Modelling Public Sentiment in Twitter: Using Linguistic Patterns to Enhance Supervised Learning

Prerna Chikersal[1], Soujanya Poria[1], Erik Cambria[1],
Alexander Gelbukh[2], and Chng Eng Siong[1]

[1] School of Computer Engineering, Nanyang Technological University, Singapore
[2] Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico
prerna1@e.ntu.edu.sg, {sporia,cambria,aseschng}@ntu.edu.sg,
gelbukh@cic.ipn.mx
http://sentic.net

Abstract. This paper describes a Twitter sentiment analysis system that classifies a tweet as positive or negative based on its overall tweet-level polarity. Supervised learning classifiers often misclassify tweets containing conjunctions such as "but" and conditionals such as "if", due to their special linguistic characteristics. These classifiers also assign a decision score very close to the decision boundary for a large number tweets, which suggests that they are simply unsure instead of being completely wrong about these tweets. To counter these two challenges, this paper proposes a system that enhances supervised learning for polarity classification by leveraging on linguistic rules and sentic computing resources. The proposed method is evaluated on two publicly available Twitter corpora to illustrate its effectiveness.

Keywords: Opinion Mining, Sentiment Analysis, Sentic Computing.

1 Introduction

Nowadays, an increasing number of people are using social media to express their opinions on various subjects, as a result of which a vast amount of unstructured opinionated data has become available. By analysing this data for sentiments, we can infer the public's opinion on several subjects and use the conclusions derived from this to make informed choices and predictions concerning those subjects [1]. However, due to the volume of data generated, manual sentiment analysis is not feasible. Thus, automatic sentiment analysis is becoming exceedingly popular [2].

Polarity classification is a sub-task of sentiment analysis that focusses on classifying text into positive and negative, or positive, negative and neutral. Document-level polarity classification involves determining the polarity of opinions expressed in an entire document, whereas sentence-level polarity classification involves determining the polarity of opinions expressed in a single sentence. Another level of polarity classification is aspect-based polarity classification, which involves extracting opinion targets from text and then determining the polarity of the text towards that particular target. Surveys of methods used for various levels of sentiment analysis can be found in [3,4,5].

A. Gelbukh (Ed.): CICLing 2015, Part II, LNCS 9042, pp. 49–65, 2015.
DOI: 10.1007/978-3-319-18117-2_4

Tweets are short microblogging texts containing a maximum of 140 characters. They can include multiple sentences and often contain misspelled words, slangs, URLs, elongations, repeated punctuations, emoticons, abbreviations and hashtags. These characteristics make extracting sentiment and opinions from tweets a challenge, and hence an interesting topic of research. This paper focusses on tweet-level polarity classification, which involves predicting the overall polarity of opinions expressed in a tweet. We focus on classifying the tweets into positive or negative, and ignore the neutral class.

This paper is organised as follows. Section 2 explains the motivation or need for the proposed method; Section 3 briefly discusses related research; Section 4 describes the method; Section 5 presents and analyses the experimental results obtained; finally, Section 6 concludes the paper.

2 Motivation for Our Method

Supervised learning classifiers commonly used for polarity classification rely on feature vectors extracted from the text to represent the most important characteristics of the text. Word N-grams, which are denoted by the frequencies of contiguous sequences of 1, 2, or 3 tokens in the text, are the most commonly used features for supervised sentiment analysis. While such classifiers [6,7,8] have been shown to perform reasonably well, studies such as [9], [10] and [11] show that using a "one-technique-fits-all" solution for all types of sentences is not good enough due to the diverse types of linguistic patterns found in sentences. That is, the presence of modal verbs such as "could" and "should", conjunctions such as "but" and "or" and conditionals such as "if", "until", "unless", and "in case" in a text substantially worsen the predictions of a supervised classifier.

Furthermore, supervised learning classifiers classify each tweet with a certain probability or decision (confidence) score. For a large number of tweets, the decision score predicted by a typical supervised classifier is very close to the decision boundary. This implies that the classifier is unsure about which class the tweets in question belong to and so cannot assign class labels to them with much confidence. Thus, the class labels assigned to such tweets are either completely incorrect or correct mostly by fluke.

To prove this notion, we train a Support Vector Machine (SVM) classifier using n-grams (n = 1,2,3) as features on ≈ 1.6 million tweets provided by [8] and test it on 1794 positive and negative tweets provided by [12] and plot the decision scores computed by the SVM in Figure 1. In the graph, we can see that frequency of misclassifications reduce as we move away from the decision boundary ($y = 0$). We find that 341 tweets out of 1794 tweets are misclassified by the SVM, however 239 out of the 341 misclassified tweets have a decision score that lies between -0.5 and $+0.5$. Thus, the SVM is simply unsure instead of being completely wrong about these 239 tweets. If we consider all the predictions of the SVM, we get a misclassification rate[1] of $\approx 19\%$. But, if we exclude all the predictions whether right (475 tweets) or wrong (239 tweets) with a decision score between -0.5 and $+0.5$, we get a misclassification rate of only $\approx 9.4\%$. This means that if we consider the classification of only those tweets that the SVM is confident about, we can say that it correctly classifies over 90% of the tweets!

[1] misclassification rate $= \frac{Number\ of\ Incorrect\ Classifications}{Total\ Number\ of\ Classifications}$

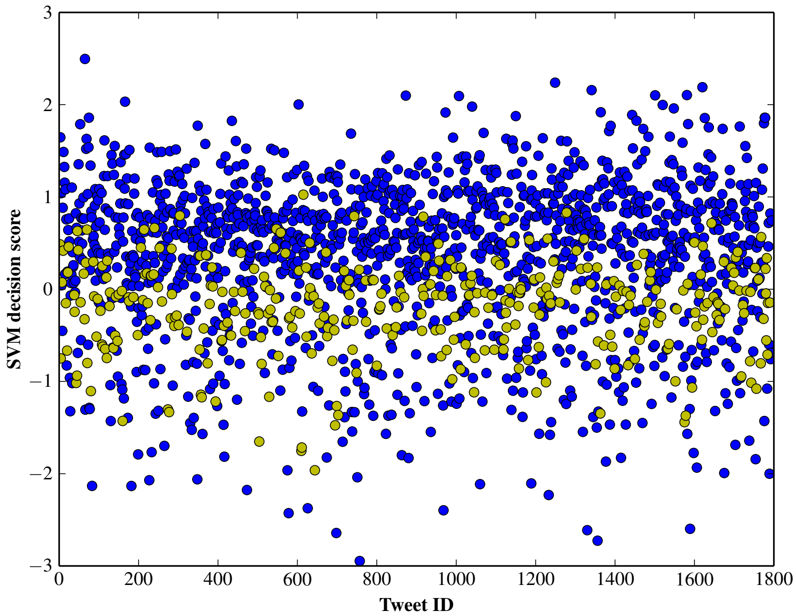

Fig. 1. SVM decision scores plotted for the classification of 1794 tweets into positive or negative using n-grams (n=1,2,3) as features. Tweets with decision score above 0 are labelled as positive, while tweets with decision score below 0 are labelled as negative. Blue = correctly classified (1453 tweets), Green = misclassified (341 tweets). 239 out of the 341 misclassified tweets have decision score between -0.5 and +0.5, implying that the SVM is simply unsure about them.

So, from the above, we can deduce that it would be beneficial to design a classifier that:

– Can handle special parts-of-speech of grammar such as conjunctions and conditionals.
– Uses a secondary (high-confidence) classifier to verify or change the classification labels of the tweets the SVM computes a very low decision or confidence score for.

To handle the special parts-of-speech of grammar, we modify the n-gram features provided as input to the classifier, based on linguistic analysis of how these parts-of-speech are used in sentences [13]. The scope of the method proposed in this paper is limited to the conjunction "but" and the conditionals "if", "unless", "until" and "in case".

Furthermore, we design an unsupervised rule-based classifier to verify or change the classification labels of the tweets the SVM computes a very low decision score for. The rules used by this classifier are based on our linguistic analysis of tweets, and leverage on sentiment analysis resources that contain polarity values of words and phrases. The primarily resource used for this purpose is SenticNet [14] – a semantic and affective resource for concept-level sentiment analysis, which basically assigns polarity values to concepts taken from a common-sense knowledge base called ConceptNet [15].

As human beings, we are able to understand the meaning of texts and determine the sentiment conveyed by them. Our common-sense plays a very important role in this process by helping us estimate the polarities of commonly used single-word and multi-word expressions or concepts occurring in text, and then use the relationships between words and concepts to ascertain the overall polarity. For example, say a text contains the phrase "good morning"; how do you interpret it and estimate its polarity? Luckily, depending on the context, our common-sense helps us deduce whether the expression "good morning" is used as a wish, as a fact, or as something else. Otherwise, without common-sense, we would need to ask each other questions such as

> *"Do you wish me a good morning, or mean that it is a good morning whether I want it or not; or that you feel good this morning; or that it is a morning to be good on?"* – J.R.R. Tolkien (from The Hobbit)

Moreover, the estimated polarity of the expression "good morning" cannot merely be the sum of the polarities of the words "good" and "morning". Hence, unlike most sentiment analysis methods, we prefer to break tweets into concepts and query those concepts in SenticNet, instead of relying completely on bag-of-words queried in lexicons containing word-level polarities.

3 Related Work

In this section, we briefly review some concepts and commonly used techniques for sentiment analysis that are relevant to the method proposed in this paper.

3.1 Supervised Learning for Sentiment Analysis

A text sample is converted to a feature vector that represents its most important characteristics. Given the feature vectors X and class labels Y for N number of training tweets, the supervised learning algorithm approximates a function F such that $F(X) = Y$. Now, in the testing phase, given feature vectors X' for T number of unlabelled tweets, the function F predicts labels Y' using $F(X') = Y'$ for each of the unlabelled tweets.

The most commonly used features for sentiment analysis are *term presence* and *term frequency* of single tokens or unigrams. The use of higher order n-grams (presence or frequency of 2,3,..,n contiguous tokens in a text) such as bigrams and trigrams is also prevalent, and allows for encoding of the tokens' positional information in the feature vector. Parts-of-speech and negation based features are also commonly used in sentiment analysis. Studies such as [16] and [17] focus on techniques used to represent negation, detect negation words, and determine the scope of negation in text.

More recent studies such as [6], [18], [19], and [20], exploit microblogging text or Twitter-specific features such as emoticons, hashtags, URLs, @symbols, capitalisations, and elongations to enhance sentiment analysis of tweets.

3.2 Unsupervised Learning and Linguistic Rules for Sentiment Analysis

Usually, unsupervised approaches for sentiment analysis such as [21] involve first creating a sentiment lexicon in an unsupervised manner, and then determining the polarity of a text using some function dependent on the number or measure of positive and negative words and/or phrases present in the text. A comparison of supervised methods and other unsupervised methods can be found in [22].

In [9], the authors define dependency-based linguistic rules for sentiment analysis, and merge those rules with common-sense knowledge, and machine learning to enhance sentiment analysis. Our proposed method is based on the idea illustrated in [9], however the linguistic rules we define are limited and not dependency based, because most dependency parsers do not perform well for microblogging texts such as tweets. Moreover, it is desirable to perform sentiment analysis of social media texts in real-time, and dependency parsers cannot be used in real-time due to the large time complexity of their algorithms. In this paper, our goal is to create a Twitter sentiment analysis classifier that classifies tweets in real-time while countering the two challenges postulated in 2.

3.3 Concept-Level Sentiment Analysis and Sentic Computing

So far sentiment analysis approaches relying on keyword spotting, word co-occurrence frequencies, and bag-of-words have worked fairly well. However, with increase in user-generated content such as microblogging text and the epidemic of deception phenomenon such as web-trolling and opinion spam, these standard approaches are becoming progressively inefficient. Thus, sentiment analysis systems will eventually stop relying solely on word-level techniques and move onto concept-level techniques. Concepts can be single-word or multi-word expressions extracted from text. Multi-word expressions are often more useful for sentiment analysis as they carry specific *semantics and sentics* [23], which include common-sense knowledge (which people acquire during their early years) and common knowledge (which people gather in their daily lives). The survey in [24] explains how Natural Language Processing research is evolving from methods based on bag-of-words to bag-of-concepts and finally on bag-of-narratives. In this paper, we define linguistic rules which rely on polarity values from a concept-level common-sense knowledge base called SenticNet [14].

4 The Proposed Method

Before analysing raw tweets for sentiments, we pre-process them. During pre-processing, all the *@<username>* references are changes to *@USER* and all the *URLs* are changed to *http://URL.com*. Then, we use the CMU Twitter Tokeniser and Parts-of-Speech Tagger [25] to tokenise the tweets and assign a parts-of-speech tag to each token. Apart from nouns, verbs, adjectives and adverbs, this tagger is also able to tag injunctions, and microblogging-specific tokens such as emoticons, hashtags, and URLs.

The proposed sentiment analysis system is illustrated in Figure 2, and is explained in detail in this section.

4.1 Emoticon Rules

Using the tokens in a tweet and the output of the tagger, we are able to find all the tokens that represent emoticons in the tweet. Since people often repeat certain punctuations to

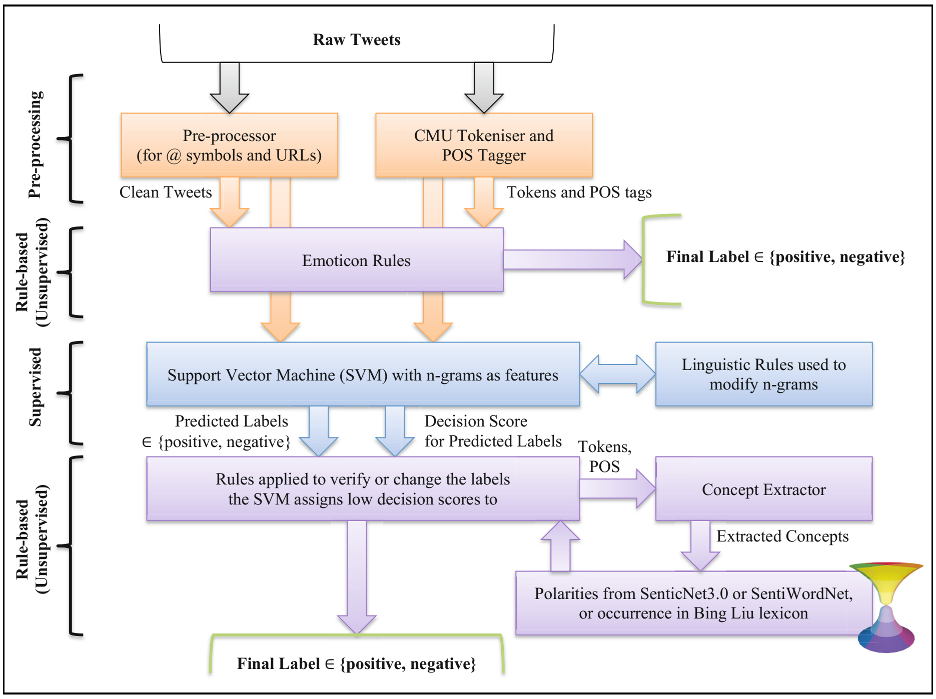

Fig. 2. Flowchart of the Proposed Twitter Sentiment Analysis System

emphasise emoticons, we remove all repeated characters from every emoticon string to obtain the bag-of-emoticons present in the tweet. Table 1 is a manually created list of usually polar emoticons along with their semantic orientation (*positive or negative*). We match emoticons in the bag-of-emoticons of the tweet to the list of positive or negative emoticons, and count the number of positive and the number of negative emoticons present in the tweet.

Table 1. Manually Created List of Positive and Negative Emoticons

Orientation	**List of Emoticons**
Positive	(-: , (: , =) , :) , :-) , =‘) , :‘) , :‘-) , =-d , =d , ;d , :d , :-d , ^_^ , ^_^ , :] , ^_- , ^_* , ^^
Negative	)-: ,): , =(,]: , :[, :(, :-(, >;(, >:(, :_(, d’x , :‘(, :”(, =’[, :’(, :’-(, \: , :/ , (~_~) , >__> , <(’-’)> , </3

Then, we apply the following rules to classify the tweet:

- If a tweet contains one or more positive emoticons and no negative emoticons, it is labeled as *positive*.
- If a tweet contains one or more negative emoticons and no positive emoticons, it is labeled as *negative*.
- If neither of the two rules above apply, the tweet is labeled as *unknown*.

If these emoticon-based rules label a tweet as *positive* or *negative*, we consider that label to be the final label outputted by our system. However, all tweets labelled as *unknown* by these rules are passed into the next stage in our sentiment analysis pipeline, that is the supervised learning classifier.

4.2 Support Vector Machine (SVM) with N-grams as Features

For supervised learning, we represent each tweet as a feature vector of case-sensitive n-grams (unigrams, bigrams, and trigrams). These n-grams are frequencies of sequences of 1, 2 or 3 contiguous tokens in a tweet. The TF-IDF [26] weighting scheme from information retrieval is applied to the frequency counts, and L1 regularisation is used for feature selection and dimensionality reduction. Finally, a Support Vector Machine is trained using the LIBLINEAR library [27].

To account for negation, we append the string "_NEG" to all negated tokens in a tweet. All tokens between certain negation words and the next punctuation mark are considered to be negated, as long as they are either nouns, adjectives, adverbs, or verbs. This is so because negating emoticons, hashtags or URLs would not make sense. Apart, from this, no other feature related to negation is used in the feature vector.

For this purpose, we take into account the following negation words: *never, no, nothing, nowhere, noone, none, not, havent, haven't, hasnt, hasn't, hadnt, hadn't, cant, can't, couldnt, couldn't, shouldnt, shouldn't, wont, won't, wouldnt, wouldn't, dont, don't, doesnt, doesn't, didnt, didn't, isnt, isn't, arent, aren't, aint, ain't.*

In section 4.4, the same method will be to find negated tokens in order to invert their polarity values.

4.3 Modifying N-grams According to Linguistic Rules

As mentioned in section 2, typical supervised learning methods based on n-grams perform badly on sentences containing special parts-of-speech such as conjunctions and conditionals commonly used in grammar, due to their peculiar linguistic characteristics. We theorise that one such characteristic is that a certain part of the sentence either becomes irrelevant for sentiment analysis or possesses a semantic orientation that is opposite to the sentence's overall orientation.

We analyse tweets containing the conjunction "but" and the conditionals "if", "unless", "until", and "in case", and formulate rules that should enable removal of irrelevant or oppositely oriented n-grams from the tweet's feature vector, before it is used for supervised learning.

Below are a few examples of tweets containing "but" at different syntactic positions. In each tweet, the most salient part that is the part that contributes considerably to the overall polarity of the tweet is underlined. In certain tweets however, if no salient part can be found or is ambiguous, nothing is underlined. The overall polarity of the tweet is indicated in parenthesis.

(1) @USER Tell you at our Friday lunch. Sorry for the late reply but yes we can eat somewhere on Marshall tomorrow haha *(positive)*

(2) it may have been against the minnows of FC Gomel, but a great performance from Rodger's Reds at Anfield tonight, and a great start! *(positive)*

(3) SP to support UPA, but oppose anti-people policies: Samajwadi Party on Saturday said it will continue to oppose *(negative)*

(4) Taylor Kitsch may not be a leading man, or able to open a movie, but he was quite good in The Bang Bang Club- Ryan Phillippe as well *(positive)*

(5) S/O to @USER ! I don't really know her but she seems real chill. She may not know how to spell Peyton Siva, but still follow her! *(positive)*

(6) you didnt win ABDC but you won over my heart you may not know me but imma true ICONiac by heart *(positive)*

(7) I laughed a little, but not sharing them out with anyone. Will the weather be good tomorrow for Boris Bikes? *(positive)*

(8) Gutted I'm missing the cardigan match on Saturday! But more important things to do *(negative)*

From the examples above, we observe that the part of the sentence posterior to the word "but" is usually (though not always) a better indicator of the overall polarity of the tweet, as compared to the anterior part. This premise holds true for examples (1) to (6), but does not work for a few examples such as (7) and (8).

In example (7), it is difficult to determine the most salient part of the tweet. This could be because that tweet appears to be only weakly positive, and could even be interpreted as negative if we only consider the posterior part of "but". In example (8), the most salient part of the tweet is anterior to the word "but", perhaps because the polarity of the posterior part is too subtle or even neutral. Nevertheless, in this paper, we will only focus on formulating rules that work for tweets similar to examples (1) through (6), as handling tweets similar to (7) and (8) is too difficult and requires more complex linguistic analysis which is beyond the scope of this study.

Furthermore, it is difficult to automatically determine the salient part in tweets similar to example (6), due to grammatical errors introduced by the writer of the tweet. That is, example (6) should contain 2 separate sentences, but there is no punctuation mark to separate "...my heart" and "you may not know me...", which makes it very hard for us to pick out the phrases "you won over my heart" and "imma true ICONiac by heart" as the most salient parts. Hence, to best handle such tweets, if there are more than one "but"s in the same sentence of a tweet, only the part posterior to the last occurrence of the word "but" is to be considered as the most salient.

Hence, we propose the following strategy to modify n-grams for tweets containing the conjunction "but":

1. We use the Punkt sentence tokeniser [28] to break a tweet into sentences.
2. In each sentence, we find the location of the last occurrence of the word “but”
3. We remove all tokens except the tokens posterior to (occurring after) that location. So, the modified sentence only contains tokens succeeding the last “but”.
4. Once we have processed all the sentences in the tweet, we merge the modified sentences together to obtain the modified tweet.

Moving forwards, below are a few examples of tweets containing “if” at different syntactic positions. In each tweet, the most salient part that is the part that contributes considerably to the overall polarity of the tweet is underlined. In certain tweets however, if no salient part can be found or is ambiguous, nothing is underlined. The overall polarity of the tweet is indicated in parenthesis.

(1) If Gerald Green doesn’t have the most hops in the league then he definitely is a strong 2nd!! *(positive)*

(2) If you’re not coming to the SVSU vs. Wayne State game tomorrow, watch it on CBS College Sports or FSN Detroit. It’s about to be hype! *(positive)*

(3) If the Lakers still had Jordan Farmar,Trevor Ariza,&Shannon Brown I’d be watching them ..I dont like the Lakers but they were entertaining. *(positive)*

(4) if you follow @USER ill make you my famous Oreo brownie on Sunday!!! *(positive)*

(5) Juniors playing powderpuff, if you aren’t at practice tomorrow you will NOT play, it starts at 5:30pm, hope to see you there! *(negative)*

(6) @USER can you please come to SXSW in Austin in March? I’ve wanted to see you for years & it would be amazing if you played a show here! *(positive)*

From the above examples, we can see that as compared to “but”, “if” has many more syntactic positions, such as:

(i) if <condition clause> then <consequent clause>
(ii) if <condition clause>, <consequent clause>
(iii) if <condition clause> <missing then/comma, or other> <consequent clause>
(iv) <consequent clause> if <condition clause>

According to syntax, example (1) is of type (i), example (2), (3) and (5) are of type (ii), example (4) is of type (iii), and example (6) is of type (iv). In examples (1) and (2), the most salient part of the tweet is the part that occurs after “then” or after the comma (,). Even in example (3), the part just after the first comma succeeding the “if” includes the most salient part of the tweet. However, example (3) contains both “if” and “but”, which makes it harder to automatically determine the most salient part.

Moreover, in examples (4) and (5), the most salient part is not preceded by a “then” or comma, due to grammatical errors introduced by the writer or due to the informal nature of tweets. In example (6), "if" occurs in the middle of the sentence, such that even though the consequent clause usually precedes the "if" in such cases, it is hard to automatically determine the scope of the most salient part. Hence, determining the most salient part in tweets similar to (4), (5), and (6) requires more complex linguistic analysis which is beyond the scope of this study.

In this paper, we will only focus on tweets similar to (1), (2), and (3). Also, while the examples above are only limited to the conditional "if", we will also handle the conditionals "unless", "until" and "in case". For these conditionals, we consider the most salient part of the tweet to be the part that occurs after the first comma succeeding the conditional, whereas for "if" we consider the part occurring after "then" as well as the comma.

Therefore, we propose the following strategy to modify n-grams for tweets containing the conditionals "if", "unless", "until" and "in case":

1. We use the Punkt sentence tokeniser [28] to break a tweet into sentences.
2. In each sentence, we find the location of the last occurrence of the conditional ("if", "unless", "until" or "in case")
3. Then, we find the location of the first comma (and also "then" in case of "if") that occurs after the conditional.
4. We remove all tokens between the conditional and the comma/"then" including the conditional and the comma/"then". All the remaining tokens now make up the modified sentence.
5. Once we have processed all the sentences in the tweet, we merge the modified sentences together to obtain the modified tweet.

In case a tweet contains a conditional as well as the conjunction "but", only "but" rules are applied.

Finally, using the modified tweets, we create new feature vectors containing modified unigrams, bigrams, and trigrams for each tweet. These modified n-grams are then provided as input to the Support Vector Machine (SVM) specified in section 4.2, instead of the n-grams that are typically used.

4.4 Tweaking SVM Predictions Using Linguistic Rules and Sentic Computing

During training, a Support Vector Machine (SVM) approximates a hyperplane or decision boundary that best separates data points (feature vectors of samples) belonging to n different classes (feature vectors of samples = n-grams of tweets, n = 2 and class $\in$ {positive, negative} in our case). The data points that "support" this hyperplane on either sides are known as support vectors.

Each trained SVM has a scoring function that computes the decision score for each new sample, based on which the class label is assigned. The SVM decision score for classifying a sample is the signed distance from the sample's feature vector x to the decision boundary, and is given by:

$$SVM\ Decision\ Score\ =\ \sum_{m}^{i=1} \alpha_i y_i G(x_i, x) + b \tag{1}$$

where $\alpha_1, \alpha_2, ...\alpha_n$, and b are the parameters estimated by the SVM, $G(x_i, x)$ is the dot product in the predictor space between x and the support vectors, and m is the number of training samples.

As explained in section 2, the decision score for a large number of tweets is too low, implying that the SVM is unsure about the label it assigns to them, because their feature vector lies very close to the decision boundary. Hence, after running the supervised classifier on all the unlabelled tweets, we get the decision score computed by it for each tweet to determine the confidence of the SVM's predictions.

For tweets with an absolute decision score or confidence below 0.5, we discard the class labels assigned by the SVM and instead use an unsupervised classifier to predict their class labels. This unsupervised classification process works as follows:

1. The tweets are modified using the method describes in section 4.3, in order to take into account conjunctions and conditionals.
2. Single-word and multi-word concepts are extracted from the tweets in order to fetch their polarities from SenticNet [14]. These concepts are extracted using algorithm 1.

Algorithm 1. Given a list of *tokens* in a tweet and a list of their corresponding POS tags, this algorithm extracts a bag-of-concepts from tweets

```
token1 = []; pos1 = [];
{First, remove all stop words from the tweet tokens}
for each token, tag in tokens, pos do
    if token is NOT a stop word then
        append token to token1 and tag to pos1
    end if
end for
concepts = []
{adjacent tokens with the following POS tags2 are extracted as multi-word concepts}
conceptTagPairs = [("N", "N"), ("N", "V"), ("V", "N"), ("A", "N"), ("R", "N"), ("P", "N"),
("P", "V")]
for ti in range(0, len(tokens1)) do
    token = tokens1[ti]; tag = pos1[ti];
    prevtoken = tokens1[ti-1]; prevtag = pos1[ti-1];
    token_stem = Stem(token); prevtoken_stem = Stem(prevtoken);
    {raw tokens and stemmed tokens are extracted as single-word concepts}
    append token to concepts
    append token_stem to concepts
    if (prevtag, tag) in conceptTagPairs then
        append prevtoken+" "+token to concepts
        append prevtoken_stem+" "+token_stem to concepts
    end if
end for
```

[2] "N" = Noun, "V" = Verb, "A" = Adjective, "R" = Adverb, "P" = Preposition.

3. Then, we query all these concepts in SenticNet in order to get their polarities. If a single-word concept is not found in SenticNet, it is queried in SentiWordNet [29], and if it is not found in SentiWordNet, it is searched for in the list of positive and negative words from the Bing Liu lexicon [30]. The number of positive and negative concepts, and the polarity of the most polar concept is noted as the tweet's most polar value. The Bing Liu lexicon only contains a list of around 2000 strongly positive and 4800 strongly negative words, and no polarity values. So, the polarity of all positive words in the Bing Liu lexicon is assumed as $+1.0$ while the polarity of all negative words is assumed as -1.0.
4. Based on the number of positive and negative concepts, and the most polar value occurring in the tweet, the following rules are applied to classify it:
   – If the number of positive concepts is greater than the number of negative concepts and the most polar value occurring in the tweet is greater than or equal to 0.6, the tweet is labelled as positive.
   – If the number of negative concepts is greater than the number of positive concepts and the most polar value occurring in the tweet is less than or equal to –0.6, the tweet is labelled as negative.
   – If neither of the two rules stated above apply, the tweet is labeled as unknown by the rule-based classifier, and the SVM's low confidence predictions are taken as the final output of the system.

5 Experiments and Results

We train our SVM [27] classifier on around 1.6 million positive and negative tweets provided by [8]. First, the training data is divided into 80% train and 20% validation sets, and the "c" parameter is selected as 0.4 through 10-fold cross-validation. Then, the model is trained on 100% of the training data.

Table 2. Results obtained on 1794 positive/negative tweets from the SemEval 2013 dataset

Method	Positive			Negative			Average		
	P	R	F	P	R	F	P	R	F
N-grams	90.48	82.67	86.40	61.98	76.45	68.46	76.23	79.56	77.43
N-grams and Emoticon Rules	90.62	83.36	86.84	62.99	76.65	69.15	76.80	80.00	78.00
Modified N-grams	89.95	84.05	86.90	63.33	74.59	68.50	76.64	79.32	77.70
Modified N-grams, and Emoticon Rules	90.10	84.73	87.33	64.41	74.79	69.22	77.26	79.76	78.27
Modified N-grams, Emoticon Rules, and Word-level Unsupervised Rules	91.40	86.79	89.04	68.55	77.89	72.92	79.97	82.34	80.98
Modified N-grams, Emoticon Rules, and Concept-level Unsupervised Rules	92.42	86.56	89.40	68.96	80.79	74.41	80.69	83.68	81.90

We evaluate our proposed method on two publicly available datasets – SemEval 2013 [12] test set and SemEval 2014 [12] test set. Neutral tweets are removed from each dataset, which leaves 1794 and 3584 positive/negative tweets in the SemEval 2013 and SemEval 2014 datasets respectively. Tables 2 and 3 show the results obtained on these two datasets. In these tables, each row shows the precision (P), recall (R), and F-score for the positive, and negative classes, followed by the average positive and negative precision, recall, and F-score. All values in the tables are between 0 and 100, and are rounded off to 2 decimal places. This section will focus on discussing and analysing the results shown.

In order to gauge the effectiveness of our method, we consider averaged positive and negative F-score (F_{avg}) as the primary evaluation metric, and the standard n-grams based supervised model as a benchmark. It is important to note that apart from TF-IDF weighed frequency counts of n-grams, this standard n-grams benchmark model also takes negation into account.

Table 3. Results obtained on 3584 positive/negative tweets from the SemEval 2014 dataset

Method	Positive			Negative			Average		
	P	R	F	P	R	F	P	R	F
N-grams	89.92	81.90	85.72	61.20	75.66	67.67	75.56	78.78	76.69
N-grams and Emoticon Rules	89.74	83.05	86.27	62.50	74.85	68.11	76.12	78.95	77.19
Modified N-grams	89.39	82.90	86.02	62.00	73.93	67.44	75.69	78.41	76.73
Modified N-grams, and Emoticon Rules	89.25	83.97	86.53	63.29	73.22	67.89	76.27	78.60	77.21
Modified N-grams, Emoticon Rules, and Word-level Unsupervised Rules	90.22	86.24	88.19	67.37	75.25	71.09	78.80	80.75	79.64
Modified N-grams, Emoticon Rules, and Concept-level Unsupervised Rules	90.41	86.20	88.25	67.45	75.76	71.37	78.93	80.98	79.81

On comparing the standard n-grams model with the n-grams and emoticon rules model, we can see that emoticon rules increase F_{avg} by 0.57 and 0.50 in the 2013 and 2014 datasets respectively. Comparison between the modified n-grams model, and modified n-grams and emoticon rules model also shows that emoticon rules increase F_{avg} by 0.57 and 0.48 in the two datasets respectively. Thus, this shows that the emoticon rules formulated by us significantly improve sentiment analysis.

Modifying n-grams using linguistic rules for conjunctions and conditionals increases F_{avg} by 0.27 and 0.04 in the two datasets respectively. While the increase is not very significant for the 2014 dataset, modified n-grams are still better than standard n-grams as (i) they do increase the overall F_{avg} and the increase is quite significant in the 2013 dataset, (ii) a typical Twitter corpus contains a very small percentage of tweets with such conjunctions and conditionals, and hence even a small improvement is very encouraging.

Next, we observe the results obtained by tweaking the SVM's predictions using the method specified in section 4.4. In this, we also compare the results obtained by using a bag-of-concepts model to the results obtained by using a bag-of-words (or single-word concepts only) model. We see that the F_{avg} of the bag-of-concepts model is 0.92 more than the bag-of-words model for the 2013 dataset, and 0.17 more than the bag-of-words model for the 2014 dataset. So, even though the effect of moving to concept-level sentiment analysis from word-level sentiment analysis will vary from one dataset to another, concept-level sentiment features will almost always perform better since they already include word-level sentiment features.

On comparing the results obtained by the modified n-grams and emoticon rules model with the modified n-grams, emoticon rules and concept-level unsupervised rules model, we see that tweaking the SVM's predictions using rules and sentic computing increases the F_{avg} by 3.63 and 2.6 in the two datasets respectively. Hence, this shows that the linguistic rules and sentic computing based secondary classifier proposed by us, substantially improve the result and is thus very beneficial for sentiment analysis.

Overall, our final sentiment analysis system achieves a F_{avg} score that is 4.47 units and 3.12 units higher than the standard n-grams model.

6 Conclusion and Future Work

In this paper, we describe the pipeline of a Twitter sentiment analysis system that enhances supervised learning, by using modified features for supervised learning as well as applying rules based on linguistics and sentic computing. Based on our results, we can conclude that unsupervised emoticon rules and modified n-grams for supervised learning help improve sentiment analysis. They do so by handling peculiar linguistic characteristics introduced by special parts-of-speech such as emoticons, conjunctions and conditionals. Moreover, we have shown that verifying or changing the low-confidence predictions of a supervised classifier using a secondary rule-based (high-confidence, unsupervised) classifier is also immensely beneficial.

In the future, we plan to further improve performance of our classifier [31]. We will do this by further analysing the linguistics of tweets to take into account other conjunctions such as "or", conditionals such as "assuming", or modal verbs such as "can", "could" , "should", "will" and "would". We also plan to develop more sophisticated rules to improve the classification of tweets that the supervised classifier assigns a low decision score to. Apart from deeper linguistic analysis and better rules, expanding common-sense knowledge bases such as SenticNet [14] and the use of concept based text analysis [32] can also help to boost the predictions of the unsupervised classifier, thereby improving the predictions of the whole system. The proposed approach can also be fed to a multimodal sentiment analysis framework [33][34]. Future work will also explore the use of common-sense vector space resources such as [35,36], construction of new ones [37], and extraction of aspects from the tweets [38], as well as richer n-gram [39], vector space [40], or graph-based [41] text representations.

Acknowledgements. We wish to acknowledge the funding for this project from Nanyang Technological University under the Undergraduate Research Experience on

CAmpus (URECA) programme. The work was partially supported by Mexican Government (CONACYT grant 122030, SIP-IPN grant 20150028, COFAA-IPN) and European project FP7-PEOPLE-2010-IRSES grant 269180.

References

1. Li, H., Liu, B., Mukherjee, A., Shao, J.: Spotting fake reviews using positive-unlabeled learning. Computación y Sistemas 18, 467–475 (2014)
2. Alonso-Rorís, V.M., Santos Gago, J.M., Pérez Rodríguez, R., Rivas Costa, C., Gómez Carballa, M.A., Anido Rifón, L.: Information extraction in semantic, highly-structured, and semi-structured web sources. Polibits 49, 69–75 (2014)
3. Cambria, E., Schuller, B., Xia, Y., Havasi, C.: New avenues in opinion mining and sentiment analysis. IEEE Intelligent Systems 28, 15–21 (2013)
4. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2, 1–135 (2008)
5. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies 5, 1–167 (2012)
6. Kouloumpis, E., Wilson, T., Moore, J.: Twitter sentiment analysis: The good the bad and the omg? In: ICWSM 2011, pp. 538–541 (2011)
7. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREC, vol. 10, pp. 1320–1326 (2010)
8. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, pp. 1–12 (2009)
9. Poria, S., Cambria, E., Winterstein, G., Huang, G.B.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowledge-Based Systems 69, 45–63 (2014)
10. Liu, Y., Yu, X., Liu, B., Chen, Z.: Sentence-level sentiment analysis in the presence of modalities. In: Gelbukh, A. (ed.) CICLing 2014, Part II. LNCS, vol. 8404, pp. 1–16. Springer, Heidelberg (2014)
11. Narayanan, R., Liu, B., Choudhary, A.: Sentiment analysis of conditional sentences. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 1, pp. 180–189. Association for Computational Linguistics (2009)
12. Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., Wilson, T.: SemEval-2013 task 2: Sentiment analysis in Twitter. In: Proceedings of the International Workshop on Semantic Evaluation, SemEval, vol. 13 (2013)
13. Datla, V.V., Lin, K.I., Louwerse, M.M.: Linguistic features predict the truthfulness of short political statements. International Journal of Computational Linguistics and Applications 5, 79–94 (2014)
14. Cambria, E., Olsher, D., Rajagopal, D.: SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In: Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1515–1521 (2014)
15. Speer, R., Havasi, C.: ConceptNet 5: A large semantic network for relational knowledge. In: The Peoples Web Meets NLP, pp. 161–176. Springer (2013)
16. Wiegand, M., Balahur, A., Roth, B., Klakow, D., Montoyo, A.: A survey on the role of negation in sentiment analysis. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, pp. 60–68. Association for Computational Linguistics (2010)
17. Councill, I.G., McDonald, R., Velikovich, L.: What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In: Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, pp. 51–59. Association for Computational Linguistics (2010)

18. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 241–249. Association for Computational Linguistics (2010)
19. Mohammad, S.M., Kiritchenko, S., Zhu, X.: NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In: Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEMSTAR 2013) (2013)
20. Chikersal, P., Poria, S., Cambria, E.: SeNTU: Sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In: Proceedings of the International Workshop on Semantic Evaluation (SemEval 2015) (2015)
21. Turney, P.D.: 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, pp. 417–424. Association for Computational Linguistics (2002)
22. Chaovalit, P., Zhou, L.: Movie review mining: A comparison between supervised and unsupervised classification approaches. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS 2005, pp. 112c–112c. IEEE (2005)
23. Cambria, E., Hussain, A.: Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis. Springer, Cham (2015)
24. Cambria, E., White, B.: Jumping NLP curves: A review of natural language processing research. IEEE Computational Intelligence Magazine 9, 48–57 (2014)
25. Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., Smith, N.A.: Improved part-of-speech tagging for online conversational text with word clusters. In: HLT-NAACL, pp. 380–390 (2013)
26. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24, 513–523 (1988)
27. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874 (2008)
28. Kiss, T., Strunk, J.: Unsupervised multilingual sentence boundary detection. Computational Linguistics 32, 485–525 (2006)
29. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC, vol. 6, pp. 417–422. Citeseer (2006)
30. Liu, B., Hu, M., Cheng, J.: Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web, pp. 342–351. ACM (2005)
31. Schnitzer, S., Schmidt, S., Rensing, C., Harriehausen-Mühlbauer, B.: Combining active and ensemble learning for efficient classification of web documents. Polibits 49, 39–45 (2014)
32. Agarwal, B., Poria, S., Mittal, N., Gelbukh, A., Hussain, A.: Concept-level sentiment analysis with dependency-based semantic parsing: A novel approach. Cognitive Computation, 1–13 (2015)
33. Poria, S., Cambria, E., Hussain, A., Huang, G.B.: Towards an intelligent framework for multimodal affective data analysis. Neural Networks 63, 104–116 (2015)
34. Poria, S., Cambria, E., Howard, N., Huang, G.B., Hussain, A.: Fusing audio, visual and textual clues for sentiment analysis from multimodal content. Neurocomputing (2015)
35. Poria, S., Gelbukh, A., Cambria, E., Hussain, A., Huang, G.B.: EmoSenticSpace: A novel framework for affective common-sense reasoning. Knowledge-Based Systems 69, 108–123 (2014)
36. Cambria, E., Fu, J., Bisio, F., Poria, S.: Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In: Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 508–514 (2015)
37. Vania, C., Ibrahim, M., Adriani, M.: Sentiment lexicon generation for an under-resourced language. International Journal of Computational Linguistics and Applications 5, 63–78 (2014)

38. Poria, S., Cambria, E., Ku, L.W., Gui, C., Gelbukh, A.: A Rule-Based Approach to Aspect Extraction from Product Reviews. In: Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP), pp. 28–37. Association for Computational Linguistics and Dublin City University (2014)
39. Sidorov, G.: Should syntactic n-grams contain names of syntactic relations? International Journal of Computational Linguistics and Applications 5, 139–158 (2014)
40. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D.: Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas 18, 491–504 (2014)
41. Das, N., Ghosh, S., Gonçalves, T., Quaresma, P.: Comparison of different graph distance metrics for semantic text based classification. Polibits 49, 51–57 (2014)