STAIRS 2016
D. Pearce and H.S. Pinto (Eds.)

doi:10.3233/978-1-61499-682-8-155

Sentiment Classification of Social Media Content with Features Generated Using Topic Models

Stuart J. BLAIR [a], Yaxin BI [a] and Maurice D. MULVENNA [a]
[a] *School of Computing and Mathematics, Ulster University, Newtownabbey, BT37 0QB, United Kingdom, emails: Blair-S4@email.ulster.ac.uk*

Abstract. This paper presents a method for using topic distributions generated from topic models as features for performing sentiment analysis on documents. This will be tested in the social media domain, specifically Twitter. The proposed approach allows for the mapping from word space to topic space which allows for less features to be needed and also reduces computational complexity. Multiple machine learning algorithms will be used to test the topic model generated features and a number of different versions of test corpus will be used, including unigrams, bigrams, part-of-speech tagging and adjectives only. The method proposed will also be compared to other notable topic-sentiment methods such as the aspect-sentiment unification model and the joint sentiment/topic model. The results show that using topic distributions can improve the accuracy of classification algorithms, however, the performance can be dependent on the algorithm used and the initial features used. Additionally, we show that using only topics as features outperforms the hybrid topic-sentiment models.

Keywords. sentiment classification, topic models, social media, feature generation

1. Introduction

In recent years sentiment analysis has gained much attention in the natural language processing community due to the variety and usefulness of its applications. For instance, companies can analyse reviews to see how customers find their products, politicians can monitor the public's opinion of them, and news companies can monitor how the public react to various events. Sentiment classification itself can take a variety of forms, from a simple polarity check of positive, negative or neutral/objective [1]; to a more complex analysis of specific emotions [2].

Perhaps one of the most interesting developments of the Internet has been the emergence of social networks, specifically microblogs. Many Internet users have been abandoning traditional methods of online communication such as blogs and newsgroups, in favour of social networks that enable microblogging, for instance, Twitter. These microblog platforms have enabled millions of users to quickly and concisely express opinions about anything from products to politics. For this reason these microblogs have become an invaluable source of information for many companies and institutions to gauge consumer opinion and help shape future product development or marketing campaigns.

In this paper we propose a new method for generating features to be used by sentiment analysis machine learning algorithms and evaluate how the new features perform by themselves and also how they perform when used in conjunction with the standard word features. Specifically, the contribution focuses on mapping features to a topic level instead of a word level. In this way features would be numeric, showing a topics proportion in a document; as opposed to being nominal and merely showing the binary presence of a word in a document. This has two distinct advantages: firstly, it reduces the number of features needed and therefore reduces computational complexity; and secondly, it allows for the concept of topics to be used as features. Meaning a group of words can be a feature rather than just one word.

The feature generation method proposed in this paper involves using topic models, specifically latent Dirichlet allocation (LDA) [3] to generate the features. Topic models discover the underlying topics in a corpus of text and outputs them as lists of words. Using these topics, a distribution of the topics within a document can be inferred. It is this document-topic distribution that gives the features to train the machine learning algorithms. For instance, if we had three topics:

- $t_1 = \{happy, excited, ..., joy\}$
- $t_2 = \{sad, upset, ..., worry\}$
- $t_3 = \{car, road, ..., grass\}$

Then a document might have a topic distribution of $\theta_d = \{0.8, 0.05, 0.15\}$ which would indicate that it has a higher presence of topic t_1 and therefore is positive due to the words in the topic being positive. Of course, in reality the topics would not be as distinct as those presented above, however, in this paper we investigate if given a training set of documents with their topic distributions, can sentiment analysis be performed accurately.

This will be tested in a social network domain, specifically Tweets; these short documents use slang and abbreviated English with little grammatical structure, resulting in a noisy text.

Machine learning algorithms will be trained using only the topic distributions, then they will be trained using a conjunction of word features and topic distributions. A variation of word features will be used for the social media domain: unigrams, bigrams, part-of-speech (POS) tags and adjectives.

2. Related Work

The standard process to perform sentiment analysis on a document collection is to generate a feature set from the vocabulary of the document collection and then use a lexical approach [4] or use machine learning to predict the sentiment of instances [5]. However, there is not as much research in the area of sentiment analysis of short text such as microblogs.

Research has been done in general social media sentiment classification based on a dataset generated by searching for Tweets containing certain emoticons [6]. There has also been research into social media sentiment analysis at a user level [7] and exploiting social relations to aid sentiment analysis [8]. A keyword approach to sentiment analysis on Twitter has also been investigated [9].

Topic models have also been used on social media. It has been shown that modelling Tweets containing certain adjectives can help to predict social sentiment [10]. Research has also investigated how to increase the performance of topic models on short text such as Tweets, and also provides insight into how the author-topic model can aid topic model use of social media [11].

Although this paper does not directly use topic models for sentiment analysis, there has been research into this approach also. Perhaps the most influential paper in this area is the Aspect Sentiment Unification Model (ASUM) [12]. This paper takes an unsupervised approach to the problem of jointly modelling sentiment and topics; the only input it requires apart from the initial data is a small list of general affective seed words known as Paradigm+ [13]. The disadvantage of this is that it will not work well in all domains because of the generality of the seed words, for example, later in this report will be a discussion of how this model performs on Twitter social media content. This implementation also falls under the assumption that one sentence is about one topic and therefore has only one sentiment; therefore it works only at a sentence level. The authors argue that this works well, however it does not always hold true, especially in cases such as social media. In order to incorporate the prior sentiment information the model utilises an asymmetric β prior, this allows the positive words to have low negative probability but high positive probability and vice versa. The paper first compares the topics found by their sentence-LDA against traditional LDA. Unfortunately, there is no qualitative study or questionnaire performed to check a group of humans' opinion of the topics' coherence found by both models. However, upon examining the topics provided within the paper, they seem to be very similar. The sentiment topics discovered by the model however, seem to be of decent quality, however, the topics discovered are asymmetric, for example not every positive topic has a corresponding negative topic. The authors report accuracy between 70% and 80% (depending on number of topics) when using the Paradigm seed word list, and accuracy of 85% when using the Paradigm+ seed word list irrespective of number of topics which seems unusual as adjusting the number of topics normally has some effect on accuracy due to over/underfitting. The accuracy tests were performed on Amazon product reviews and Yelp restaurant reviews.

ASUM can be seen as a derivation of the Joint Sentiment/Topic Model (JST) [14] and both are in some ways quite similar. Similarly to ASUM, JST utilises a Paradigm seed word list in order to incorporate the sentiment information into the posterior distribution. However, JST also combines this with the multi-perspective question answering (MPQA) subjectivity lexicon using the mutual information between the seed words and the words in the lexicon, this is then filtered based on term frequency within the models corpus. Despite the resulting substantially large lexicon, it is still general to some degree and has no domain specificities. JST creates topics on a document level, unlike ASUM, which assigns topics on a sentence level. Unlike ASUM, JST also accounts for a neutral label, this can be useful in some sentiment classification problems; however, during the authors testing phase, they discarded the neutral label as they view the test dataset (movie reviews) as a binary sentiment classification test. This model proved fairly successful on classifying movie review sentiment with a best score of 84.6%, in contrast the best support vector machine they tested against scored 90.2%. One thing to note about JST is that its accuracy is based on the average of its positive classification accuracy and negative classification accuracy; in all cases the classification accuracy of negative reviews was

significantly lower than the positive accuracy. This may be due to some movie genres (such as horror) having negative words in them but being described positively.

Another method for jointly modelling topics and sentiment is to utilise a twofold approach [15]. Similar to JST and ASUM, this method uses a modified version of Gibbs sampling in order to add seed words to the posterior distribution. However, this model differs in that rather than a single topic model being used, a separate model is used for both topic and sentiment, and hence the twofold approach. This model was later extended to allow multiple aspects in one sentence [16].

Another important paper in the area of jointly modelling topics and sentiment is a MaxEnt-LDA approach [17]. Like the other models described that are quite weakly supervised, this model requires a relatively low degree of supervision as a simple seed word list is only. Again, this model works at a sentence level. It utilises an LDA model to identify the topic of a sentence and then support vector regression (SVR) is used to get the sentiment of each sentence. The SVR was trained on supersets of hotel and restaurant reviews; the superset was formed by combining all reviews for individual hotels or restaurants, and using the star rating to label the supersets. Relatively high accuracy was achieved by the model coming in at 80.3%, compared to 83% for a standard SVM using 5-fold cross validation.

3. Topic Models

Topic models originated with latent semantic analysis (LSA), however, when applied to an information retrieval task it is commonly referred to as latent semantic indexing [18]. LSA utilises a document-term matrix and singular value decomposition to find similar documents, making the assumption that words which frequently appear together are related. Two notable disadvantages of LSA are that the model is based around the bag-of-words method and that it struggles with polysemy. This means that word order in documents is abandoned and that it cannot distinguish between the different meanings of a single word. For example, *crane* can refer to both a bird as well as a piece of construction machinery.

The advent of latent Dirichlet allocation (LDA) has helped to eliminate the polysemy difficulties by introducing a probabilistic element to the model but it has continued to struggle with the bag-of-words assumption, abandoning all sentence structure when creating the model [3].

In this paper, the topic model that will be utilised is LDA; it is a generative probabilistic model that finds latent topics in a collection of documents by learning the relationship between words (w_j), documents (D_j) and topics (z_j). The data used by an LDA model is in bag-of-words form, word counts are preserved but the ordering of the words is lost.

The generative process for document D_i assumes the following:

- There is a fixed number of topics K.
- Each topic z has a multinomial distribution over vocabulary ϕ_z drawn from Dirichlet prior $Dir(\beta)$.
- $i \in \{1, \ldots, M\}$ where M is the number of documents in the corpus.
- $Dir(\alpha)$ is the document-topic Dirichlet distribution.

This is the generative process for document D_i:

1. Choose $\theta_i \sim Dir(\alpha)$.
2. For word $w_j \in D_i$:
   (a) Draw a topic $z_j \sim \theta_i$.
   (b) Draw a word $w_j \sim \phi_{z_j}$.

Specifically in this paper the focus will be on the topic distributions in documents ($\theta_{d,k=1...K}$), where d is a document and k is a topic from topic list K. Each topic generated by the model will be a feature used by the sentiment classifiers in this paper. The numeric values associated with the topics are real numbers in the range $0 < n < 1$. While most classification approaches simply use word frequency or word presence as a feature, using the topic distribution will allow for the weight of a group of words (a topic) to be a feature. Ideally, each topic will point towards a certain polarity, i.e., positive or negative, and that if a certain topic has a higher weight it will indicate that polarity. For example, if we have three topics generated from a topic model and their inferred distribution in a document are 0.65, 0.15 and 0.2, respectively; then the first topic's polarity will likely be the documents polarity.

4. Classification Methods

This work assumes the classification problem to have two distinct classes, positive and negative. The neutral/objective class was not considered in this work. Three popular classification algorithms are used in this work, namely the Naïve Bayes classifier (NB), support vector machines (SVM) and the maximum entropy classifier (MaxEnt).

To represent the documents the bag-of-words model was used, however, rather than using frequency counts of words this paper simply uses a presence check. Therefore, we can define a document as a vector. If we have a set of features $\{f_1, ..., f_n\}$, where n is the number of features and a feature is a unigram or bigram, for example, "sad" or "really happy". A document can be seen as a vector $\vec{d} := (w_1(d), w_2(d), ..., w_n(d))$ where $w_i(d)$ has a value $[0, 1]$ which indicates if feature f_i is present in document d.

4.1. Naïve Bayes

Although naïve Bayes classification is fairly simple, it has still been shown to perform well [19]; this is particularly true if the features are highly dependent [20].

The NB classifier uses Bayes theorem as shown in Eq. (1), where d is a document and c is a class.

$$p(c|d) = \frac{p(d|c)p(c)}{p(d)} \tag{1}$$

In order to assess $p(c|d)$ for a given document and class, Eq. (2) is used. Where $p(c_i)$ is the probability of class i and $p(f_j|c_i)$ is the probability of word f_j occurring in class c_i. The product of these probabilities is then taken and whichever class has the highest probability is the assigned class.

$$p(c|d) = \underset{c_i \in C}{argmax} \; p(c_i) \prod_j p(f_j|c_i) \tag{2}$$

4.2. Support Vector Machine

Support vector machines (SVMs) have been shown to be generally more accurate than the NB classifier at text classification, although this is not true for every problem [21].

SVMs are supervised models that can be seen as non-probabilistic binary linear classifiers. It should be noted that SVMs can also be non-linear by mapping their features into a higher dimension of feature space using the kernel trick, such as a radial basis function kernel or polynomial kernel.

The general idea for a binary classification problem is to find a hyperplane separating the data points for each class so that the margin between the set of points for each class is maximised. Using the kernel trick allows for a non-linear margin to be mapped into higher dimension feature space in which the classifier is a hyperplane.

In this paper, due to the large number of features, sequential minimal optimisation (SMO) with a polynomial kernel was used to train the SVM [22]. SMO is used to solve the SVM training quadratic programming problem faster by iteratively breaking it into smaller sub-problems and solving the smallest optimisation problem using Lagrange multipliers until convergence.

4.3. Maximum Entropy

A maximum entropy classifier (MaxEnt) is mathematically the same as logistic regression and has seen previous use in natural language applications [23]; and has been shown to occasionally be more successful than NB for classification problems [24].

The method MaxEnt uses for assessing a document's class is shown in Eq. (3), Where $Z(d)$ is a normalisation function and $F_{i,c}$ is the feature-class function. Finally, $\lambda_{i,c}$ is a parameter that defines a feature's weight where a high value infers that feature i is a compelling indicator of the document belonging to class c.

$$p(c|d) = \frac{1}{Z(d)} exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right) \tag{3}$$

5. Social Media Domain

For this set of experiments the Sentiment140[1] dataset was used. This is a collection of labelled Tweets with positive, negative and neutral labels. For this use, all neutral Tweets were removed. A subset of 1000 positive and 1000 negative Tweets was then created by random sampling from the Sentiment140 dataset. This same random subset was used for all experiments in this paper.

The following steps were performed to preprocess the Tweets. All Tweets were converted to lowercase, non-alphanumeric characters were removed and only words appearing five times or more were kept. Additionally, usernames (words beginning with the '@' symbol) were removed from the Tweets. An attempt to normalise hashtags was implemented; if PascalCase or CamelCase was used in the hashtag then it would be split at each capital letter into separate words. For example, '*#ThisIsAnExample*' would become '*this is an example*'. Stop word removal was also performed using the rainbow library.

[1] Available at: http://help.sentiment140.com/for-students/

Following the preprocessing of the dataset, it was further processed to create four distinct versions, they are as follows:

- Unigrams: Each word in the dataset will count as a feature.
- Bigrams: Each set of bigrams in the dataset will count as a feature.
- Part-of-Speech Tags: Each unigram in the dataset will be tagged with its part-of-speech tag.
- Adjectives: This version utilises the part-of-speech tagged dataset and removes all features that do not have adjective related tags, the tags used are as follows: adjectives (*JJ*), comparative adjectives (*JJR*), superlative adjectives (*JJS*), adverbs (*RB*), comparative adverbs (*RBR*) and superlative adverbs (*RBS*).

A further four versions of these datasets were then created by appending the topic distribution for each review. This resulted in a feature vector for each document $\vec{d} := (w_1(d), ..., w_n(d), t_1(d), ..., t_m(d))$ where $w_n(d)$ indicates the presence of a particular word w_n appearing in document d and t_m indicates the distribution of topic t_m in document d. Three topic models were created from the unigram dataset, each with a different number of topics (10, 50 and 100). The topic models were created using LDA, with 2000 iterations of Gibbs sampling, an alpha value of $50/T$ (where T is the number of topics) and a beta value of 0.001.

This process resulted in a total of 16 datasets which will be analysed using three machine learning algorithms, NB, SVM and MaxEnt. Each classifier will be evaluated using 10 fold cross validation. Additionally, datasets containing only the topic distribution and no word features will be tested with these same classifiers to see how the topic distributions perform by themselves.

5.1. Sentiment Classification using only Topic Features

The first experiment was to train the classifiers using only the topic distributions as features; to generate these topic distributions, multiple topic models (10, 50 and 100 topics) were generated for each dataset. The results for this can be seen in Table 1. As seen in

Table 1. Classifier accuracy for Tweets using only topics as features

(a) Unigrams

	10	50	100
NB	53.3	**56.7**	54.85
SVM	54.3	**57.9**	55.6
MaxEnt	54.95	**58.45**	56

(b) Bigrams

	10	50	100
NB	49.9	51.85	**54.05**
SVM	50.2	50.75	**55.75**
MaxEnt	49.05	51.2	**55.3**

(c) POS Tags

	10	50	100
NB	55.35	54.9	**58.05**
SVM	57.25	54	**58.25**
MaxEnt	56.95	54.05	**58.6**

(d) Adjectives

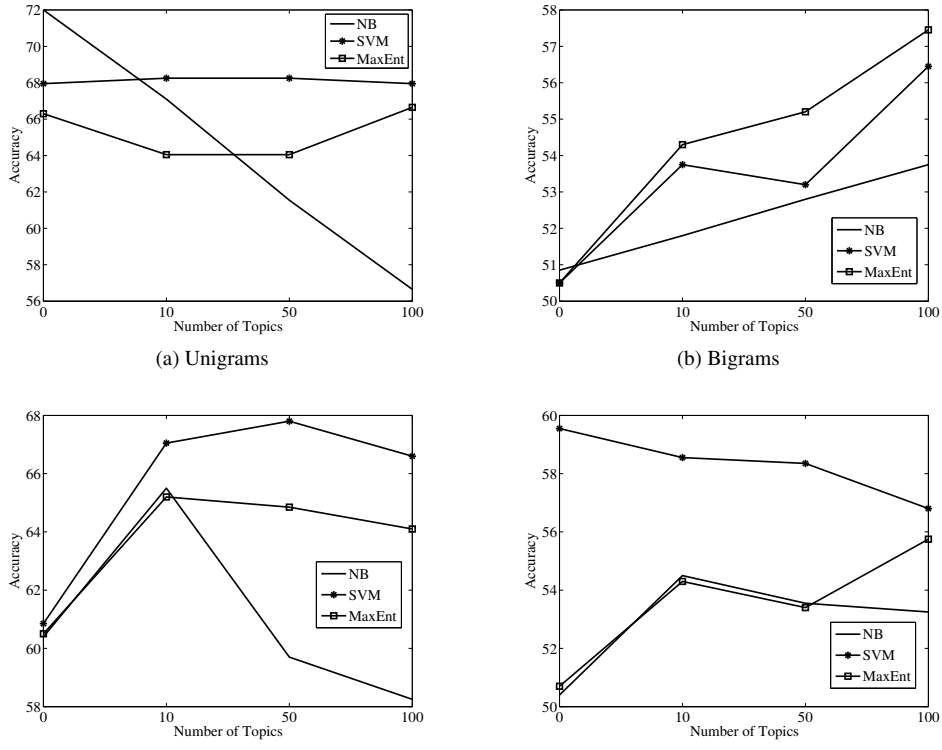
	10	50	100
NB	50.95	**53.5**	53.4
SVM	50.55	52.75	**54.1**
MaxEnt	51.15	52.65	**53**

Table 1a, the MaxEnt classifier performs the best and the optimal number of topics is 50. In Tables 1b and 1c the optimal number of topics is 100 topics, this could suggest overfitting of the topics to the dataset because the accuracy is increasing as more topics are added. In all cases, the support vector machine and maximum entropy classifiers perform with the best accuracy. This suggests that using the numeric distribution of topics as a feature is not a task well suited to probabilistic classifiers such as naïve Bayes.

5.2. Sentiment Classification using Word Features and Topic Features

This section will focus on the results of running the classification algorithms on the different social media datasets, the results can be seen in the charts in Figure 1.

The analysis will begin with unigrams, visible in Figure 1a. As can be seen, the NB classifier performs the best with 0 topics and gets substantially worse as more topics are added, this could be due to the independence assumption that the NB classifier uses. MaxEnt also performs similarly, getting a slight accuracy boost when 100 topics are used, but then falling when 10 or 50 topics are added, this could be a sign of overfitting. The SVM classifier has mixed results, seeing its accuracy increase slightly as topics are added, but starts to fall again when 100 topics are added.



(a) Unigrams (b) Bigrams

(c) POS Tags (d) Adjectives

Figure 1. Classifier accuracy for Tweets for each feature type with topics appended as additional features

Next, the bigram dataset's accuracies will be discussed, seen in Figure 1b. All the classifiers begin with an accuracy around 50% and see notable gains in accuracy when any amount of topics are added. Interestingly, the SVM classifier shows a drop in accuracy at 50 topics, which can be seen as somewhat encouraging as it does not continually increase in accuracy with more topics, as this could indicate overfitting of data.

The part-of-speech tagged dataset shown in Figure 1c has an interesting set of results. After adding 10 topics, all of the classifiers see a substantial increase in accuracy, however, past that point the accuracies begin to decrease; most notable of which is the NB classifier. The explanation for the NB classifier's sharp decrease could be due to the fact that it makes independence assumptions about the data it classifies and therefore sees all topics as independent of each other despite being linked through their proportions. The explanation for the SVM and MaxEnt classifier's drop in accuracy could be linked to the topics that were generated from the model; because these topics are modelled from a part-of-speech tagged corpus, they may be too specific for the classifier to use to make accurate predictions.

Finally, the results of the adjective only dataset will be examined, they can be seen in Figure 1d. Interestingly, these results are similar to the unigram dataset. Like the NB classifier in Figure 1a, this time the SVM performs best with no topics appended, before falling sharply in accuracy. However, unlike the unigram results, this time the MaxEnt and NB classifiers do improve noticeably when the topics are added before beginning to drop in accuracy. The reason for results in this experiment may be due to the fact that only adjectives were used. This resulted in a much smaller set of features and the corpus documents being rendered down to only a few words, losing all syntactic and grammatical structure.

5.3. Hybrid Topic-Sentiment Models

As mentioned previously, there are hybrid topic-sentiment models that jointly model sentiment and topics at the same time. In this section these models are compared to using topic features to train a series classification algorithms.

The two most notable hybrid topic-sentiment models are JST and ASUM, which are very similar except for the level at which they work, JST at the document level and ASUM at the sentence level. Both of these algorithms require a list of words used as prior knowledge for each sentiment to be discovered. In this experiment the paradigm+ word list will be used [13]. After the model has been created, an additional distribution π_i is created for each document modelled. This distribution π_i is the per document sentiment distribution and whichever sentiment has the highest value is the sentiment of the document.

As seen in Table 1a, the highest accuracy achieved using only topics as features was 58.45% using 50 topics and a MaxEnt classifier. Conversely, after running the dataset

Table 2. Confusion matrices for topic only classification

(a) MaxEnt

	Pos	Neg
Pos	585	415
Neg	416	584

(b) JST

	Pos	Neg
Pos	488	512
Neg	463	537

through the JST model an accuracy of 51.25% was achieved. This shows that using topics as features to train traditional classification algorithms can produce better results than hybrid topic-sentiment models.

The confusion matrices for both methods can be seen in Table 2. As is shown in Table 2a, the MaxEnt classifier achieves almost identical classification performance between true positives and true negatives. However, the JST method performs very poorly when classifying positive Tweets, as can be seen in Table 2b.

5.4. Results

In all of the experiments conducted, the addition of topics as features helps improve accuracy; most noticeably with part-of-speech tags and bigrams. Interestingly, the unigram and adjective experiments show that the addition of topics actually makes the classification performance worse as these experiments performed best when the topic features were absent. Worryingly, the bigram experiment continues to get more accurate as more topics are added, this could be a sign of overfitting.

Despite the fact that the highest performance was achieved by using unigram features with no additional topic features, there is still value in using topics as features for a sentiment analysis task. We have shown that training a classifier using the topic features results in a better accuracy than the hybrid topic-sentiment models such as ASUM and JST, by achieving a noticeably better result.

6. Discussion

The experiments conducted in this paper provide many interesting discussion points.

It was observed that there may be some problems with overfitting, especially in linear classifiers such as the SVM classifier. It was shown that they continually see their accuracy improving as more topics were added. This suggests that as topics become more document specific, it makes it easier for the classifier to make predictions.

In the social media experiment, it was shown that using the topic distributions as features can improve accuracy. The probabilistic NB classifier did not perform as well as the SVM or MaxEnt classifiers, it is somewhat disappointing that the best accuracy was achieved by using unigrams with no topic features; however, it was shown that using topic features is a more effective classification method than using hybrid topic-sentiment models.

An interesting observation from this set of experiments was how highly correlated the classifier accuracies are for unigrams, part-of-speech tags and adjectives. This could suggest that the classifiers are putting the most weight in features that are common be-

Table 3. Pearson's correlation between classifier accuracies

	Unigrams	Bigrams	POS Tags	Adjectives
Unigrams		0.28700817	0.94250963	0.799081453
Bigrams	0.28700817		0.242632163	0.230602273
POS Tags	0.94250963	0.242632163		0.791420345
Adjectives	0.799081453	0.230602273	0.791420345	

Table 4. Top four gain ratio social media topics

t_4	t_6	t_8	t_9
night	today	time	haha
great	work	lol	working
fun	happy	long	bad
twitter	one	friends	yeah
awesome	amazing	sun	ya
tomorrow	birthday	damn	sucks
hot	ve	phone	weekend
people	friend	gotta	lost
feeling	didn	sick	weather
sunday	baby	found	thing
0.139	0.182	-0.139	-0.322

tween them, in this case the only features they share are the topic distributions. Bigrams is not correlated with the other datasets, but this could be due to how different it is from the others; it focuses on pairs of words whereas the other datasets focus on individual words. This correlation can be seen in Table 3.

For most of the social media experiments, 10 topics produced the highest accuracy for the classifier, therefore, the information gain ratio was calculated for each topic in the distribution. The top four topics are in Table 4. It is clear that each topic has a distinct polarity with t_4 and t_6 being positive and the other two topics, t_8 and t_9, being negative as indicated by the average SentiWordNet score for each topic on the bottom row [4].

We can conclude that using topics as features can increase the classification accuracy in sentiment analysis, but it is largely classification algorithm dependent and the current results show that this method works better on short texts such as Tweets, further experiments on longer texts would be interesting. Additionally, further work to be conducted will need to pay more attention to how the topics are created to ensure they have more of a focus on sentiment and promote topics which are specific towards sentiment.

References

[1] Peter D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, 2002.

[2] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1556–1560, 2008.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[4] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, LREC '10, pages 2200–2204, 2010.

[5] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, EMNLP '02, pages 79–86, 2002.

[6] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, LREC '10, pages 1320–1326, 2010.

[7] Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1397–1405, 2011.

[8] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 537–546, 2013.

[9] M. Baumgarten, M. D. Mulvenna, N. Rooney, and J. Reid. Keyword-based sentiment mining using twitter. *International Journal of Ambient Computing and Intelligence*, 5(2):56–69, April 2013.

[10] Masahiro Ohmura, Koh Kakusho, and Takeshi Okadome. Social mood extraction from twitter posts with document topic model. In *2014 International Conference on Information Science and Applications*, pages 1–4, 2014.

[11] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, 2010.

[12] Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, 2011.

[13] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, October 2003.

[14] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 375–384, 2009.

[15] Nicola Burns, Yaxin Bi, Hui Wang, and Terry Anderson. A twofold-lda model for customer review analysis. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '11, pages 253–256, 2011.

[16] Nicola Burns, Yaxin Bi, Hui Wang, and Terry Anderson. Extended twofold-lda model for two aspects in one sentence. In *Proceedings of the 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, IPMU '12, pages 265–275, 2012.

[17] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 56–65, 2010.

[18] Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[19] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 4–15, 1998.

[20] Pedro Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2):103–130, 1997.

[21] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, 1998.

[22] John C. Platt. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. 1999.

[23] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, March 1996.

[24] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *Proceedings of the International Joint Conference on Artificial Intelligence 1999 Workshop on Machine Learning for Information Filtering*, IJCAI '99, pages 61–67, 1999.