Semantic Sentiment Analysis of Twitter

Hassan Saif, Yulan He and Harith Alani

Knowledge Media Institute, The Open University, United Kingdom
{h.saif,y.he,h.alani}@open.ac.uk

Abstract. Sentiment analysis over Twitter offer organisations a fast and effective way to monitor the publics' feelings towards their brand, business, directors, etc. A wide range of features and methods for training sentiment classifiers for Twitter datasets have been researched in recent years with varying results. In this paper, we introduce a novel approach of adding semantics as additional features into the training set for sentiment analysis. For each extracted entity (e.g. iPhone) from tweets, we add its semantic concept (e.g. "Apple product") as an additional feature, and measure the correlation of the representative concept with negative/positive sentiment. We apply this approach to predict sentiment for three different Twitter datasets. Our results show an average increase of F harmonic accuracy score for identifying both negative and positive sentiment of around 6.5% and 4.8% over the baselines of unigrams and part-of-speech features respectively. We also compare against an approach based on sentiment-bearing topic analysis, and find that semantic features produce better Recall and F score when classifying negative sentiment, and better Precision with lower Recall and F score in positive sentiment classification.

Keywords: Sentiment analysis, semantic concepts, feature interpolation.

1 Introduction

The emergence of social media has given web users a venue for expressing and sharing their thoughts and opinions on all kinds of topics and events. Twitter, with nearly 600 million users[1] and over 250 million messages per day,[2] has quickly become a gold mine for organisations to monitor their reputation and brands by extracting and analysing the sentiment of the Tweets posted by the public about them, their markets, and competitors.

Sentiment analysis over Twitter data and other similar microblogs faces several new challenges due to the typical short length and irregular structure of such content. Two main research directions can be identified in the literature of sentiment analysis on microblogs. First direction is concerned with finding new methods to run such analysis, such as performing sentiment label propagation on Twitter follower graphs [14], and employing social relations for user-level sentiment analysis [15, 5]. The second direction is focused on identifying new sets of features to add to the trained model for sentiment identification, such as microblogging features including hashtags, emoticons [2], the presence of intensifiers such as all-caps and character repetitions [6] etc., and sentiment-topic features [12].

[1] twopcharts.com/twitter500million.php

[2] www.geekosystem.com/twitter-250-million-tweets-per-day

The work in this paper falls into the second direction, by investigating a novel set of features derived from the semantic conceptual representation of the entities that appear in tweets. The *semantic features* consist of the semantic concepts (e.g. "person", "company", "city") that represent the entities (e.g. "Steve Jobs", "Vodafone", "London") extracted from tweets. The rational behind introducing these features is that certain entities and concepts tend to have a more consistent correlation with positive or negative sentiment. Knowing these correlations can help determining the sentiment of semantically relevant or similar entities, and thus increasing accuracy of sentiment analysis. To the best of our knowledge, using these semantic features in the model training for sentiment analysis has not been explored before. We evaluated three popular tools for entity extraction and concept identification; AlchemyAPI,[3] Zemanta,[4] and OpenCalais,[5] and used the one that performed best in terms of quantity and accuracy of the identified concepts.

While previous work on feature engineering for sentiment classification on tweets [1, 6] simply incorporate features through augmentation, our experimental results show that it is more effective to incorporate semantic features through interpolation. Hence we incorporate the semantic features into Naïve Bayes (NB) model training using an interpolation approach.

We experiment and evaluate our proposed approach with three datasets collected from Twitter; a general Stanford Twitter Sentiment (STS) dataset, a dataset on the Obama-McCain Debate (OMD), and one on Health Care Reform (HCR). Our results show that combining our semantic features with word unigrams outperforms the baseline model trained from unigrams only across all three datasets by an average accuracy of 6.47%. It also outperforms the accuracy of sentiment analysis using the common part-of-speech (POS) features often used in the literature [9, 1] by an average of 4.78%. Although these improvements may appear modest, they are very notable in comparison to the scale of improvements reported in similar literatures. Our results show that the advantage of using semantic features in microblog sentiment analysis over other techniques is mostly restricted to negative sentiment identification, in large topically-diverse datasets.

The main contributions of this paper can be summarised as follows:

- Introduce and implement a new set of semantic features for training a model for sentiment analysis of tweets
- Investigate three approaches for adding such features into the training model; by replacement, by argumentation, and by interpolation, and show the superiority of the latter approach.
- Test accuracy of sentiment identification when using semantic features with unigrams on three Twitter datasets, and produce an average harmonic mean (F score) accuracy of 75.95%, with 77.18% Precision and 75.33% Recall
- Demonstrate the value of *not* removing stowords in increasing sentiment identification accuracy

[3] www.alchemyapi.com
[4] www.zemanta.com
[5] www.opencalais.com

- Show an average of 6.47% increase in the F score against a baseline approach based on unigrams only
- Show an average of 4.78% increase in F score in comparison to using the common POS features alongside unigrams
- Compare results with sentiment-bearing topic features [12] and show that semantic features improve F by 1.22% when identifying negative sentiment, but worsens F by 2.21% when identifying positive sentiment.

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 three Twitter datasets used in our experiments. Section 4 presents our proposed approach of using semantic features for sentiment analysis, and describes three methods for incorporating these features into the sentiment classifier. In Section 5 we describe the baselines we use for evaluating and comparing our results. Experimental results are fully detailed and discussed in Section 6. Discussion and future work are covered in Section 7. Finally, we conclude our work in Section 8.

2 Related Work

Sentiment analysis of tweets data is considered as a much harder problem than that of conventional text such as review documents. This is partly due to the short length of tweets, the frequent use of informal and irregular words, and the rapid evolution of language in Twitter. A large amount of work has been conducted in Twitter sentiment analysis following the feature-based approaches. Go et al. [4] explored augmenting different n-gram features in conjunction with POS tags into the training of supervised classifiers including Naive Bayes (NB), Maximum Entropy (MaxEnt) and Support Vector Machines (SVMs). They found that MaxEnt trained from a combination of unigrams and bigrams outperforms other models trained from a combination of POS tags and unigrams by almost 3%. However, a contrary finding was reported in [9] that adding POS tag features into n-grams improves the sentiment classification accuracy on tweets.

Barbosa and Feng [2] argued that using n-grams on tweet data may hinder the classification performance because of the large number of infrequent words in Twitter. Instead, they proposed using microblogging features such as re-tweets, hashtags, replies, punctuations, and emoticons. They found that using these features to train the SVMs enhances the sentiment classification accuracy by 2.2% compared to SVMs trained from unigrams only. A similar finding was reported by Kouloumpis et al. [6]. They explored the microblogging features including emoticons, abbreviations and the presence of intensifiers such as all-caps and character repetitions for Twitter sentiment classification. Their results show that the best performance comes from using the n-grams together with the microblogging features and the lexicon features where words tagged with their prior polarity. However, including the POS features produced a drop in performance.

Agarwal et al. [1] also explored the POS features, the lexicon features and the microblogging features. Apart from simply combining various features, they also designed a tree representation of tweets to combine many categories of features in one succinct representation. A partial tree kernel [8] was used to calculate the similarity between two trees. They found that the most important features are those that combine prior polarity of words with their POS tags. All other features only play a marginal role. Furthermore,

they also showed that combining unigrams with the best set of features outperforms the tree kernel-based model and gives about 4% absolute gain over a unigram baseline.

Rather than directly incorporating the microblogging features into sentiment classifier training, Speriosu et al. [14] constructed a graph that has some of the microblogging features such as hashtags and emoticons together with users, tweets, word unigrams and bigrams 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 unigrams that they contain etc.). They then applied a label propagation method where sentiment labels were propagated from a small set 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 [4].

Existing work mainly concentrates on the use of three types of features; lexicon features, POS features, and microblogging features for sentiment analysis. Mixed findings have been reported. Some [9, 1] argued the importance of POS tags with or without word prior polarity involved, while others emphasised the use of microblogging features [2, 6]. In this paper, we propose a new type of features for sentiment analysis, called semantic features, where for each entity in a tweet (e.g. iPhone, iPad, MacBook), the abstract concept that represents it will be added as a new feature (e.g. Apple product). We compare the accuracy of sentiment analysis against other types of features; unigrams, POS features, and the sentiment-topic features. To the best of our knowledge, using such semantic features is novel in the context of sentiment analysis.

3 Datasets

For the work and experiments described in this paper, we used three different Twitter datasets as detailed below. The statistics of the datasets are shown in Table 1.

Dataset	Type	No. of Tweets	Positive	Negative
Stanford Twitter Sentiment Corpus (STS)	Train	60K	30K	30K
	Test	1,000	470	530
Health Care Reform (HCR)	Train	839	234	421
	Test	839	163	536
Obama-McCain Debate (OMD)	n-fold cross validation	1,081	393	688

Table 1. Statistics of the three Twitter datasets used in this paper.

Stanford Twitter Sentiment Corpus (STS)

This dataset consists of 60,000 tweets randomly selected from the Stanford Twitter Sentiment corpus (STS) [4]. Half of the tweets in this dataset contains positive emoticons, such as :), :-), :), :D, and =), and the other half contains negative emoticons such as :(, :-(, or : (. The original dataset from [4] contained 1.6 million general tweets, and its test set of manually annotated tweets consisted of 177 negative and 182 positive 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 product names, companies and people. To extend the testing set, we added 641 tweets randomly selected from the original dataset, and annotated manually by 12 users (researchers in our lab), where each tweet was annotated by one user. Our final STS dataset consists of

60K general tweets, with a test set of 1,000 tweets of 527 negatively, and 473 positively annotated ones.

Health Care Reform (HCR)

The Health Care Reform (HCR) dataset was built by crawling tweets containing the hashtag "#hcr" (health care reform) in March 2010 [14]. A subset of this corpus was manually annotated with three polarity labels (*positive, negative, neutral*) and split into training and test sets. In this paper, we focus on identifying positive and negative tweets, and therefore we exclude neutral tweets from this dataset. Identifying neutral tweets is part of our future work plan. The final HCR dataset for training contains 839 tweets, and another 839 tweets were used for testing.

Obama-McCain Debate (OMD)

The Obama-McCain Debate (OMD) dataset was constructed from 3,238 tweets crawled during the first U.S. presidential TV debate in September 2008 [13]. Sentiment ratings of these tweets were acquired using Amazon Mechanical Turk, where each tweet was rated by one or more voter as either *positive, negative, mixed*, or *other*. 'Other" tweets are those that couldn't be rated. We only keep those tweet rated by at least three voters with half of the votes being either *positive* or *negative* to ensure their sentiment polarity. This resulted in a set of 1,081 tweets with 393 positive and 688 negative ones. Due to the relative small size of this dataset, and the lack of a test set, we opted for a 5-fold cross validation approach instead.

4 Semantic Features for Sentiment Analysis

This section describes our semantic features and their incorporation into our sentiment analysis method. As mentioned earlier, the semantic concepts of entities extracted from tweets can be used to measure the overall correlation of a group of entities (e.g. all Apple products) with a given sentiment polarity. Hence adding such features to the analysis could help identifying the sentiment of tweets that contain any of the entities that such concepts represent, even if those entities never appeared in the training set (e.g. a new gadget from Apple).[6]

Semantic features refer to those semantically hidden concepts extracted from tweets [11, 12]. An example for using semantic features for sentiment classifier training is shown in Figure 1 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.

4.1 Extracting Semantic Entities and Concepts

There are several open APIs that provide entity extraction services for online textual data. Rizzo and Troncy [10] evaluated the use of five popular entity extraction tools

[6] Assuming of course that the entity extractor successfully identify the new entities as sub-types of concepts already correlated with negative or positive sentiment

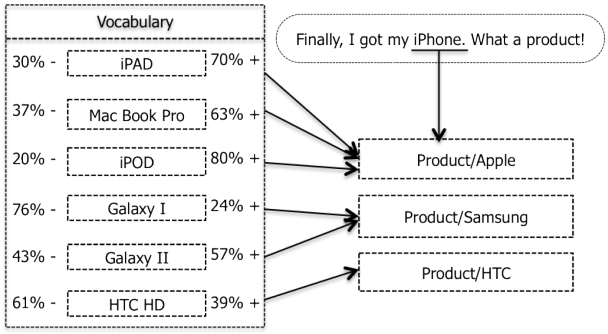


Fig. 1. Measuring correlation of semantic concepts with negative/positive sentiment. These semantic concepts are then incorporated in sentiment classification.

on a dataset of news articles, including AlchemyAPI, DBPedia Spotlight,[7] Extractiv,[8] OpenCalais and Zemanta. Their experimental results showed that AlchemyAPI performs best for entity extraction and semantic concept mapping. Our datasets consist of informal tweets, and hence are intrinsically different from those used in [10]. Therefore we conducted our own evaluation, and randomly selected 500 tweets from the STS corpus and asked 3 evaluators to evaluate the semantic concept extraction outputs generated from AlchemyAPI, OpenCalais and Zemanta.

	No. of Concepts	Entity-Concept Mapping Accuracy (%)		
Extraction Tool	Extracted	Evaluator 1	Evaluator 2	Evaluator 3
AlchemyAPI	108	73.97	73.8	72.8
Zemanta	70	71	71.8	70.4
OpenCalais	65	68	69.1	68.7

Table 2. Evaluation results of AlchemyAPI, Zemanta and OpenCalais.

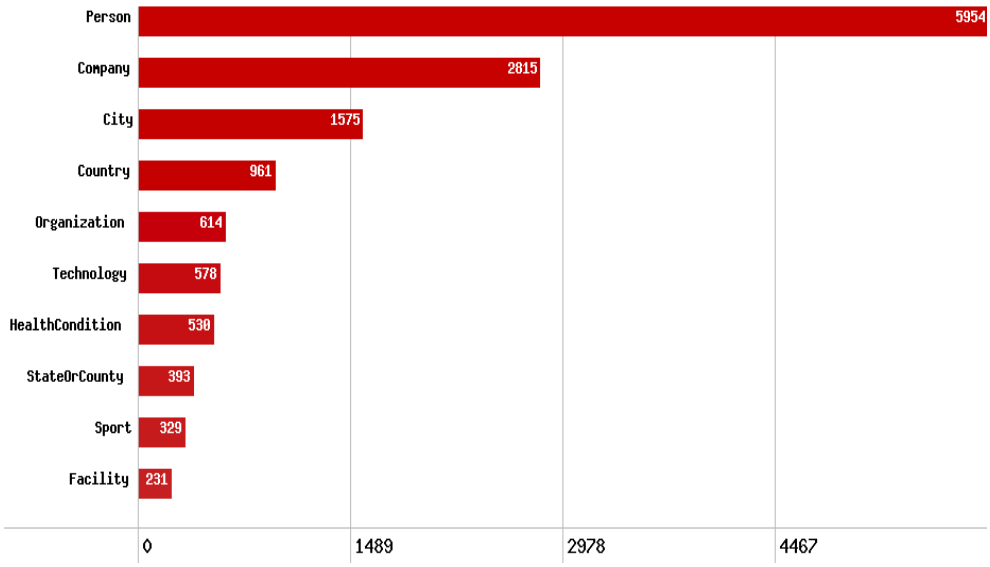
The assessment of the outputs was based on (1) the correctness of the extracted entities; and (2) the correctness of the entity-concept mappings. The evaluation results presented in Table 2 show that AlchemyAPI extracted the most number of concepts and it also has the highest entity-concept mapping accuracy compared to OpenCalais and Zematna. As such, we chose AlchemyAPI to extract the semantic concepts from our three datasets. Table 3 lists the total number of entities extracted and the number of semantic concepts mapped against them for each dataset.

	STS	HCR	OMD
No. of Entities	15139	723	1194
No. of Concepts	29	17	14

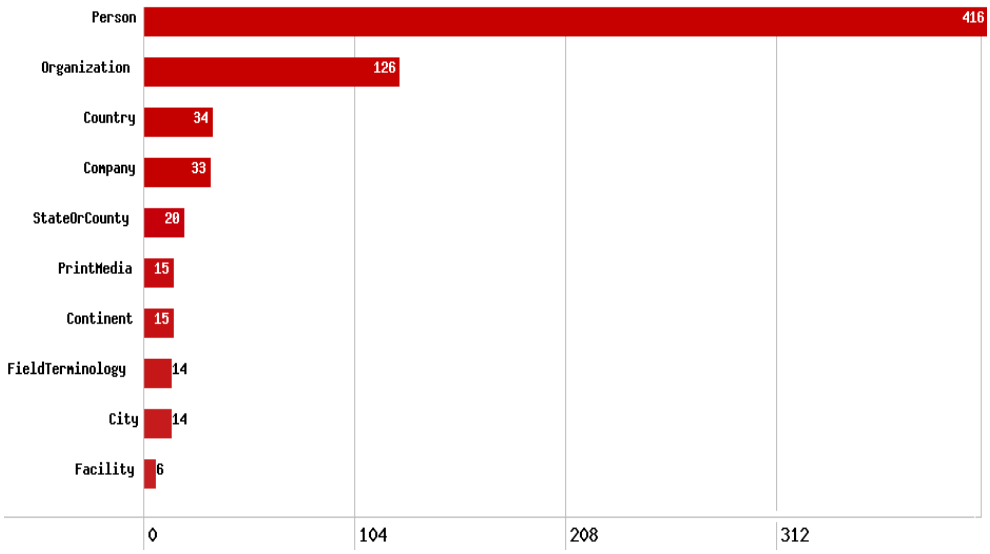
Table 3. Entity/concept extraction statistics of STS, OMD and HCR using AlchemyAPI.

[7] http://dbpedia.org/spotlight/

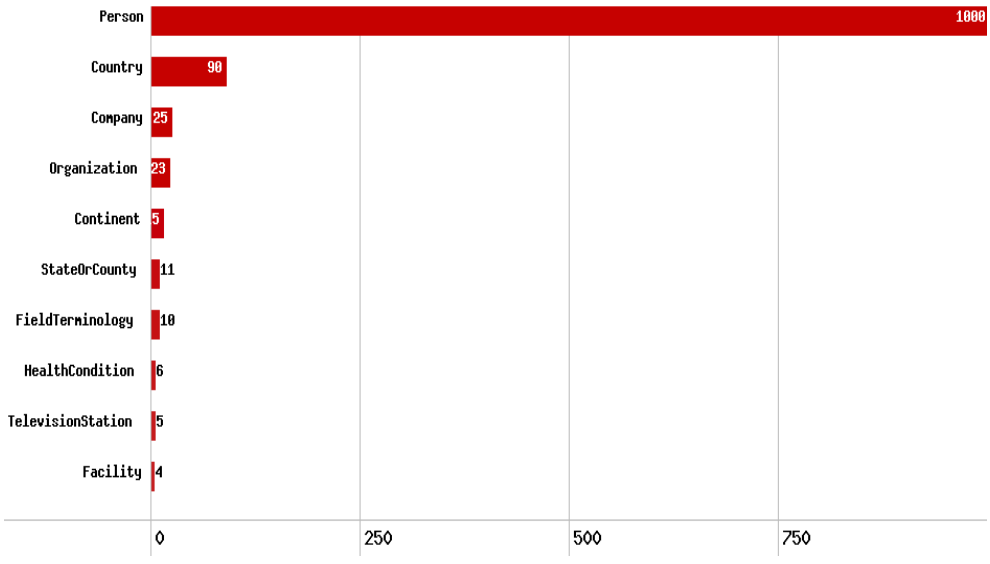
[8] http://wiki.extractiv.com/w/page/29179775/Entity-Extraction



(a) STS.



(b) HCR.



(c) OMD.

Fig. 2. Top 10 frequent concepts extracted with the number of entities associated with them.

Figure 2 shows the top ten high-level extracted concepts from the three datasets with the number of entities associated with each of concept. It can be observed that the most frequent semantic concept is PERSON across all the three corpora. The next two most frequent concepts are COMPANY and CITY for STS, ORGANISATION and COUNTRY for HCR, and COUNTRY and COMPANY for OMD. The level of specificity of these concepts is determined by AlchemyAPI.

4.2 Semantic Feature Incorporation

In this section, we propose three different methods to incorporate semantic features into Naive Bayes (NB) classifier training. We start by an overview of the NB followed by our proposed incorporation methods.

NB is a probabilistic classifier, where the assignment of a sentiment class c to a given tweet $\mathbf{w}$ can be computed as:

$$\begin{aligned}\hat{c} &= \arg\max_{c \in \mathcal{C}} P(c|\mathbf{w}) \\ &= \arg\max_{c \in \mathcal{C}} P(c) \prod_{1 \leq i \leq N_{\mathbf{w}}} P(w_i|c),\end{aligned} \tag{1}$$

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:

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

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.

To incorporate semantic concepts into NB learning, we propose three different methods as described below.

Semantic Replacement: In this method, we replace all entities in tweets with their corresponding semantic concepts. This leads to the reduction of the vocabulary size, where the new size is determined by:

$$|V'| = |V| - |W_{\text{entity}}| + |S|, \tag{3}$$

where $|V'|$ is the new vocabulary size, $|V|$ is the original vocabulary size, $|W_{\text{entity}}|$ is the total number of unique entity words that have been replaced by the semantic concepts, and $|S|$ is the the total number of semantic concepts.

Semantic Augmentation: This method augments the original feature space with the semantic concepts as additional features for the classifier training. The size of the vocabulary in this case is enlarged by the semantic concepts introduced:

$$|V'| = |V| + |S|. \tag{4}$$

Semantic Interpolation: A more principal way to incorporate semantic concepts is through interpolation where we interpolate the unigram language model in NB with the generative model of words given semantic concepts. We propose a general interpolation method below which is able to interpolate arbitrary type of features such as semantic concepts, POS sequences, sentiment-topics etc.

Thus, the new language model with interpolation has the following formula:

$$P_f(W|C) = \alpha P_u(W|C) + \sum_i \beta_i P(W, F_i, C) \tag{5}$$

Where $P_f(W|C)$ is the new language model with interpolation, $P_u(W|C)$ is the original unigram class model and can be calculated using the maximum likelihood estimation, $P(W, F_i, C)$ is the interpolation component, and F_i is a feature vector of type i. The coefficients α and β_i are used to control the influence of the interpolated features in the new language model where:

$$\alpha + \sum_i \beta_i = 1$$

By setting α to 1 the class model becomes a unigram language model without any feature interpolation. On the other hand, setting α to 0 reduces the class model to a feature mapping model. In this work, values of these coefficients have been set by conducting a sensitivity test on the three corpora as will be discuss in Section 6.2.

The interpolation component in the equation 5 can be decomposed as follows:

$$P(W, F_i, C) = \sum_j P(W|f_{ij}) P(f_{ij}|C) \tag{6}$$

Where f_{ij} is the j-th feature of type i, $P(f_{ij}|C)$ is the distribution of features f_{ij} in the training data given the class C and $P(W|f_{ij})$ is the distribution of words in the training data given the feature f_{ij}. Both distributions can be computed via the maximum likelihood estimation.

5 Baselines

We compare the performance of our semantic sentiment analysis approach against the baselines described below.

5.1 Unigrams Features

Word unigrams are the simplest features are being used for sentiment analysis of tweets data. Models trained from word unigrams were shown to outperform random classifiers by a decent margin of 20% [1]. In this work, we use NB classifiers trained from word unigrams as our first baseline model. Table 4 lists, for each dataset, the total number of the extracted unigram features that are used for the classification training.

5.2 Part-of-Speech Features

POS features are common features that have been widely used in the literature for the task of Twitter sentiment analysis. In this work, we build various NB classifiers trained

Dataset	No. of Unigrams
STS	37054
HCR	2060
OMD	2364

Table 4. Total number of unigram features extracted from each dataset.

using a combination of word unigrams and POS features and use them as baseline models. We extract the POS features using the TweetNLP POS tagger,[9] which is trained specifically from tweets. This differs from the previous work, which relies on POS taggers trained from treebanks in the newswire domain for POS tagging. It was shown that TweetNLP tagger outperforms the Stanford tagger[10] with a relative error reduction of 25% when evaluated on 500 manually annotated tweets [3]. Moreover, the tagger offers additional recognition capabilities for abbreviated phrases, emoticons and interjections (e.g. "lol", "omg").

5.3 Sentiment-Topic Features

The sentiment-topic features are extracted from tweets using the weakly-supervised joint sentiment-topic (JST) mode that we developed earlier [7]. We trained this model on the training set with tweet sentiment labels discarded. The resulting model assigns each word in tweets with a sentiment label and a topic label. Hence JST essentially groups different words that share similar sentiment and topic.

We list some of the topic words extracted by this model from the STS and OMD corpora in Table 5. 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 sentiment. For example, Topic 2 under positive sentiment is about the movie "Twilight", while Topic 5 under negative sentiment is about a complaint of feeling sick possibly due to cold and headache. The rational behind this model is that grouping words under the same topic and bearing similar sentiment could reduce data sparseness in Twitter sentiment classification and improves accuracy.

6 Evaluation Results

In this section, we evaluate the use of the sentiment features discussed in 4 and present the sentiment identification results on the STS, HCR and OMD datasets. We then compare these results with those obtained from using the baseline features described in Section 5.

We use NB trained from word unigrams as the starting-point baseline model. The features are incorporated into NB by either the interpolation approach described in Section 4.2 or by simply augmenting into the original bag-of-words feature space. For evaluation on STS and HCR, we use the training and testing sets shown in Table 1. For OMD, we perform 5-fold cross validation and report the results averaged over 10 such runs.

The raw tweets data can be very noisy, and hence some pre-processing was necessary, such as replacing all hyperlinks with "*URL*", converting some words with apostro-

[9] `http://www.ark.cs.cmu.edu/TweetNLP/`

[10] `http://nlp.stanford.edu/software/tagger.shtml`

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Positive	win	twilight	make	tomorrow	today
	final	movie	mccain	weekend	nice
	watch	award	debate	school	Sunday
	game	moon	good	start	enjoy
	luck	tonight	point	plan	weather
	today	watch	interest	fun	love
	week	mtv	right	yai	walk
	hope	excited	answer	wait	sunny
Negative	iphone	dog	obama	miss	feel
	internet	sad	question	far	sick
	download	death	understand	travel	bad
	apple	accident	doesn't	mum	hurt
	store	today	answer	away	pain
	slow	car	comment	dad	flu
	issue	awful	back	love	sore
	crash	cry	debate	country	horrible

Table 5. Extracted sentiment-topic words by the sentiment-topic model.

phe, such as "*hate'n*", to their complete form "*hating*", removing repeated letters (e.g. "*loovee*" becomes "*love*"), etc.

6.1 Stopwords

It is a common practice to perform stopwords removal as a standard pre-processing step by removing those common words which tend to have little meaning. Nevertheless, Bei [16] argued that stopwords can be used as discriminative features for specific classification tasks. We have conducted a set of experiments to evaluate the impact of stopwords removal on sentiment classification on tweets. We compare the performance of a NB classifier trained from word unigrams before and after removing the stopwords. It can be observed from Table 6 that the classifiers learned with stopwords outperform those learned with stopwords removed. Similar outcome was observed when using all out sentiment analysis features. Hence, we chose to keep the stopwords in our subsequent experiments.

Dataset	With Stopwords	Without Stopwords
Stanford Twitter Sentiment (STS)	**80.7%**	77.5%
Health Care Reform (HCR)	**71.1%**	68.5%
Obama-McCain Debate (OMD)	**75.4%**	73.7%

Table 6. Sentiment classification accuracy ((True Positives + True Negatives) / Total) with and without stopwords using unigram features.

6.2 Incorporating Semantic Features

Semantic features can be incorporated into NB training in three different ways, *replacement*, *augmentation*, and *interpolation* (Section 4.2). Table 7 shows the F measures produced when using each of these feature incorporation methods. With *semantic re-*

placement, where all entities in tweets are *replaced* with their corresponding semantic concepts, the feature space shrunk substantially by nearly 15-20%, and produced an average F measure of 68.9%. However, this accuracy is 3.5% and 10.2% less than when using semantic augmentation and interpolation respectively. The performance degradation is due to the information loss caused by this term replacement which subsequently hurts NB performance.

Augmenting the original feature space with semantic concepts (*semantic augmentation*) performs slightly better than *sentiment replacement*, though it still performs 6.5% worse than interpolation. With *Semantic interpolation*, semantic concepts are incorporated into NB training taking into account the generative probability of words given concepts. This method produces the highest accuracy amongst all three incorporation methods, with an average F of 75.95%.

Method	STS	HCR	OMD	Average
Semantic replacement	74.10	61.35	71.25	68.90
Semantic augmentation	77.65	63.65	72.70	71.33
Semantic interpolation	**83.90**	**66.10**	**77.85**	**75.95**

Table 7. Average sentiment classification accuracy (%) using different methods for incorporating the semantic features. Accuracy here is the average harmonic mean (F measure) obtained from identifying positive and negative sentiment.

The contribution of semantic features in the interpolation model is controlled by the interpolation coefficients in Equation 5. We conducted a sensitivity test to evaluate the impact of the interpolation coefficients on sentiment classification accuracy by varying β between 0 and 1. Figure 3 shows that accuracy reaches its peak with β set between 0.3 and 0.5. In our evaluation, we used 0.4 for STS dataset, and 0.3 for the other two.

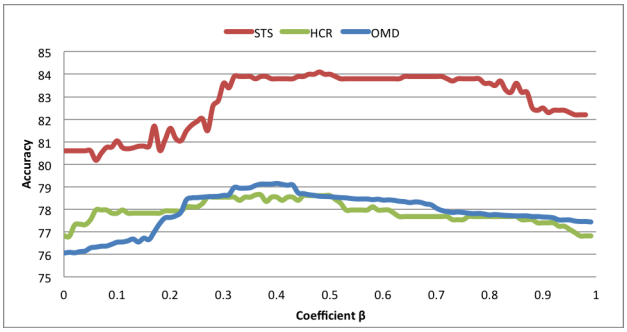


Fig. 3. Sensitivity test of the interpolation coefficient for semantic interpolation.

6.3 Comparison of Results

In this section we will compare the Precision, Recall, and F measure of our semantic sentiment analysis against the baselines described in Section 5. We report the semantic

classification results for identifying positive and negative sentiment separately to allow for deeper analysis of results. This is especially important given how some analysis methods perform better in one sentiment polarity than in the other.

Table 8 shows the results of our sentiment classification using *Unigrams*, *POS*, *Sentiment-Topic*, and *Semantic* features, applied over the STS, HCR, and OMD datasets which are detailed in Section 3. The table reports three sets of P, R, and F1, one for positive sentiment identification, one for negative sentiment identification, and the third shows the averages of the two.

Dataset	**Feature**	**Positive Sentiment**			**Negative Sentiment**			**Average**		
		P	R	F1	P	R	F1	P	R	F1
STS	Unigrams	82.20	75.20	78.50	79.30	85.30	82.20	80.75	80.25	80.35
	POS	83.70	75.00	79.10	79.50	86.90	83.00	81.60	80.95	81.05
	Sentiment-Topic	80.70	82.20	81.40	83.70	82.30	83.00	82.20	82.25	82.20
	Semantics	85.80	79.40	82.50	82.70	88.20	85.30	**84.25**	**83.80**	**83.90**
HCR	Unigrams	39.00	36.10	37.50	81.00	82.80	81.90	60.00	59.45	59.70
	POS	56.20	22.00	31.70	80.00	94.70	86.70	68.10	58.35	59.20
	Sentiment-Topic	53.80	47.20	50.30	84.50	87.60	86.00	**69.15**	**67.40**	**68.15**
	Semantics	53.60	40.40	46.10	83.10	89.30	86.10	68.35	64.85	66.10
OMD	Unigrams	64.20	70.90	67.10	83.30	78.60	80.80	73.75	74.75	73.95
	POS	69.50	68.30	68.70	83.10	83.90	83.40	76.30	76.10	76.05
	Sentiment-Topic	68.20	75.60	71.70	87.10	82.40	84.70	77.65	**79.00**	**78.20**
	Semantics	75.00	66.60	70.30	82.90	88.10	85.40	**78.95**	77.35	77.85

Table 8. Cross comparison results of all the four features.

According to these results in Table 8, the Semantic approach outperforms the Unigrams and POS baselines in all categories and for all three datasets. However, for the HCR and OMD datasets, the sentiment-topic analysis approach seem to outperform the semantic approach by a small margin. For example, the semantic approach produced higher P, R, and F1 for the STS dataset, with F1 4.4% higher than Unigrams, 3.5% higher than POS, and 2.1% higher than the sentiment-topic features. In HCR, F1 from the semantic features were 8.9% and 11.7% higher than Unigrams and POS, but 3% lower than F1 from sentiment-topic features. For OMD, semantic features also outperformed the Unigrams and POS baselines, with 5.2% and 2.4% higher F1 respectively. However, in the OMD dataset, F1 from semantic features was 0.4% lower than from the topic model, although Precision was actually higher by 1.7%.

As detailed in Section 3 and Table 1, the STS dataset consists of a large collection of general tweets with no particular topic focus. Unlike STS, the other two datasets are much smaller in size and their tweets discuss very specific topics; the US Health Care Reform bill in the HCR dataset, and the Obama McCain debate in the OMD dataset. Using semantic features seem to perform best in the large and general dataset, whereas the sentiment-topic features seem to take the lead in small, topic-focused datasets. The reason is likely to be that classifying with sentiment-topic features group words into a number of topics. In our experiments, we found that for the STS dataset, increasing the number of topics leads to the increase of classification accuracy with the peak value of 82.2% reached at topic number 50. Further increasing topic numbers degrades the

classifier performance. However, for HCR and OMD, the best accuracy was obtained with only one topic (68.15% for HCR and 78.20% for OMD). The classification performance drops significantly by any further increment. This can be explained by the nature of these three datasets. HCR was collected using the hashtag "#hcr" (health care reform) while OMD consists of tweets about the Obama-McCain debate. Hence these two datasets are topic-specific. On the contrary, STS was collected using more general queries and thus it contains a potentially large number of topics.

Hence the benefits of using the sentiment-topic features seem to be reduced in comparison to semantic features when the training set is of general content as in the STS tweets dataset.

The average results across all three datasets are shown in Table 9. Here we can see that semantic features do better than sentiment-topic features and the other baselines when identifying *negative sentiment*. However, sentiment-topic features seem to perform better for *positive sentiment*. For positive sentiment, using the semantic approach produces Precision that is better than Unigrams, POS, and sentiment-topic by 15.6%, 2.4%, and 5.8% respectively. However, the Recall produced by the semantic approach when identifying positive sentiment is 2.3% and 12.8% higher than in Unigrams and POS, but 9% lower than Recall from the sentiment-topic approach. Overall, F for positive sentiment from semantic features is 2.2% lower than when using sentiment-topic features. It is worth emphasising that the average Precision from identifying both positive and negative sentiment is the highest at 77.18% when using semantic features. When analysing large amounts of continuously flowing data as with social media resources, Precision could well be regarded as much more important than Recall.

Features	**Positive Sentiment**			**Negative Sentiment**			**Average**		
	P	R	F1	P	R	F1	P	R	F1
Unigrams	61.80	60.73	61.03	81.20	82.23	81.63	71.50	71.48	71.33
POS	69.80	55.10	59.83	80.87	88.50	84.37	75.53	72.23	72.48
Sentiment-Topic	67.57	**68.33**	**67.80**	**85.10**	84.10	84.57	77.02	**76.73**	**76.75**
Semantics	**71.47**	62.13	66.30	82.90	**88.53**	**85.60**	**77.18**	75.33	75.95

Table 9. Averages of Precision, Recall, and F measures across all three datasets.

7 Discussion and Future Work

In this paper we demonstrated the value of using semantic features for the classification of positive and negative sentiment in Tweets. We tested several off-the-shelf semantic entity extractors and decided on using AlchemyAPI due to its better performance in terms of coverage and accuracy. One thing that impacts our results is the abstraction level of the concepts retrieved from the entity extractor. In many cases, these concepts were too abstract (e.g. Person) which were equally used for mentions of ordinary people, as well as for famous musicians or politicians. For the tweet "i wish i could go to france and meet president Obama haha", AlchemyAPI provided the concept *Person* to represent "president Obama", whereas Zemanta identified him with the concept */government/politician* which is more specific. In future work we plan to devise an approach to increase the specificity of such concepts, perhaps with the aid of DBpedia or

using multiple entity extractors and comparing the specificity level of their proposed concepts.

In our evaluation of AlchemyAPI, Zemanta, and OpenCalais, we observed that some of them perform better than others for specific type of entities. For example, Zemanta produced more accurate and specific concepts to describe entities related to music tracks and bands. It might be possible to implement a more selective approach, where certain semantic extractors and concept identifiers are used, or trusted more, for certain type of entities.

When using semantic features, all identified concepts in a tweet are added to the analysis. However, it might be the case that semantic features improve sentiment analysis accuracy for some type of concepts (e.g. cities, music) but reduce accuracy in some other concept types (e.g. people, companies). We will investigate the impact of each group of concepts on our analysis accuracy, to determine their individual contribution and impact on our sentiment analysis. We can also assign weights to each concept type to represent its correlation with positive or negative sentiment.

We experimented with multiple datasets of varying sizes and topical-focus. Our results showed that the accuracy of classifying with some feature selections can be sensitive to the size of the datasets and their topical-focus. For example, our evaluation showed that the semantic approach excels when the dataset is large and of diverse topic coverage. In future work we will apply these approaches on larger datasets to examine the consistency of their performance patterns. Furthermore, we also intend to explore various feature selection strategies to improve the sentiment classification performance.

Our sentiment analysis focused on positive and negative tweets. Neutral sentiment tend to be much harder to identify as it requires the determination of the context of the tweet message. For example, some words of a tweet may have both subjective and objective senses. Handling such tweets will therefore require the introduction of another classifier, to identify subjective/objective tweets.

8 Conclusions

We proposed the use of semantic features in Twitter sentiment classification and explored three different approaches for incorporating them into the analysis; with replacement, augmentation, and interpolation. We found that best results are achieved when interpolating the generative model of words given semantic concepts into the unigram language model of the NB classifier. We conducted extensive experiments on three Twitter datasets and compared the semantic features with the the Unigrams and POS sequence features as well as with the sentiment-topic features. Our results show that the semantic feature model outperforms the Unigram and POS baseline for identifying both negative and positive sentiment. We demonstrated that adding semantic features produces higher Recall and F1 score, but lower Precision, than sentiment-topic features when classifying negative sentiment. We also showed that using semantic features outperforms the sentiment-topic features for positive sentiment classification in terms of Precision, but not in terms of Recall and F1. One average, the semantic features appeared to be the most precise amongst the four other feature selections we experimented with.

Our results indicates that the semantic approach is more appropriate when the datasets being analysed are large and cover a wide range of topics, whereas the sentiment-topic approach was most suitable for relatively small datasets with specific topical foci.

We believe that our findings demonstrated the high potential of the novel approach of interpolating semantic features into the sentiment classifier. In our current implementation, we rely on Alchemy API which is only able to produce rather coarse semantic concept mappings. However, our results indicate that further gains could be achieved when entities are mapped into a more fine-grained semantic concept space.

Acknowledgment

The work of the authors was supported by the EU-FP7 project ROBUST (grant no. 257859).

References

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proc. ACL 2011 Workshop on Languages in Social Media. pp. 30–38 (2011)
2. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of COLING. pp. 36–44 (2010)
3. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.: Part-of-speech tagging for twitter: Annotation, features, and experiments. Tech. rep., DTIC Document (2010)
4. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford (2009)
5. Guerra, P., Veloso, A., Meira Jr, W., Almeida, V.: From bias to opinion: A transfer-learning approach to real-time sentiment analysis. In: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2011)
6. Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg! In: Proceedings of the ICWSM (2011)
7. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceeding of the 18th ACM conference on Information and knowledge management. pp. 375–384. ACM (2009)
8. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Proceedings of the European Conference on Machine Learning. pp. 318–329 (2006)
9. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of LREC 2010 (2010)
10. Rizzo, G., Troncy, R.: Nerd: Evaluating named entity recognition tools in the web of data. In: Workshop on Web Scale Knowledge Extraction (WEKEX11). vol. 21 (2011)
11. Saif, H., He, Y., Alani, H.: Semantic Smoothing for Twitter Sentiment Analysis. In: Proceeding of the 10th International Semantic Web Conference (ISWC) (2011)
12. Saif, H., He, Y., Alani, H.: Alleviating Data Sparsity for Twitter Sentiment Analysis. In: Proceedings, 2nd Workshop on Making Sense of Microposts (#MSM2012): Big things come in small packages: in conjunction with WWW 2012 (2012)
13. Shamma, D., Kennedy, L., Churchill, E.: Tweet the debates: understanding community annotation of uncollected sources. In: Proceedings of the first SIGMM workshop on Social media. pp. 3–10. ACM (2009)
14. Speriosu, M., Sudan, N., Upadhyay, S., 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 pp. 53–63 (2011)
15. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2011)
16. Yu, B.: An evaluation of text classification methods for literary study. Literary and Linguistic Computing 23(3), 327–343 (2008)