prior probability of a tweet appearing in class c, $P(w_i|c)$ is the conditional probability of word w_i occurring in a tweet of class c.

In multinomial NB, $P(c)$ can be estimated by $P(c) = N_c/N$ Where N_c is the number of tweets in class c and N is the total number of tweets. $P(w_i|c)$ can be estimated using maximum likelihood with Laplace smoothing:

$$P(w|c) = \frac{N(w,c)+1}{\sum_{w' \in V} N(w'|c) + |V|} \quad (2)$$

Where $N(w,c)$ is the occurrence frequency of word w in all training tweets of class c and $|V|$ is the number of words in the vocabulary. Although using Laplace smoothing helps to prevent zero probabilities of the "unseen" words, it assigns equal prior probabilities to all of these words.

We propose a new smoothing method where we interpolate the unigram language model in NB with the generative model of words given semantic concepts. Thus, the new class model with semantic smoothing has the following formula:

$$P_s(w|c) = (1-\alpha)P_u(w|c) + \alpha \sum_j P(w|s_j)P(s_j|c) \quad (3)$$

Where $P_s(w|c)$ is the unigram class model with semantic smoothing, $P_u(w|c)$ is the unigram class model with maximum likelihood estimate, s_j is the j-th concept of the word w, $P(s_j|c)$ is the distribution of semantic concepts in training data of a given class and it can computed via the maximum likelihood estimation. $P(w|s_j)$ is the distribution of words in the training data given a concept and it can be also computed via the maximum likelihood estimation. Finally, the coefficient α is used to control the influence of the semantic mapping in the new class model. By setting α to 0 the class model becomes a unigram language model without any semantic interpolation. On the other hand, setting α to 1 reduces the class model to a semantic mapping model. In this work, α was empirically set to 0.5.

5. SENTIMENT-TOPIC FEATURES

The joint sentiment-topic (JST) model [11] is a four-layer generative model which allows the detection of both sentiment and topic simultaneously from text. The generative procedure under JST boils down to three stages. First, one chooses a sentiment label l from the per-document sentiment distribution π_d. Following that, one chooses a topic z from the topic distribution $\theta_{d,l}$, where $\theta_{d,l}$ is conditioned on the sampled sentiment label l. Finally, one draws a word w_i from the per-corpus word distribution $\phi_{l,z}$ conditioned on both topic z and sentiment label l. The JST model does not require labelled documents for training. The only supervision is word prior polarity information which can be obtained from publicly available sentiment lexicons such as the MPQA subjectivity lexicon.

We train JST on the training set with tweet sentiment labels being discarded. The resulting model assigns each word in tweets with a sentiment label and a topic label. Hence, JST essentially clusters different words sharing similar sentiment and topic. We list some of the topic words extracted by JST in Table 2. Words in each cell are grouped under one topic and the upper half of the table shows topic words bearing positive sentiment while the lower half shows topic words bearing negative polarity. It can be observed that words groups under different sentiment and topic are quite informative and coherent. For example, Topic 3 under positive sentiment is related to a good music album, while Topic 1 under negative sentiment is about a complaint of feeling sick possibly due to cold and headache.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Positive	dream	bought	song	eat	movi
	sweet	short	listen	food	show
	train	hair	love	coffe	award
	angel	love	music	dinner	live
	love	wear	play	drink	night
	goodnight	shirt	album	yummi	mtv
	free	dress	band	chicken	concert
	club	photo	guitar	tea	vote
Negative	feel	miss	rain	exam	job
	today	sad	bike	school	hard
	hate	cry	car	week	find
	sick	girl	stop	tomorrow	hate
	cold	gonna	ride	luck	interview
	suck	talk	hit	suck	lost
	weather	bore	drive	final	kick
	headache	feel	run	studi	problem

Table 2: Extracted polarity words by JST.

Inspired by the above observations, grouping words under the same topic and bearing similar sentiment could potentially reduce data sparseness in twitter sentiment classification. Hence, we extract sentiment-topics from tweets data and augment them as additional features into the original feature space for NB training. Algorithm 1 shows how to perform NB training with sentiment-topics extracted from JST. The training set consists of labeled tweets, $\mathcal{D}^{train} = \{(\mathbf{w}_n; c_n) \in \mathcal{W} \times \mathcal{C} : 1 \le n \le N^{train}\}$, where $\mathcal{W}$ is the input space and $\mathcal{C}$ is a finite set of class labels. The test set contains tweets without labels, $\mathcal{D}^{test} = \{\mathbf{w}_n^t \in \mathcal{W} : 1 \le n \le N^{test}\}$. A JST model is first learned from the training set and then infer sentiment-topic for each tweet in the test set. The original tweets are augmented with those sentiment-topics as shown in Step 4 of Algorithm 1, where $l_i\_z_i$ denotes a combination of sentiment label l_i and topic z_i for word w_i. Finally, an optional feature selection step can be performed according to the information gain criteria and a classifier is then trained from the training set with the new feature representation.

6. EXPERIMENTAL RESULTS

In this section, we present the results obtained on the twitter sentiment data using both semantic features and sentiment-topic features and compare with the existing approaches.

6.1 Pre-processing

The raw tweets data are very noisy. There are a large number of irregular words and non-English characters. Tweets data have some unique characteristics which can be used to reduce the feature space through the following pre-processing:

Algorithm 1 NB training with sentiment-topics extracted from JST.

Input: The training set $\mathcal{D}^{train}$ and test set $\mathcal{D}^{test}$
Output: NB sentiment classifier

```
Train a JST model on D^train with the document labels discarded
Infer sentiment-topic from D^test
for each tweet w_n = (w_1, w_2, ..., w_m) ∈ {D^train, D^test} do
    Augment tweet with sentiment-topics generated from JST,
    w'_n = (w_1, w_2, ..., w_m, l_1_z_1, l_2_z_2, ..., l_m_z_m)
end for
Create a new training set D^train' = {(w'_n; c_n) : 1 ≤ n ≤ N^train}
Create a new test set D^test' = {w'_n : 1 ≤ n ≤ N^test}
Perform feature selection using IG on D^train'
Return NB trained on D^train'
```

Pre-processing	Vocabulary Size	% of Reduction
None	95,130	0%
Username	70,804	25.58%
Hashtag	94,200	0.8%
URLS	92,363	2.91%
Repeated Letters	91,824	3.48%
Digits	92,785	2.47%
Symbols	37,054	29.47%
All	37,054	61.05%

Table 3: The effect of pre-processing.

- All Twitter usernames, which start with @ symbol, are replaced with the term "USER".
- All URL links in the corpus are replaced with the term "URL".
- Reduce the number of letters that are repeated more than twice in all words. For example the word "looooooveeee" becomes "loovee" after reduction.
- Remove all Twitter hashtags which start with the # symbol, all single characters and digits, and non-alphanumeric characters.

Table 3 shows the effect of pre-processing on reducing features from the original feature space. After all the pre-processing, the vocabulary size is reduced by 62%.

6.2 Semantic Features

We have tested both the NB classifier from WEKA[8] and the maximum entropy (MaxEnt) model from MALLET[9]. Our results show that NB consistently outperforms MaxEnt. Hence, we use NB as our baseline model. Table 4 shows that with NB trained from unigrams only, the sentiment classification accuracy of 80.7% was obtained.

We extracted semantic concepts from tweets data using Alchemy API and then incorporated them into NB training by the following two simple ways. One is to replace all entities in the tweets corpus with their corresponding semantic concepts (*semantic replacement*). Another is to augment the original feature space with semantic concepts as additional features for NB training (*semantic augmentation*). With *semantic replacement*, the feature space shrunk substantially by nearly 20%. However, sentiment classification accuracy drops by 4% compared to the baseline as shown in Table 4. The performance degradation can be explained as the mere use of semantic concepts replacement which leads to information loss and subsequently hurts NB performance. Augmenting the original feature space with semantic concepts performs slightly better than *sentiment replacement*, though it still performs worse than the baseline.

With *Semantic interpolation*, semantic concepts were incorporated into NB training taking into account the generative probability of words given concepts. The method improves upon the baseline model and gives a sentiment classification accuracy of 84%.

Method	Accuracy
Unigrams	80.7%
Semantic replacement	76.3%
Semantic augmentation	77.6%
Semantic interpolation	**84.0%**
Sentiment-topic features	82.3%

Table 4: Sentiment classification results on the 1000-tweet test set.

6.3 Sentiment-Topic Features

To run JST on the tweets data, the only parameter we need to set is the number of topics T. It is worth noting that the total number of the sentiment-topics that will be extracted is $3 \times T$. For example, when T is set to 50, there are 50 topics under each of positive, negative and neutral sentiment labels. Hence the total number of sentiment-topic features is 150. We augment the original bag-of-words representation of the tweet messages by the extracted sentiment-topics. Figure 4 shows the classification accuracy of NB trained from the augmented features by varying the number of topics from 1 to 65. The initial sentiment classification accuracy is 81.1% with topic number 1. Increasing the number of topics leads to the increase of classification accuracy with the peak value of 82.3% being reached at topic number 50. Further increasing topic numbers degrades the classifier performance.

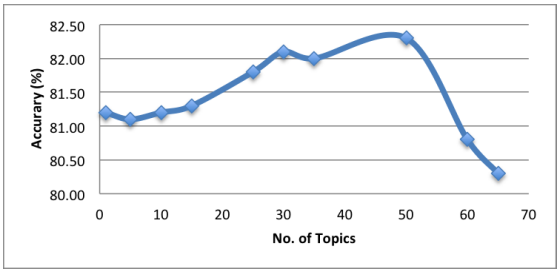


Figure 4: Classification accuracy vs. number of topics.

6.4 Comparison with Existing Approaches

In order to compare our proposed methods with the existing approaches, we also conducted experiments on the original Stanford Twitter Sentiment test set which consists of 177 negative and 182 positive tweets. The results are shown in Table 5. The sentiment classification accuracy of 83% reported in [5] was obtained using MaxEnt trained on a combination of unigrams and bigrams. It should be noted that while Go et al. used 1.6 million tweets for training, we only used a subset of 60,000 tweets as our training set.

[8] http://www.cs.waikato.ac.nz/ml/weka/
[9] http://mallet.cs.umass.edu/

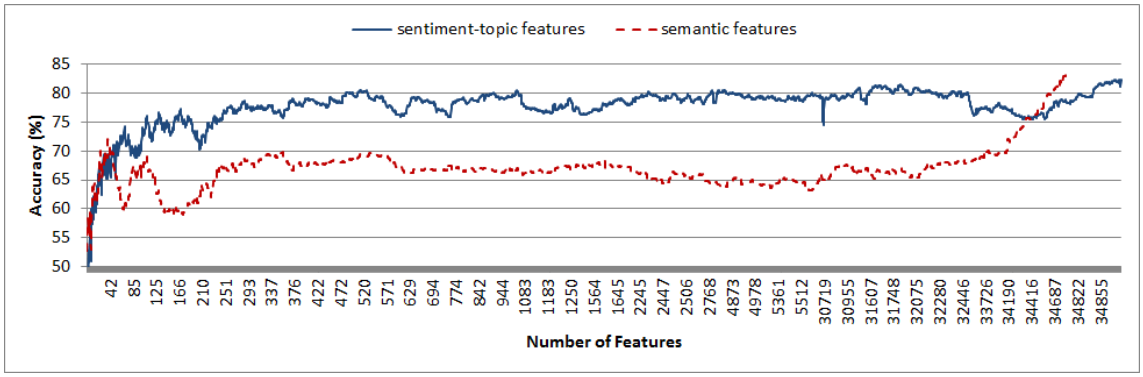


Figure 5: Classification accuracy vs. number of features selected by information gain.

Speriosu et al. [18] tested on a subset of the Stanford Twitter Sentiment test set with 75 negative and 108 positive tweets. They reported the best accuracy of 84.7% using label propagation on a rather complicated graph that has users, tweets, word unigrams, word bigrams, hashtags, and emoticons as its nodes.

It can be seen from Table 5 that *sentiment replacement* performs worse than the baseline. *Sentiment augmentation* does not result in the significant decrease of the classification accuracy, though it does not lead to the improved performance either. Our *semantic interpolation* method rivals the best result reported on the Stanford Twitter Sentiment test set. Using the sentiment-topic features, we achieved 86.3% sentiment classification accuracy, which outperforms the existing approaches.

Method	Accuracy
Unigrams	81.0%
Semantic replacement	77.3%
Semantic augmentation	80.45%
Semantic interpolation	84.1%
Sentiment-topic features	**86.3%**
(Go et al., 2009)	83%
(Speriosu et al., 2011)	84.7%

Table 5: Sentiment classification results on the original Stanford Twitter Sentiment test set.

6.5 Discussion

We have explored incorporating semantic features and sentiment-topic features for twitter sentiment classification. While simple *semantic replacement* or *augmentation* does not lead to the improvement of sentiment classification performance, *sentiment interpolation* improves upon the baseline NB model trained on unigrams only by 3%. Augmenting feature space with sentiment-topics generated from JST also results in the increase of sentiment classification accuracy compared to the baseline. On the original Stanford Twitter Sentiment test set, NB classifiers learned from sentiment-topic features outperform the existing approaches.

We have a somewhat contradictory observation here. Using sentiment-topic features performs worse than using semantic features on the test set comprising of 1000 tweets. But the reverse is observed on the original Stanford Twitter Sentiment test set with 359 tweets. We therefore conducted further experiments to compare these two approaches.

We performed feature selection using information gain (IG) on the training set. We calculated the IG value for each feature and sorted them in descending order based on IG. Using each distinct IG value as a threshold, we ended up with different sets of features to train a classifier. Figure 5 shows the sentiment classification accuracy on the 1000-tweet test set versus different number of features. It can be observed that there is an abrupt change in x-axis from around 5600 features jumping to over 30,000 features. Using sentiment-topic features consistently performs better than using semantic features. With as few as 500 features, augmenting the original feature space with sentiment-topics already achieves 80.2% accuracy. Although with all the features included, NB trained with semantic features performs better than that with sentiment-topic features, we can still draw a conclusion that sentiment-topic features should be preferred over semantic features for the sentiment classification task since it gives much better results with far less features.

7. CONCLUSIONS AND FUTURE WORK

Twitter is an open social environment where users can tweet about different topics within the 140-character limit. This poses a significant challenge to Twitter sentiment analysis since tweets data are often noisy and contain a large number of irregular words and non-English symbols and characters. Pre-processing by filtering some of the non-standard English words leads to a significant reduction of the original feature space by nearly 61.0% on the Twitter sentiment data. Nevertheless, the pre-processed tweets data still contain a large number of rare words.

In this paper, we have proposed two sets of features to alleviate the data sparsity problem in Twitter sentiment classification, semantic features and sentiment-topic features. Our experimental results on the Twitter sentiment data show that while both methods improve upon the baseline Naïve Bayes model trained from unigram features only, using sentiment-topic features gives much better results than using semantic features with less features.

Compared to the existing approaches to twitter sentiment analysis which either rely on sophisticated feature engineering or complicated learning procedure, our approaches are much more simple and straightforward and yet attain comparable performance.

There are a few possible directions we would like to explore as future work. First, in the semantic method all entities where simply replaced by the associated semantic concepts. It is worth to perform a selective statistical replacement, which is determined based on the contribution of each concept towards making a better classification

decision. Second, sentiment-topics generated by JST model were simply augmented into the original feature space of tweets data. It could lead to better performance by attaching a weight to each extracted sentiment-topic feature in order to control the impact of the newly added features. Finally, the performance of the NB classifiers learned from semantic features depends on the quality of the entity extraction process and entity-concept mapping method. It is worth to investigate a filtering method which can automatically filter out low-confidence semantic concepts.

Acknowledgement This work is partially funded by the EU project ROBUST (grant number 257859).

8. REFERENCES

[1] AGARWAL, A., XIE, B., VOVSHA, I., RAMBOW, O., AND PASSONNEAU, R. Sentiment analysis of twitter data. In *Proceedings of the ACL 2011 Workshop on Languages in Social Media* (2011), pp. 30–38.

[2] BARBOSA, L., AND FENG, J. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of COLING* (2010), pp. 36–44.

[3] BHUIYAN, S. Social media and its effectiveness in the political reform movement in egypt. *Middle East Media Educator 1*, 1 (2011), 14–20.

[4] BOIY, E., HENS, P., DESCHACHT, K., AND MOENS, M. Automatic sentiment analysis in on-line text. In *Proceedings of the 11th International Conference on Electronic Publishing* (2007), pp. 349–360.

[5] GO, A., BHAYANI, R., AND HUANG, L. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* (2009).

[6] HATZIVASSILOGLOU, V., AND WIEBE, J. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1* (2000), Association for Computational Linguistics, pp. 299–305.

[7] HE, Y., AND SAIF, H. Quantising Opinons for Political Tweets Analysis. In *Proceeding of the The eighth international conference on Language Resources and Evaluation (LREC) - In Submission* (2012).

[8] HU, M., AND LIU, B. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004), ACM, pp. 168–177.

[9] HUSSAIN, M., AND HOWARD, P. the role of digital media. *Journal of Democracy 22*, 3 (2011), 35–48.

[10] KOULOUMPIS, E., WILSON, T., AND MOORE, J. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the ICWSM* (2011).

[11] LIN, C., AND HE, Y. Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM conference on Information and knowledge management* (2009), ACM, pp. 375–384.

[12] NARAYANAN, R., LIU, B., AND CHOUDHARY, A. Sentiment Analysis of Conditional Sentences. In *EMNLP* (2009), pp. 180–189.

[13] PAK, A., AND PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of LREC 2010* (2010).

[14] PANG, B., AND LEE, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* (2004), Association for Computational Linguistics, p. 271.

[15] PANG, B., LEE, L., AND VAITHYANATHAN, S. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (2002), Association for Computational Linguistics, pp. 79–86.

[16] READ, J., AND CARROLL, J. Weakly supervised techniques for domain-independent sentiment classification. In *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion* (2009), pp. 45–52.

[17] SAIF, H., HE, Y., AND ALANI, H. Semantic Smoothing for Twitter Sentiment Analysis. In *Proceeding of the 10th International Semantic Web Conference (ISWC)* (2011).

[18] SPERIOSU, M., SUDAN, N., UPADHYAY, S., AND BALDRIDGE, J. Twitter polarity classification with label propagation over lexical links and the follower graph. *Proceedings of the EMNLP First workshop on Unsupervised Learning in NLP* (2011), 53–63.

[19] TABOADA, M., AND GRIEVE, J. Analyzing appraisal automatically. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Report SS-04-07)* (2004), pp. 158–161.

[20] TURNEY, P. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)* (2002).

[21] WARD, J., AND OSTROM, A. The internet as information minefield:: An analysis of the source and content of brand information yielded by net searches. *Journal of Business research 56*, 11 (2003), 907–914.

[22] YOON, E., GUFFEY, H., AND KIJEWSKI, V. The effects of information and company reputation on intentions to buy a business service. *Journal of Business Research 27*, 3 (1993), 215–228.

[23] ZHAO, J., LIU, K., AND WANG, G. Adding redundant features for CRFs-based sentence sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2008), pp. 117–126.