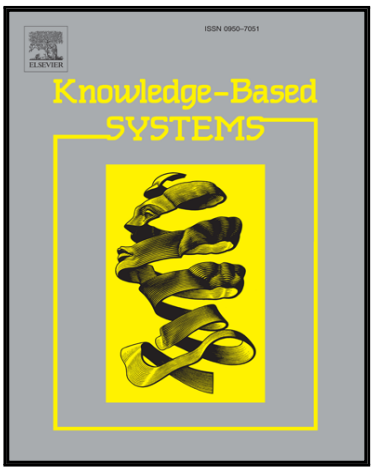

Building a Twitter Opinion Lexicon from Automatically-annotated Tweets

Felipe Bravo-Marquez, Eibe Frank, Bernhard Pfahringer

PII: S0950-7051(16)30106-X
DOI: 10.1016/j.knosys.2016.05.018
Reference: KNOSYS 3515

To appear in: *Knowledge-Based Systems*

Received date: 12 November 2015
Revised date: 9 May 2016
Accepted date: 9 May 2016

Please cite this article as: Felipe Bravo-Marquez, Eibe Frank, Bernhard Pfahringer, Building a Twitter Opinion Lexicon from Automatically-annotated Tweets, *Knowledge-Based Systems* (2016), doi: 10.1016/j.knosys.2016.05.018

Highlights

- We propose a supervised model for expanding an opinion lexicon for Twitter.
- We combine automatically annotated tweets with existing hand-made opinion lexicons.
- We use POS tags and associations between words and sentiment as word-level features.
- Expanded words are mapped to a positive, negative, and neutral distribution.
- We outperform the performance obtained by using semantic orientation alone.

Building a Twitter Opinion Lexicon from Automatically-annotated Tweets

Felipe Bravo-Marquez*, Eibe Frank, Bernhard Pfahringer

Department of Computer Science, The University of Waikato, Private Bag 3105, Hamilton 3240, New Zealand

Abstract

Opinion lexicons, which are lists of terms labelled by sentiment, are widely used resources to support automatic sentiment analysis of textual passages. However, existing resources of this type exhibit some limitations when applied to social media messages such as tweets (posts in Twitter), because they are unable to capture the diversity of informal expressions commonly found in this type of media.

In this article, we present a method that combines information from automatically annotated tweets and existing hand-made opinion lexicons to expand an opinion lexicon in a supervised fashion. The expanded lexicon contains part-of-speech (POS) disambiguated entries with a probability distribution for positive, negative, and neutral polarity classes, similarly to SentiWordNet.

To obtain this distribution using machine learning, we propose word-level attributes based on (a) the morphological information conveyed by POS tags and (b) associations between words and the sentiment expressed in the tweets that contain them. We consider tweets with both hard and soft sentiment labels. The sentiment associations are modelled in two different ways: using point-wise-mutual-information semantic orientation (PMI-SO), and using stochastic gradient descent semantic orientation (SGD-SO), which learns a linear relationship between words and sentiment. The training dataset is labelled by a seed lexicon formed by combining multiple hand-annotated lexicons.

Our experimental results show that our method outperforms the three-dimensional word-level polarity classification performance obtained by using PMI-SO alone. This is significant because PMI-SO is a state-of-the-art measure for establishing world-level sentiment. Additionally, we show that lexicons created with our method achieve significant improvements over SentiWordNet for classifying tweets into polarity classes, and also outperform SentiStrength in the majority of the experiments.

Keywords: Lexicon Expansion, Sentiment Analysis, Twitter

1. Introduction

Many sentiment analysis methods rely on opinion lexicons as resources for evaluating the sentiment of a text passage. An opinion or sentiment lexicon is a dictionary of opinion words with their corresponding sentiment categories or semantic orientations. A semantic orientation is a numerical measure for representing the polarity and strength of words or expressions. Lexicons can be used to compute the polarity of a message by aggregating the orientation values of the opinion words it contains [17, 35]. They have also proven to be useful when used to extract features in supervised classification schemes [8, 19, 22, 23, 47].

Social media platforms, particularly microblogging services such as **Twitter**[1], are increasingly being adopted by people to access and publish information about a great variety of topics. The language used in **Twitter** provides substantial challenges for sentiment analysis. The words used in this platform include many abbreviations, acronyms,

*Corresponding author

Email addresses: fjb11@students.waikato.ac.nz (Felipe Bravo-Marquez), eibe@waikato.ac.nz (Eibe Frank), bernhard@waikato.ac.nz (Bernhard Pfahringer)

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

and misspelled words that are not observed in traditional media or covered by popular lexicons, e.g., omg, loove, #screwthis. The diversity and sparseness of these informal words make the manual creation of a Twitter-oriented opinion lexicon a time-consuming task.

In this article, we propose a method for opinion lexicon expansion for the language used in Twitter[2]. Taking SentiWordNet [2] as inspiration, each word in our expanded lexicon has a probability distribution, describing how positive, negative, and neutral it is. Additionally, all the entries of the lexicon are associated with a corresponding part-of-speech tag. Estimating the sentiment distribution of POS-tagged words is useful for the following reasons:

1. A word can present certain levels of intensity [37] for a specific sentiment category, e.g., the word *awesome* is more positive than the word *adequate*. The estimated probabilities can be used to represent these levels of intensity. These probabilities provide a probabilistic interpretation of the underlying sentiment intensities conveyed by a word and can be used as prior knowledge in Bayesian models for sentiment inference [24]. In contrast, scores obtained by unsupervised methods such as point-wise-mutual information semantic orientation (PMI-SO) [39] lack a probabilistic interpretation.
2. The neutral score provided by the lexicon is useful for discarding non-opinion words in text-level polarity classification tasks. This can easily be done by discarding words classified as neutral. Note that unsupervised lexicon expansion techniques such as PMI-SO [39] provide a single numerical score for each word, and it is unclear how to impose thresholds on this score for neutrality detection.
3. Homographs, which are words that share the same spelling but have different meanings, should have different lexicon entries for each different meaning. By using POS-tagged words, homographs with different POS-tags will be disambiguated [42]. For instance, the word *fine* will receive different sentiment scores when used as an adjective (e.g., *I'm **fine** thanks*) and as a common noun (e.g., *I got a parking **fine** because I displayed the ticket upside down*).

This is not the first work exploring these properties for lexicon expansion. Sentiment intensities were described with probabilities in [2], and the disambiguation of the sentiment of words based on POS tags was studied in [35]. However, this is the first time that these properties are explored for the informal language used in Twitter.

Our expanded lexicon is built by training a word-level sentiment classifier for the words occurring in a corpus of positive and negative polarity-annotated tweets. The training words are labelled using a seed lexicon of positive, negative, and neutral words. This lexicon is taken from the union of four different hand-made lexicons after discarding all polarity clashes from the intersection. The expanded words are obtained after deploying the trained classifier on the remaining unlabelled words from the corpus of tweets that are not included in the seed lexicon.

All the words from the polarity-annotated corpus of tweets are represented by features that capture morphological and sentiment information of the word in its context. The morphological information is captured by including the POS tag of the word as a nominal attribute, and the sentiment information is captured by calculating association values between the word and the polarity labels of the tweets in which it occurs.

We calculate two types of word-level sentiment associations: PMI-SO [39], which is based on the point-wise mutual information (PMI) between a word and tweet-level polarity classes, and stochastic gradient descent semantic orientation (SGD-SO), which is based on incrementally learning a linear association between words and the sentiment of the tweets in which they occur.

To avoid the high costs of manually annotating tweets into polarity classes for calculating the word-level sentiment associations, we rely on two heuristics for automatically obtaining polarity-annotated tweets: **emoticon-based annotation** and **model transfer**. In the first approach, only tweets with positive or negative emoticons are considered and labelled according to the polarity indicated by the emoticon. This idea, which has been widely used before to train message-level sentiment classifiers [5, 16] is affected by two main limitations:

1. The removal of tweets without emoticons may cause a loss of valuable words that do not co-occur with emoticons.
2. There are many domains, such as politics, in which emoticons are not frequently used to express positive and negative opinions. Thus, it is very difficult to obtain emoticon-annotated data from these domains.

[2]This article extends a previous conference paper [7] and provides a more thorough and detailed report.

To overcome these limitations, we pursue a model transfer approach by training a probabilistic message-level classifier from a corpus of emoticon-annotated tweets and using it to label a target corpus of unlabelled tweets with a probability distribution of positive and negative sentiment. Note that the model transfer produces soft sentiment labels, in contrast to the hard labels provided by the emoticons. We study how to compute our word-level sentiment association attributes from tweets annotated with both hard and soft labels.

We test our word-level sentiment classification approach on words obtained from different collections of automatically labelled tweets. The results indicate that our supervised framework outperforms using PMI-SO by itself when the detection of neutral words is considered. We also evaluate the usefulness of the expanded lexicon for classifying entire tweets to polarity classes, showing significant improvement in performance compared to the original lexicon.

This article is organised as follows. In Section 2, we provide a review of existing work on opinion lexicon expansion. In Section 3, we describe the proposed method in detail. In Section 4, we present the experiments we conducted to evaluate the proposed approach and discuss results. The main findings and conclusions are discussed in Section 5.

2. Related Work on Lexicon Expansion

There are two types of resources that can be exploited for automatically building or expanding opinion lexicons: semantic networks, and document collections. Previous work on opinion lexicon expansion from these two type of resources is presented in the following two subsections.

2.1. Semantic Networks

A semantic network is a network that represents semantic relations between concepts. The simplest approach, based on a semantic network of words such as WordNet[3], is to expand a seed lexicon of labelled opinion words using synonyms and antonyms from the lexical relations [18, 21]. The hypothesis behind this approach is that synonyms have the same polarity and antonyms have the opposite. This process is normally iterated several times. In [20], a graph is created using WordNet adjectives as vertices and the synonym relations as edges. The orientation of a term is determined by its relative distance from the two seed terms *good* and *bad*. In [12], a supervised classifier is trained using a seed of labelled words that is obtained through expansion based on synonyms and antonyms. For each word, a vector space model is created from the definition or *gloss* provided by the WordNet dictionary. This representation is used to train a word-level classifier that is used for lexicon expansion. An equivalent approach was applied later to create SentiWordNet[4] [2, 13]. In SentiWordNet, each WordNet *synset* or group of synonyms is assigned into classes *positive*, *negative* and *neutral*, with soft labels in the range [0, 1].

Another well-known lexical resource for sentiment analysis built from concept-level semantic networks is SenticNet[5], which labels multi-word concepts according to both affective and semantic information. SenticNet is based on the *sentic computing* paradigm, which focuses on a semantic-preserving representation of natural language concepts and on sentence structure [11]. Multiple techniques have been exploited along the different versions of SenticNet. The first two versions were built using graph-mining and dimensionality-reduction techniques, and the third version integrates multiple knowledge sources by setting up pathways between concepts.

The automatic processing of emotions is the main focus of the field of *Affective Computing*, which is closely related to sentiment analysis [10]. WordNet-Affect[6] is a semantic network for affective computing in which some WordNet synsets are mapped into affective states corresponding to emotion and mood [40]. WordNet-Affect was used together with SenticNet for building *EmoSenticSpace* [34], a knowledge-base of natural language concepts annotated according to emotion labels and polarity scores. This resource was built using fuzzy c-means clustering and support vector machine (SVM) classification.

ConceptNet[7] is a semantic network of commonsense knowledge formed by over 1.6 million assertions composed of two concepts connected by a relation e.g., *car usedFor driving*. There are 33 different types of relations such as

[3] http://wordnet.princeton.edu/
[4] http://sentiwordnet.isti.cnr.it/
[5] http://sentic.net/
[6] http://wndomains.fbk.eu/wnaffect.html
[7] http://conceptnet5.media.mit.edu/

IsA, PartOf and UsedFor. Lexicon expansion methods based on this resource were proposed in [38, 41, 44]. In [38], each concept on ConceptNet is given a sentiment score using iterative regressions that are then propagated via random walks. However, considering that not all relations from ConceptNet are necessarily related to sentiment, the model was further improved in [44] using sequential forward search to find the best combination of sentiment-associated relations from ConceptNet. The model performs a bias correction step after the random walk process to reduce the variability in the obtained polarities.

A drawback of opinion lexicons is their lack of contextual information. A method for contextualising and interpreting ambiguous sentiment terms in opinion lexicons is proposed in [41]. The method performs three steps to add positive and negative context terms to extend the expressiveness of the target resource: 1) identify ambiguous sentiment terms from SenticNet, 2) extract context information from a domain-specific corpus, and 3) associate the extracted context information with knowledge sources such as ConceptNet and WordNet.

Lexicons built from semantic networks are unable to capture sentiment information from words or concepts that go beyond the exploited network. Because words or concepts included in semantic networks such as WordNet and ConceptNet are based on formal English rather than informal expressions, the resources expanded from these networks will exhibit limitations when used with Twitter.

2.2. Corpus-based approaches

Corpus-based approaches exploit syntactic or co-occurrence patterns to expand a lexicon to the words found within a collection of unstructured textual documents. In [39], the expansion is done through a measure referred to as *PMI semantic orientation* (PMI-SO), which is based on the point-wise mutual information (PMI) between two random variables:

$$\mathrm{PMI}(term_1, term_2) = \log_2\left(\frac{Pr(term_1 \wedge term_2)}{Pr(term_1)Pr(term_2)}\right) \tag{1}$$

The PMI semantic orientation of a word is the difference between the PMI of the word with a positive and a negative sentiment. Different ways have been proposed to represent the joint probabilities of words and sentiment. In Turney's work [39], they are estimated using the number of hits returned by a search engine in response to a query composed of the target word together with the word "excellent" and another query using the word "poor".

The same idea was used for Twitter lexicon expansion in [4, 22, 28, 46], which all model the joint probabilities of words and sentiment from sentiment-annotated collections of tweets. In [4], the tweets are labelled with a classifier trained from manually-annotated tweets using thresholds for the different classes to ensure high precision. In [46], the tweets are labelled with emoticons to create domain-specific lexicons. In [22, 28], the tweets are labelled with emoticons and hashtags associated with emotions to create two different lexicons. These lexicons are tested for tweet-level polarity classification.

Bahrainnian et al. [3] proposed another corpus-based model by relying on an unsupervised message-level sentiment classifier and the Twitter API. The message-level classifier is based on a seed lexicon and opinion rules for handling intensifiers, diminishers, and negations. For each target word to be included in the lexicon, a set of tweets containing the word is retrieved by sending it to the API. Then, the word is classified by averaging the predicted sentiment obtained by the message-level classifier for the retrieved tweets.

In [1, 6, 36], the expansion is conducted by representing Twitter words from a corpus as vectors that are classified into sentiment classes using machine learning. The word labels are provided by a seed lexicon. The word vectors used in [6] correspond to the centroid of the bag-of-word vectors of the tweets in which the words occur. Word embeddings, which are low-dimensional continuous dense word vectors trained from document corpora, were used in [1] and [36]. In [1], state-of-the-art word embeddings were used as features in a regression model for determining the association between Twitter words and positive sentiment. In [36], a hybrid loss function for learning sentiment-specific word embeddings is proposed. The embeddings are obtained by combining syntactic information provided the skip-gram model [26] and sentiment information provided by emoticon-annotated tweets.

As we can see from the above lexicon expansion models for Twitter, there are three kinds of information sources that can be used for building a Twitter-specific opinion lexicon: 1) sentiment-annotated tweets (manually or automatically annotated), 2) unlabelled tweets, and 3) hand-annotated seed lexicons. Our proposed approach is the first lexicon expansion model that uses all three of them. This enables the model to incorporate prior knowledge from both existing seed lexicons and sentiment-annotated tweets and learn the sentiment of words occurring in tweets that are not necessarily annotated according to sentiment labels.

3. Proposed Method

In this section we describe the proposed method for opinion lexicon expansion from automatically annotated tweets. The proposed process is illustrated in Figure 1, and can be summarised in the following steps:

1. Collect tweets from the target domain and the time period for which the lexicon needs to be expanded.
2. If the target collection has a significant number of positive and negative emoticons, label it using the emoticon-based annotation approach. Otherwise, collect tweets with positive and negative emoticons from a general domain and use it to label the target collection with the model transfer approach discusses below.
3. Tag all the words from the target collection using a part-of-speech tagger.
4. Calculate word-level features for all tagged words.
5. Label these words with a sentiment that match an existing hand-made polarity lexicon.
6. Train a word-level classifier using the word-level features and the words labels from the seed lexicon.
7. Use the trained classifier to estimate the polarity distribution of the remaining unlabelled words.

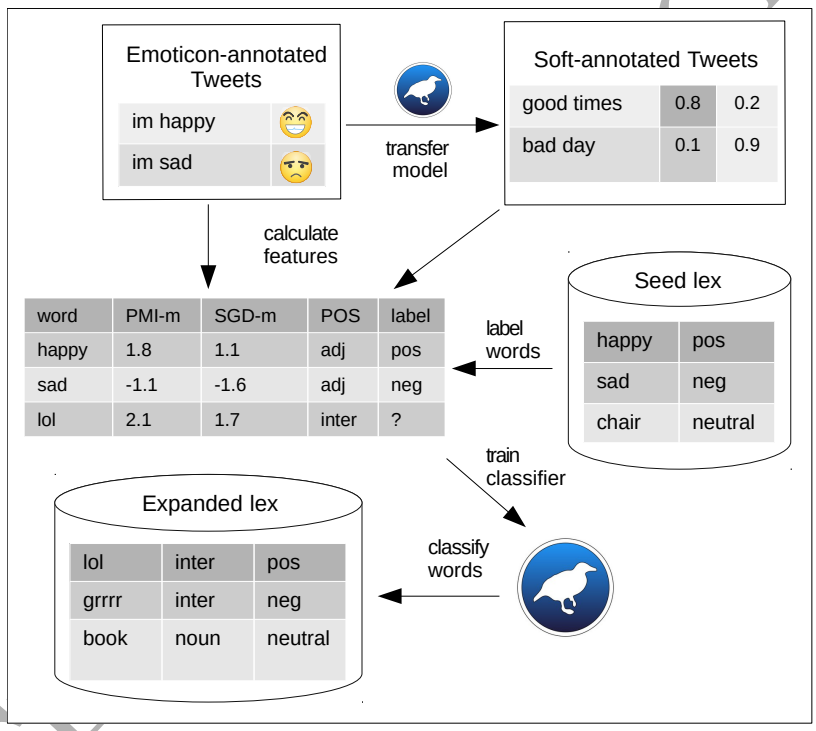

Figure 1: Twitter-lexicon expansion process. The bird represents the Weka machine learning software.

The key parts of the methodology are described in the following subsections. The mechanisms studied to automatically create collections of labelled tweets are described in Section 3.1. The proposed word-level attributes are described in Section 3.2. The seed lexicon used to label the training words is described in Section 3.3.

3.1. Automatically-annotated Tweets

The proposed method requires a collection of tweets with two particular properties: 1) the tweets must be labelled according to positive and negative polarity classes, and 2) they must be sorted in chronological order.

The first property is necessary for calculating our word-level features based on associations between words and the polarity of the tweets in which they occur (Section 3.2).

The second property is also necessary for calculating our word-level features, because they exploit how the associations between words and tweet-level polarities evolve over time.

The limitation of depending on polarity-annotated tweets is that the process of annotating tweets into polarity classes is labor-intensive and time-consuming. We tackle this problem by employing two automatic heuristics for data annotation: emoticon-based annotation and model transfer annotation.

3.1.1. Emoticon-based Annotation

In the emoticon-based annotation approach, tweets exhibiting positive :) and negative :(emoticons are labelled according to the emoticon's polarity [16]. Afterwards, the emoticon used to label the tweet is removed from the content. The emoticon-based labels are denoted with the letter y, and are assumed to be in $\{+1, -1\}$, corresponding to positively and negatively labelled tweets, respectively.

In the same way as in [16], the attribute space is reduced by replacing sequences of letters occurring more than two times in a row with two occurrences of them (e.g., huuungry is reduced to huungry, loooove to loove), and replacing user mentions and URLs with the generic tokens "USER" and "URL", respectively.

We consider two collections of tweets covering multiple topics for building our datasets: The Stanford Sentiment corpus (STS) [16], and The Edinburgh corpus (ED) [33]. These collections were gathered from two Twitter APIs: the search API[8], which allows the submission of queries composed of key terms, and the streaming API[9], from which a real-time sample of public posts can be retrieved.

The STS corpus is an emoticon-annotated collection created by periodically sending queries :) and :(to the Twitter search API between April 6th 2009 to June 25th 2009. The ED corpus is a general purpose collection of 97 million unlabelled tweets in multiple languages. It was collected with the Twitter streaming API between November 11th 2009 and February 1st 2010. We applied the emoticon-based annotation approach to the tweets written in English from this collection. We refer to this emoticon-annotated collection as ED.EM. The number of tweets for each polarity class in the two emoticon-annotated corpora is given in Table 1. We can observe that when using the streaming API (ED), positive emoticons occur much more frequently than negative ones.

	ED.EM	STS
Positive	1,813,705	800,000
Negative	324,917	800,000
Total	2,138,622	1,600,000

Table 1: Emoticon-annotated datasets.

As was discussed in Section 1, the shortcoming of the emoticon-based approach is that it discards a large amount of potentially valuable information and is useless in domains where emoticons are infrequently used to express sentiment.

3.1.2. Model Transfer Annotation

The model transfer approach enables the extraction of opinion words from any collection of tweets. It tackles the problems of the emoticon-based approach by relying on a self-training framework. The idea is to train a probabilistic message-level classifier from a source corpus C_s of emoticon-annotated tweets and then use it to classify a target-corpus of unlabelled data C_t. We use an L_2-regularised logistic regression model with unigrams as attributes for training the classifier and labelling the target collection with soft labels. The soft labels of a tweet $d \in C_t$ are denoted as $pos(d)$ and $neg(d)$, and represent a probability distribution of positive and negative sentiment (i.e., $1 - pos(d) = neg(d)$).

An important difference between the model transfer and emoticon-based approach is the nature of the sentiment labels they produce. The emoticon-based annotation approach produces hard labels $y \in \{+1, -1\}$; the model transfer produces soft ones $pos(d), neg(d) \in [0, 1]$.

The soft labels can easily be converted into hard ones by impossing a threshold λ and discarding tweets for which the classifier is not confident enough in its prediction:

[8] https://dev.twitter.com/rest/public/search
[9] https://dev.twitter.com/streaming/overview

$$y(d) = \begin{cases} 1 & \text{if } pos(d) \geq \lambda \\ -1 & \text{if } neg(d) \geq \lambda \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The tweets for which $y(d) = 0$ are then discarded. It is worth pointing out that the removal of these tweets may lead to the loss of valuable data, and that λ is a tuning parameter that needs to be adjusted with possible values between 0.5 and 1. An alternative approach is to use the soft labels directly. We will study strategies for extracting word-level attributes for both hard and soft labels in Section 3.2.

It is noteworthy to mention that in contrast to the emoticon-annotated source corpus, which is intentionally biased towards positive and negative tweets, the target collection may contain a substantial amount of neutral data or even tweets with mixed positive and negative sentiment. It is unclear how a classifier trained to discriminate between positive and negative tweets will behave when deployed on tweets that have a different sentiment class such as neutral or mixed. This might be a shortcoming of the model transfer approach. However, it is plausible that neutral tweets or tweets with mixed sentiment will be located close to the decision boundary of the classifier trained from positive and negative emoticons. Therefore, we can expect that the soft labels obtained with logistic regression for these types of tweets will have similar probabilities for both positive and negative classes and will be discarded when setting a high value of λ.

The data we use for our model transfer experiments is obtained using the STS corpus as the source collection, and a sample of 10 million tweets from ED as the target collection. The classifier we use is an L_2-regularised logistic regression model with the regularisation parameter C set to 1, generated using LIBLINEAR[10]. We refer to this corpus of tweets annotated with soft labels as ED.SL. The average values for the positive and negative soft labels in ED.SL are 0.64 and 0.36 respectively. We also convert this soft-annotated corpus into multiple hard-annotated datasets using different thresholds values. We refer to these collections as ED.Tα, where α is the value of the threshold. The number of positive and negative tweets in the resulting datasets is shown in Table 2. Note that the higher the value of α, the more tweets are discarded.

Dataset	ED.T06	ED.T07	ED.T08	ED.T09
Positive	6,279,007	5,164,387	3,761,683	2,030,418
Negative	2,182,249	1,609,195	1,090,086	586,441
Total	8,461,256	6,773,582	4,851,769	2,616,859

Table 2: Model transfer datasets with different threshold values.

3.2. Word-level Features

In this subsection, we provide a detailed description of the word-level features used for classifying words from a corpus of polarity-annotated tweets into positive, negative, and neutral classes.

We preprocess the given corpus before calculating the features. All the tweets are lowercased, tokenised and POS-tagged. We use the TweetNLP library [15], which provides a tokeniser and a tagger specifically for the language used in Twitter. We prepend a POS-tag prefix to each word in order to differentiate homographs exhibiting different POS-tags.

The first feature is a nominal attribute corresponding to the POS tag of the word in its context. This feature provides morphological information of the word. There is empirical evidence that subjective and objective texts have different distributions of POS tags [32]. According to [46], non-neutral words are more likely to exhibit the following POS tags in Twitter: noun, adjective, verb, adverb, abbreviation, emoticon and interjection. These findings suggest that POS tags may provide useful information for discriminating between neutral and non-neutral words.

The remaining features aim to capture the association between the POS-tagged word and sentiment. We sort the tweets from the collection chronologically and create two semantic orientation time series for each word: the SGD-SO

[10] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

series, and the PMI-SO series. These time series are designed to capture the evolution of the relationship between a word and the sentiment that it expresses. The way in which these series are calculated varies according to the nature of the sentiment labels of the tweets. As was described in Section 3.1, there are two types of message-level sentiment labels we can obtain from our methods for annotating tweets automatically: hard labels (positive or negative), and soft labels ($pos(d), neg(d)$). The hard labels are obtained when using the emoticon-annotation approach and the soft ones from the transfer model one. It is also possible to obtain hard-labels from the model transfer approach by applying a threshold parameter λ.

3.2.1. Features Calculated from Hard Labels

The first semantic orientation time series is calculated by incrementally training a linear model using stochastic gradient descent (SGD-SO). The weights of this model correspond to POS-tagged words and are updated in an incremental fashion. For the hard-labelled data ($y \in \{+1, -1\}$), we incrementally train a support vector machine [45] by optimising the hinge loss function with an L_2 penalty and a learning rate equal to 0.1:

$$\frac{\lambda}{2}\|w\|^2 + \sum [1 - y(\mathbf{x}\mathbf{w} + b)]_+. \tag{3}$$

The variables $\mathbf{w}$, b, and λ correspond to the weight vector, the bias, and the regularisation parameter, respectively. The regularisation parameter was set to 0.0001. The model's weights determine how strongly the presence of a word influences the prediction of negative and positive classes [5]. The SGD-SO time series is created by applying this learning process to a collection of labelled tweets and storing the word's coefficients in different time windows. We use time windows of $1,000$ examples.

The second time series corresponds to the accumulated PMI semantic orientation (PMI-SO), which is the difference between the PMI of the word with a positive sentiment and the PMI of the word with a negative sentiment [39]:

$$\begin{aligned} \text{PMI-SO}(w) &= \text{PMI}(w, \text{pos}) - \text{PMI}(w, \text{neg}) \\ &= \log_2\left(\frac{\Pr(w, pos)}{\Pr(w) \times \Pr(pos)}\right) - \log_2\left(\frac{\Pr(w, neg)}{\Pr(w) \times \Pr(neg)}\right) \\ &= \log_2\left(\frac{\Pr(w, pos) \times \Pr(neg)}{\Pr(pos) \times \Pr(w, neg)}\right) \end{aligned} \tag{4}$$

Let *count* be a function that counts the number of times that a word or a sentiment label has been observed until a certain time period. For hard-labelled tweets, we calculate the PMI-SO score for each POS-tagged word according to the following expression:

$$\text{PMI-SO}(w) = log_2\left(\frac{\text{count}(w \wedge y = 1) \times \text{count}(y = -1)}{\text{count}(y = 1) \times \text{count}(w \wedge y = -1)}\right) \tag{5}$$

We use time windows of $1,000$ examples and the Laplace correction to avoid the zero-frequency problem.

We use the time series to extract features that are used to train our world-level polarity classifier. These features summarise location and dispersion properties of the time series, and are listed in Table 3. The location-oriented features *mean*, *trunc.mean* and *median* measure the central tendency of the time series. The dispersion oriented features *sd*, *iqr*, *sg*, and *sg.diff* measure the variability of the time series. The feature *last.element* corresponds to the last value observed in the time series. This attribute would be equivalent to the traditional PMI semantic orientation measure for the PMI-SO time series calculated from hard labels.

3.2.2. Features Calculated from Soft Labels

In the scenario of tweets with soft labels, the PMI-SO and SGD-SO time series are calculated in a different way.

For the SGD-SO time series we use an L_2 regularised squared loss function. Let z be a real value that corresponds to the log odds of the positive and negative sentiment labels of a given tweet: $z = \log_2(\frac{pos(d)}{neg(d)})$, and let the variables $\mathbf{w}$,

Feature	Description
mean	The mean of the time series.
trunc.mean	The truncated mean of the time series.
median	The median of the time series.
last.element	The last observation of the time series.
sd	The standard deviation of the time series .
iqr	The inter-quartile range.
sg	The fraction of times the time series changes its sign.
sg.diff	The sg value applied to the differenced time series ($X_t - X_{t-1}$).

Table 3: Time series features.

b, and λ be analogous to the ones from the hinge loss. The squared loss function is defined as follows:

$$\frac{\lambda}{2}\|w\|^2 + \sum(z - (\mathbf{x}\mathbf{w} + b))^2. \tag{6}$$

The PMI-SO time series is calculated using soft counts. Let C be the set of tweets seen so far and $C(w)$ be the tweets from C in which the word w is observed. Then, the soft version of PMI-SO is calculated as follows:

$$\text{PMI-SO}'(w) = log_2\left(\frac{\sum_{d\in C(w)} \text{pos}(d) \times \sum_{d\in C} \text{neg}(d)}{\sum_{d\in C} \text{pos}(d) \times \sum_{d\in C(w)} \text{neg}(d)}\right) \tag{7}$$

We calculate the same features (Table 3) from the soft versions of the SGD-SO and PMI-SO time series as the ones calculated from their corresponding hard versions.

3.3. Ground-Truth Word Polarities

In this subsection, we describe the seed lexicon used to label the training words for our word sentiment classifier. In order to create an expanded lexicon similar to SentiWordNet, we require a seed lexicon of words manually labelled according to mutually exclusive positive, negative, and neutral sentiment classes. We create it by fusing the following manually created lexical resources:

- *MPQA Subjectivity Lexicon*: This lexicon was created by Wilson et al. [43] and is part of OpinionFinder[11], a system that automatically detects subjective sentences in document corpora. The lexicon has positive, negative, and a few neutral words.
- *Bing Liu*: This lexicon is maintained and distributed by Bing Liu[12] and is used in several of his papers [25]. It has positive and negative entries.
- *Afinn*: This strength-oriented lexicon [30] has positive words scored from 1 to 5 and negative words scored from -1 to -5. It includes slang, obscene words, acronyms and Web jargon. We tagged words with negative and positive scores to negative and positive classes respectively.
- *NRC emotion Lexicon*: This emotion-oriented lexicon [27] was created by conducting a tagging process on the crowdsourcing Amazon Mechanical Turk platform. In this lexicon, the words are annotated according to eight emotions: joy, trust, sadness, anger, surprise, fear, anticipation, and disgust, and two polarity classes: positive and negative. These categories are not mutually exclusive, and hence, a word can be tagged according to multiple emotions or polarities. Additionally, there are many words that are not associated with any emotion or polarity category. These are the ones we consider as neutral in this work. We consider positive, negative, and neutral words from this lexicon, and the words associated with both positive and negative categories are discarded.

[11] http://mpqa.cs.pitt.edu/opinionfinder/opinionfinder_2/
[12] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

As discussed in [8], different lexical resources may assign different categories to the same word. In order to create mutually exclusive polarity classes and to reduce the noise in our training data, we discard all words for which a polarity clash is observed. A polarity clash is a word that receives two or more different tags in the union of lexicons. The number of words for the different polarity classes in the different lexicons is displayed in Table 4.

	Positive	Negative	Neutral
AFINN	564	964	0
Bing Liu	2003	4782	0
MPQA	2295	4148	424
NRC-Emo	2312	3324	7714
Seed Lexicon	3730	6368	7088

Table 4: Lexicon Statistics.

The total number of clashes is 1074. Examples of words that are labelled as positive and negative by different lexicons are: audacious, excuse, futile, intense, and joke. This high number of clashes found among different hand-made lexicons indicates two things: 1) Different human annotators can disagree when tagging a word to polarity classes, and 2) there are words that can belong to more than one sentiment class. Hence, we can say that word-level polarity classification is a hard and subjective problem.

4. Evaluation

In this section, we conduct an experimental evaluation of the proposed model for Twitter opinion lexicon expansion. The evaluation is divided into four parts. In the first part we conduct an exploratory analysis of word-level features calculated from real Twitter data. In the second part we evaluate the word-level classifiers. In the third part we perform lexicon expansion using the trained classifiers and study the expanded resources. In the fourth part we conduct an extrinsic evaluation by using the expanded words for message-level polarity classification of tweets.

4.1. Exploratory Analysis

In this subsection, we explore the proposed time series described in Section 3.2 and the features extracted from them with the aim of observing how these variables correlate with the sentiment of words. The time series are calculated for the most frequent 10,000 POS-tagged words found in each of our two emoticon-annotated datasets (STS and ED.EM) using MOA[13], a data stream mining framework.

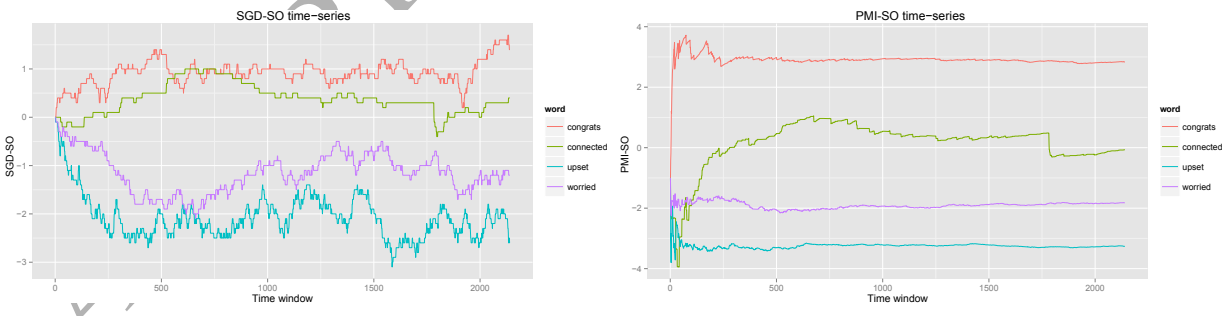

Figure 2: Word-level time series.

Figure 2 shows the resulting semantic orientation time series **SGD-SO** and **PMI-SO** for a sample of words in the ED.EM dataset. We can observe that the positive and neutral words *congrats* and *connected* exhibit greater PMI-SO

[13] http://moa.cs.waikato.ac.nz/

and SGD-SO scores than the negative words *worried* and *upset*. This suggests a correspondence between time series values and word polarities. We can also see that the PMI-SO time series are much more stable than the SGD-SO ones. We believe that PMI-SO is more stable than SGD-SO because, as shown in Equation 5, PMI-SO treats each word independently from all other words given the sentiment class. In contrast, SGD-SO scores are updated according to the SGD-SO learning rule, which comes from the sub-gradient of Equation 3. In this rule, the coefficients are updated every time the learning SVM misclassifies an example from the stream of emoticon-labelled tweets ($y(\mathbf{xw} + b) < 1$). Therefore, the change of the weight of a particular word depends both on the sentiment label and the co-occurring words within the tweets from the training collection.

To create training and test data for learning a word classifier, all POS-tagged words matching the seed lexicon, and, thus, their corresponding time series, are labelled according to the lexicon's polarities. It is interesting to consider how frequently positive, negative, and neutral words occur in a collection of tweets. The number of words labelled as positive, negative, and neutral for both the ED.EM and STS dataset is given in Table 5. As shown in the table, neutral words are the most frequent words in both datasets. Moreover, positive words are more frequent than negative ones.

	ED.EM	STS
Positive	1027	1023
Negative	806	985
Neutral	1814	1912
Total	3647	3920

Table 5: Word-level polarity classification datasets.

As the lexicon's entries are not POS-tagged, we assume that all possible POS tags of a word have the same polarity. However, this assumption can introduce noise in the training data. For example, the word *ill*, which is labelled as negative by the lexicon, will be labelled as negative for two different POS-tags: adjective, and nominal+verbal contraction. This word is very likely to express a negative sentiment when used as an adjective, but it is unlikely to express a negative sentiment when it refers, in a misspelled way, to the contraction *I'll*. A simple outlier removal technique to deal with this problem will be discussed in Section 4.3.

Once our time series are created, we extract from them the word-level features described in Section 3.2. The feature values obtained for some example words are given in Table 6. We can see that each entry has a POS-tag prefix. As expected, features related to measures of central tendency (e.g., mean, median) from the same time series exhibit similar values.

Attribute	!-congrats	A-connected	A-upset	V-worried
sgd-so.last	1.4	0.4	-2.6	-1.2
sgd-so.mean	0.9	0.4	-2.1	-1.2
sgd-so.trunc.mean	0.9	0.4	-2.1	-1.2
sgd-so.median	0.9	0.4	-2.1	-1.2
sgd-so.sd	0.3	0.3	0.4	0.4
sgd-so.sg	0.0	0.0	0.0	0.0
sgd-so.sg.diff	0.0	0.0	0.1	0.0
sgd-so.iqr	0.2	0.3	0.5	0.6
pmi-so.last	2.8	-0.1	-3.3	-1.8
pmi-so.mean	2.9	0.1	-3.2	-1.9
pmi-so.trunc.mean	2.9	0.3	-3.3	-1.9
pmi-so.median	2.9	0.3	-3.2	-1.9
pmi-so.sd	0.2	0.8	0.1	0.1
pmi-so.sg	0.0	0.0	0.0	0.0
pmi-so.sg.diff	0.2	0.4	0.4	0.4
pmi-so.iqr	0.1	0.6	0.1	0.1
pmi-so.tag	interjection	adjective	adjective	verb
label	positive	neutral	negative	negative

Table 6: Word-level feature example.

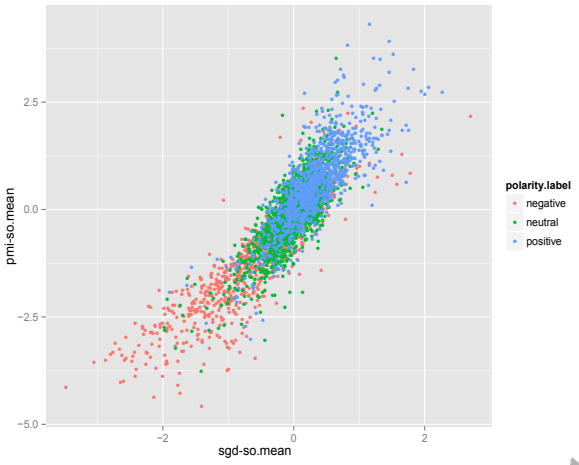

Figure 3: PMI-SO vs SGD-SO scatterplot.

A scatterplot between attributes **sgd-so.mean** and **pmi-so.mean** for the labelled words from the STS corpus is shown in Figure 3. From the figure we can observe that the two variables are highly correlated. The correlation is 0.858 and 0.877 for the ED.EM and STS corpora respectively. Positive, negative, and neutral words are depicted with different colours. We can observe that negative words tend to show low values of sgd-so.mean and pmi-so.mean, and positive words tend to show the opposite. Neutral words are more spread out and hard to distinguish. This pattern can also be clearly seen in the boxplots shown in Figure 4.

We can observe from the boxplots that the three classes of words can exhibit different statistical properties in both sgd-so.mean and pmi-so.mean. The medians of the classes show an accurate correspondence with the word's polarity, i.e., median(*pos*) > median(*neu*) > median(*neg*). However, note that the boxplots also show a substantial number of outliers.

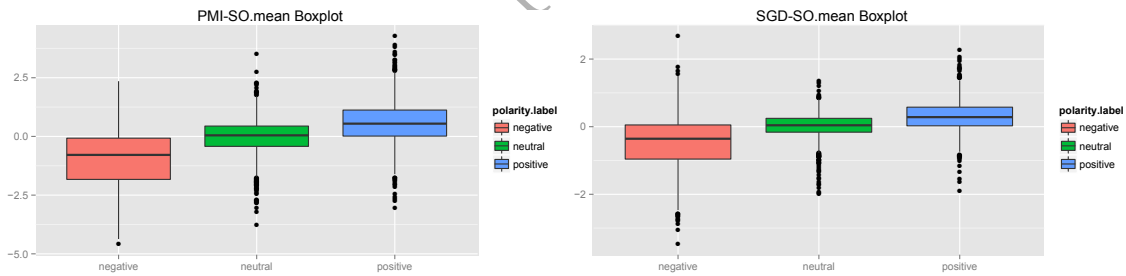

Figure 4: PMI-SO and SGD-SO Boxplots.

4.2. Word-level Classification

In this subsection, we focus on the word-level classification problem. With the aim of gaining a better understanding of the problem, we study three word-level classification problems:

1. *Neutrality*: Classify words as neutral (objective) or non-neutral (subjective). We label positive and negative words as non-neutral for this task.
2. *PosNeg*: Classify words as positive or negative. We remove all neutral words for this task.
3. *PosNegNeu*: Classify words as positive, negative or neutral. This is the primary classification problem we aim to solve.

In the first part of this subsection, we study word-level sentiment classification using tweets annotated with the emoticon-based annotation approach. Afterwards, we will study the same problem using tweets annotated with the model transfer approach.

4.2.1. Word Classification from Emoticon-annotated Tweets

We study the information provided by each feature with respect to the three classification tasks described above. This is done by calculating the information gain of each feature using the *R* package *FSelector*[14]. This score is normally used for decision tree learning and measures the reduction of entropy within each class after performing the best split induced by the feature. The information gain obtained for the different attributes in relation to the three classification tasks is shown in Table 7. The attributes achieving the highest information gain per task are marked in bold.

Dataset	ED.EM			STS		
Task	Neutrality	PosNeg	PosNegNeu	Neutrality	PosNeg	PosNegNeu
pos-tag	0.062	0.017	0.071	0.068	0.016	0.076
sgd-so.mean	**0.082**	0.233	0.200	**0.104**	0.276	**0.246**
sgd-so.trunc.mean	0.079	0.237	0.201	**0.104**	0.276	0.242
sgd-so.median	0.075	0.233	0.193	0.097	0.275	0.239
sgd-so.last	0.057	0.177	0.155	0.086	0.258	0.221
sgd-so.sd	0.020	0.038	0.034	0.030	0.030	0.052
sgd-so.sg	0.029	0.000	0.030	0.049	0.017	0.062
sgd-so.sg.diff	0.000	0.000	0.008	0.005	0.000	0.000
sgd-so.iqr	0.018	0.012	0.019	0.015	0.014	0.017
pmi-so.mean	0.079	0.283	**0.219**	0.081	**0.301**	0.232
pmi-so.trunc.mean	0.077	**0.284**	0.215	0.079	0.300	0.229
pmi-so.median	0.077	0.281	0.215	0.076	0.300	0.228
pmi-so.last	0.069	0.279	0.211	0.084	0.300	0.240
pmi-so.sd	0.000	0.015	0.008	0.000	0.012	0.007
pmi-so.sg	0.013	0.216	0.126	0.019	0.239	0.142
pmi-so.sg.diff	0.000	0.012	0.009	0.000	0.000	0.000
pmi-so.iqr	0.000	0.000	0.000	0.000	0.008	0.000

Table 7: Information gain values. Best result per column is given in bold.

We can observe that variables measuring the location of the PMI-SO and SGD-SO time series tend to be more informative than those measuring dispersion. Moreover, the information gain of these variables is much higher for PosNeg than for Neutrality. SGD-SO and PMI-SO are competitive measures for neutrality, but PMI-SO is better for PosNeg. An interesting insight is that features that measure the central tendency of the time series tend to be more informative than those giving the last value of the time series, especially for SGD-SO. These measures smooth the fluctuations of the SGD-SO time series. We can see that the feature *sgd-so.mean* is the best attribute for neutrality classification in both datasets. We can also see that POS tags are useful for neutrality detection, but useless for PosNeg. Therefore, we can conclude that positive and negative words have a similar distribution of POS tags.

We trained supervised classifiers for the three different classification problems using both emoticon-annotated datasets, STS and ED.EM. The classification experiments were performed using WEKA[15], a machine learning environment. We studied the following learning algorithms in preliminary experiments: RBF SVM, logistic regression, C4.5, and random forest. As the RBF SVM produced the best performance among the different methods, we used this method in our classification experiments, with a nested grid search procedure for parameter tuning where internal cross-validation is used to find the C and σ parameters of the RBF SVM.

The evaluation was done using stratified 10 times 10-fold-cross-validation and different subsets of attributes are compared. All the methods are compared with the baseline of using the last value of PMI-SO, based on the corrected resampled paired t-student test with an α level of 0.05 [29]. We used the following subsets of attributes: 1) *PMI-SO*: Includes only the feature *pmi-so.last*. This is the baseline and is equivalent to the standard PMI semantic orientation measure with the decision boundaries provided by the SVM. 2) *ALL*: Includes all the features. 3) *SGD-SO.TS+POS*:

[14] http://cran.r-project.org/web/packages/FSelector/

[15] http://www.cs.waikato.ac.nz/ml/weka/

Includes all the features from the SGD-SO time series and the POS tag. 4) *PMI-SO.TS+POS*: Includes all the features from the PMI-SO time series and the POS tag. 5) *PMI-SO+POS*: Includes the feature *pmi-so.last* and the POS tag.

We use two evaluation measures that are appropriate for imbalanced datasets: the weighted area under the ROC curves (AUCs) and the kappa statistic. ROC curves are insensitive to class balance because they include all true positive and false positive rates that are observed [14]. The kappa statistic is also insensitive to class imbalance because it normalises the classification accuracy by the imbalance of the classes in the data.

The classification results for the four different subsets of attributes in the two datasets are presented in Table 8. The symbols + and − correspond to statistically significant improvements and degradations with respect to the baseline, respectively.

We can observe a much lower performance in Neutrality detection than in PosNeg. This indicates that the detection of neutral Twitter words is much harder than distinguishing between positive and negative words. The performance on both datasets tends to be similar. However, the results for STS are better than for ED.EM. This suggests that a collection of balanced positively and negatively labelled tweets is more suitable for lexicon expansion. Another result is that the combination of all features leads to a significant improvement over the baseline for Neutrality and PosNegNeu classification. In the PosNeg classification task, we can see that the baseline is very strong. This suggests that PMI-SO is very good for discriminating between positive and negative words, but not strong enough when neutral words are included. Regarding PMI-SO and SGD-SO time series, we can conclude that they are competitive for Neutrality detection. However, PMI-SO-based features are better for the PosNeg and PosNegNeu tasks.

AUC					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.EM-Neutrality	0.62 ± 0.02	**0.65** ± 0.02 +	**0.65** ± 0.02 +	**0.65** ± 0.02 +	0.64 ± 0.02 +
ED.EM-PosNeg	0.74 ± 0.03	**0.75** ± 0.03	0.71 ± 0.03 -	0.74 ± 0.03	0.73 ± 0.03
ED.EM-PosNegNeu	0.62 ± 0.02	**0.65** ± 0.02 +	0.64 ± 0.02	**0.65** ± 0.02 +	0.64 ± 0.02 +
STS-Neutrality	0.63 ± 0.02	**0.67** ± 0.02 +	0.66 ± 0.02 +	0.66 ± 0.02 +	0.66 ± 0.02 +
STS-PosNeg	**0.77** ± 0.03	**0.77** ± 0.03	0.75 ± 0.03 -	**0.77** ± 0.03	**0.77** ± 0.03
STS-PosNegNeu	0.64 ± 0.02	**0.66** ± 0.01 +	0.65 ± 0.02 +	**0.66** ± 0.02 +	**0.66** ± 0.02 +
Kappa					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.EM-Neutrality	0.23 ± 0.04	**0.3** ± 0.04 +	0.29 ± 0.05 +	**0.3** ± 0.04 +	0.28 ± 0.04 +
ED.EM-PosNeg	0.48 ± 0.06	**0.5** ± 0.06	0.44 ± 0.05	0.49 ± 0.06	0.48 ± 0.06
ED.EM-PosNegNeu	0.28 ± 0.04	**0.33** ± 0.04 +	0.3 ± 0.04	**0.33** ± 0.04 +	0.32 ± 0.04 +
STS-Neutrality	0.26 ± 0.04	**0.33** ± 0.04 +	0.31 ± 0.05 +	0.32 ± 0.04 +	0.32 ± 0.04 +
STS-PosNeg	**0.54** ± 0.06	**0.54** ± 0.06	0.51 ± 0.06 -	0.53 ± 0.06	**0.54** ± 0.05
STS-PosNegNeu	0.31 ± 0.04	**0.35** ± 0.03 +	0.34 ± 0.03 +	0.34 ± 0.03 +	0.34 ± 0.03 +

Table 8: World-level classification performance with emoticon-based annotation. Best result per row is given in bold.

4.2.2. Word Classification from Model Transfer Annotated Tweets

We also study the classification of words from the data annotated with the model transfer approach. As described in Section 3.1, our soft-annotated collection ED.SL is built by taking STS as the source corpus and a sample of 10 million tweets from ED as the target one. We study two different mechanisms for extracting word-level attributes from the soft-annotated collection of tweets. In the first one, we convert the message-level soft labels into hard ones by imposing different thresholds (λ) for the positive soft label, and we calculate the same attributes used for the emoticon-annotated data. Taking steps of 0.1, we vary the value of λ from 0.6 to 0.9 and obtain four hard-annotated datasets. In the second approach, we calculate the features directly from the soft labels by relying on the squared loss (Equation 6) for building the SGD-SO time series and on partial counts (Equation 7) for building the PMI-SO time series.

In this way, we obtain four hard-annotated datasets and one soft-annotated dataset. We calculate the corresponding

word-level attributes (see Section 3.2) for the 10,000 most frequent POS-disambiguated words from each of the five datasets. As the most frequent words matching the seed lexicon are not necessarily the same among the different datasets, we take the intersection of them in order to make them comparable. We trained RBF SVMs on the different collections over the intersection of the labelled words using two different feature spaces. In the first one, we use all the attributes, and in the second one, we discard the POS attribute, which is the only feature that is independent of the threshold or the message-level label. The 3-class word-level polarity classification accuracies and kappa values obtained by the different RBF SVMs are shown in Table 9.

	ALL		NO POS	
Dataset	Accuracy	Kappa	Accuracy	Kappa
ED.T06	62.82 ± 1.78	**0.34** ± 0.03	61.29 ± 2.02	0.31 ± 0.04
ED.T07	62.77 ± 1.78	**0.34** ± 0.03	**61.60** ± 1.98	**0.32** ± 0.04
ED.T08	62.43 ± 1.83	0.33 ± 0.04	61.03 ± 1.83	0.30 ± 0.03
ED.T09	62.46 ± 1.82	0.33 ± 0.03	60.20 ± 1.89	0.29 ± 0.04
Soft Labels	**63.05** ± 1.81	**0.34** ± 0.03	60.92 ± 2.10	0.30 ± 0.04

Table 9: Word classification performance using model transfer. Best result per column is given in bold.

The results indicate that the different thresholds and the soft labels produce similar results. Indeed, there are no statistically significant differences among them. However, it is worth mentioning that the soft labels produce a better accuracy than the hard ones when all the attributes are included. Regarding the kappa values, we observe that they become more distinguishable when the POS label is discarded. As ED.T07 achieved the best kappa values for both attribute spaces, we select 0.7 as the best value of λ.

Next, we study the performance of the different feature subsets in the model transfer data. We repeat the previous word-level classification experiments conducted on the emoticon-annotated datasets on the the soft-annotated collection (ED.SL) and the best hard-annotated collection (ED.T07). The same four different subsets of attributes are compared and we use again the last value of the PMI-SO series as the baseline. The results are exhibited in Table 10.

AUC					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.T07-Neutrality	0.62 ± 0.02	**0.65** ± 0.02 +	**0.65** ± 0.02 +	0.64 ± 0.02	0.64 ± 0.02
ED.T07-PosNeg	0.77 ± 0.03	0.76 ± 0.02	0.74 ± 0.03 -	**0.78** ± 0.03	0.77 ± 0.03
ED.T07-PosNegNeu	0.62 ± 0.02	**0.65** ± 0.02 +	0.65 ± 0.02 +	0.64 ± 0.02	0.64 ± 0.01
ED.SL-Neutrality	0.62 ± 0.02	**0.65** ± 0.02 +	**0.65** ± 0.02 +	0.64 ± 0.02 +	0.64 ± 0.02 +
ED.SL-PosNeg	**0.78** ± 0.03	**0.78** ± 0.03	0.74 ± 0.03 -	**0.78** ± 0.03	**0.78** ± 0.03
ED.SL-PosNegNeu	0.63 ± 0.02	**0.65** ± 0.02 +	0.64 ± 0.02	0.64 ± 0.02	0.64 ± 0.02
Kappa					
Dataset	PMI-SO	ALL	SGD-SO.TS+POS	PMI-SO.TS+POS	PMI-SO+POS
ED.T07-Neutrality	0.23 ± 0.03	0.29 ± 0.04 +	**0.31** ± 0.04 +	0.27 ± 0.04	0.28 ± 0.04
ED.T07-PosNeg	**0.56** ± 0.06	0.54 ± 0.05	0.49 ± 0.06 -	**0.56** ± 0.05	**0.56** ± 0.05
ED.T07-PosNegNeu	0.27 ± 0.03	**0.34** ± 0.04 +	0.33 ± 0.03 +	0.31 ± 0.05	0.31 ± 0.03
ED.SL-Neutrality	0.24 ± 0.05	**0.3** ± 0.04 +	0.29 ± 0.04 +	0.28 ± 0.04 +	0.28 ± 0.04 +
ED.SL-PosNeg	0.56 ± 0.06	**0.57** ± 0.05	0.49 ± 0.05 -	**0.57** ± 0.05	**0.57** ± 0.06
ED.SL-PosNegNeu	0.3 ± 0.04	**0.33** ± 0.04 +	0.31 ± 0.03	0.32 ± 0.04	0.32 ± 0.04

Table 10: World-level classification performance using model transfer. Best result per row is given in bold.

Similarly to the results for the emoticon-annotated experiments shown in Table 8, the results are better for PosNeg than for Neutrality, and the combination of all the attributes produced a significant improvement over semantic orientation for 3-class PosNegNeu detection. Indeed, the full attribute space is the only representation that outperforms the

baseline for both collections in both AUC and kappa evaluation measures. Another difference between these results and the previous ones is observed for the detection of neutrality. Both PMI-SO and SGD-SO achieved very similar results in the previous experiments, but SGD-SO produces better results here.

We can see that the way in which our different features complement each other becomes clearer when they are calculated from tweets annotated with the model transfer approach.

4.3. Lexicon expansion

The ultimate goal of the polarity classification of words is to produce a Twitter-oriented opinion lexicon emulating the properties of SentiWordet, i.e., a lexicon of POS-tagged disambiguated entries with their corresponding distribution for positive, negative, and neutral classes. To do this, we fit logistic regression models to the outputs of the support vector machines trained for the *PosNegNeu* problem, using all the attributes. The resulting models are then used to classify the remaining unlabelled words. This process is performed for the STS, ED.EM, ED.T07, and ED.SL datasets.

A sample from the expanded word list produced with the STS collection is given in Table 11. We can see that each entry has the following attributes: the word, the POS-tag, the sentiment label that corresponds to the class with maximum probability, and the distribution. We inspected the expanded lexicon and observed that the estimated probabilities are intuitively plausible. However, there are some words for which the estimated distribution is questionable, such as the word *same* in Table 11. We can also observe that words such as *close* and *laugh*, which have more than one POS-tag, receive disambiguated sentiment distributions. We observe that these disambiguations are intuitively plausible as well.

word	POS	label	negative	neutral	positive
alrighty	interjection	positive	0.021	0.087	0.892
anniversary	common.noun	neutral	0.074	0.586	0.339
boooooo	interjection	negative	0.984	0.013	0.003
close	adjective	positive	0.352	0.267	0.381
close	verb	neutral	0.353	0.511	0.136
french	adjective	neutral	0.357	0.358	0.285
handsome	adjective	positive	0.007	0.026	0.968
laugh	common.noun	neutral	0.09	0.504	0.406
laugh	verb	positive	0.057	0.214	0.729
lmaoo	interjection	positive	0.19	0.338	0.472
relaxing	verb	positive	0.064	0.244	0.692
saddest	adjective	negative	0.998	0.002	0
same	adjective	negative	0.604	0.195	0.201
tear	common.noun	negative	0.833	0.124	0.044
wikipedia	proper.noun	neutral	0.102	0.644	0.254

Table 11: Example list of words in expanded lexicon.

The provided probabilities can also be used to explore the sentiment intensities of words. In Figure 5, we visualise the expanded lexicon intensities of words classified as positive and negative through word clouds. The sizes of the words are proportional to the log odds ratios $\log_2(\frac{P(pos)}{P(neg)})$ and $\log_2(\frac{P(neg)}{P(pos)})$ for positive and negative words, respectively.

4.4. Extrinsic Evaluation of the Expanded Lexicons

In this subsection we study the usefulness of our expanded lexicons in an extrinsic task: polarity classification of tweets. This involves categorising entire tweets into a positive or negative sentiment class. The goal of this experiment is to show how the expanded lexicons can be used to improve the message-level classification performance achieved by using the manually annotated seed lexicon and to compare the created resources with two other existing resources that have been widely used for sentiment analysis: SentiWordNet and SentiStrength [37].

As was previously discussed in Section 2, SentiWordNet is a resource in which each WordNet synset is assigned a probability distribution of positive, negative, and neutral classes. Synsets in WordNet are sets of word senses with equivalent meaning. A word with multiple senses or meanings is included in multiple WordNet synsets and, in turn, is associated with multiple sentiment distributions in SentiWordNet. In WordNet, all the senses of a word are ranked according to their frequency of use or popularity. As suggested in the sample code provided by the

(a) (b)

Figure 5: Word clouds of positive and negative words using log odds proportions.

SentiWordNet webpage[16], we calculate a weighted average of the sentiment distributions of the synsets in which a POS-disambiguated word occurs in order to obtain a single sentiment distribution for it. The weights correspond to the reciprocal ranks of the senses in order to give higher weights to the most popular senses of a word.

SentiStrength[17] is a lexicon-based sentiment analysis method that returns positive and negative numerical scores for a given text passage. The positive score ranges from 1 (not positive) to 5 (extremely positive) and the negative one ranges from -1 (not negative) to -5 (extremely negative). The lexicon is hand-annotated and includes both formal English words and informal words used in social media (e.g., luv, lol), scored by sentiment. The scores can also be adapted to a specific domain using machine learning. SentiStrength applies linguistic rules for dealing with negations, questions, booster words, and emoticons. These are used together with the lexicon for computing the positive and negative outputs.

The evaluation is performed on three collections of tweets that were manually assigned to the positive and negative class. The first collection is *6HumanCoded*[18], which was used to evaluate SentiStrength. In this dataset, tweets are scored according to positive and negative numerical scores. We use the difference of these scores to create polarity classes and discard messages where it is equal to zero. The other datasets are *Sanders*[19], and *SemEval*[20]. The number of positive and negative tweets per corpus is given in Table 12.

	Positive	Negative	Total
6HumanCoded	1340	949	2289
Sanders	570	654	1224
SemEval	5232	2067	7299

Table 12: Message-level polarity classification datasets.

We train different logistic regression models on the labelled collections of tweets, based on simple features calculated from the seed lexicon, SentiWordNet, SentiStrength, and from the four expanded lexicons: STS, ED.EM, ED.SL, and ED.T07. For each resource we compute a positive and a negative feature. From the seed lexicon we count the number of positive and negative words matching the content of the tweet. In order to use the POS-disambiguated lexicons such as SentiWordNet and our expanded lexicons, we tag the tweet's words according to POS classes. Then, we calculate the corresponding positive feature by adding the positive probabilities of POS-tagged words labelled as positive within the tweet's content. Likewise, the corresponding negative feature is calculated in an analogous way

[16] http://sentiwordnet.isti.cnr.it/code/SentiWordNetDemoCode.java
[17] http://sentistrength.wlv.ac.uk/
[18] http://sentistrength.wlv.ac.uk/documentation/6humanCodedDataSets.zip
[19] http://www.sananalytics.com/lab/twitter-sentiment/
[20] http://www.cs.york.ac.uk/semeval-2013/task2/

from the negative probabilities. In the expanded lexicons, words are discarded as non-opinion words whenever the class with the highest probability corresponds to the neutral one. For SentiStrength we use the positive and negative scores returned by the method for the target tweet.

We study eight different setups based on these attributes. The first one is the seed lexicon baseline which includes only the two attributes calculated from the seed lexicon. The second one corresponds to the SentiWordNet baseline and includes the positive and negative features calculated from it. The third one corresponds to the SentiStrength baseline, which includes the positive and negative scores returned by the method. The next four setups, STS, ED.EM, ED.SL, and ED.707, include the pair of features provided by each corresponding expanded lexicon together with the two features from the seed lexicon. This is done because the expanded lexicons do not contain the words from the seed lexicons that were used to train them. Finally, the last setup, ENS is an ensemble of the four expanded lexicons and the seed lexicon by including the ten features associated with these resources.

In the same way as in the word-level classification task, we use the weighted AUC and the kappa coefficient as evaluation measures, estimated using 10-times 10-fold cross-validation, and we compare the different setups with the three baselines using corrected paired t-tests. The classification results obtained for the different setups are shown in Table 13. The statistical significance tests of each setup with respect to each of the three baselines (seed lexicon, SentiWordNet, and SentiStrength) are indicated by a sequence of three symbols. Improvements are denoted by a plus (+), degradations by a minus (-), and cases where no statistical significant difference is observed by an equals (=).

AUC			
Dataset	6HumanCoded	Sanders	SemEval
Seed.Lex	0.77 ± 0.03 = + -	0.77 ± 0.04 = + =	0.77 ± 0.02 = + -
SW	0.74 ± 0.03 - = -	0.7 ± 0.05 - = -	0.76 ± 0.02 = = -
SS	0.81 ± 0.02 + + =	0.78 ± 0.03 = + =	0.81 ± 0.02 + + =
STS	0.82 ± 0.02 + + =	**0.84** ± 0.04 + + +	**0.83** ± 0.02 + + +
ED.EM	0.82 ± 0.03 + + =	0.83 ± 0.04 + + +	0.81 ± 0.02 + + =
ED.SL	0.81 ± 0.02 + + =	0.83 ± 0.04 + + +	0.81 ± 0.02 + + =
ED.T07	0.81 ± 0.03 + + =	0.83 ± 0.04 + + +	0.82 ± 0.02 + + +
ENS	**0.83** ± 0.02 + + =	**0.84** ± 0.04 + + +	**0.83** ± 0.02 + + +
Kappa			
Dataset	6HumanCoded	Sanders	SemEval
Seed Lex	0.4 ± 0.06 = + -	0.42 ± 0.08 = + =	0.35 ± 0.04 = + -
SW	0.32 ± 0.06 - = -	0.26 ± 0.1 - = -	0.3 ± 0.04 - = -
SS	**0.52** ± 0.05 + + =	0.45 ± 0.06 = + =	0.38 ± 0.03 + + =
STS	0.47 ± 0.06 + + =	**0.55** ± 0.08 + + +	0.38 ± 0.04 + + =
ED.EM	0.47 ± 0.05 + + -	0.54 ± 0.07 + + +	0.35 ± 0.04 = + -
ED.SL	0.46 ± 0.05 + + -	0.54 ± 0.08 + + +	0.36 ± 0.04 = + =
ED.T07	0.47 ± 0.05 + + -	0.53 ± 0.08 + + +	0.4 ± 0.04 + + =
ENS	0.49 ± 0.05 + + =	0.54 ± 0.07 + + +	**0.42** ± 0.04 + + +

Table 13: Message-level polarity classification performance. Best result per column is given in bold.

The results indicate that the expanded lexicons produce meaningful improvements in performance over the seed lexicon and over SentiWordNet on the different datasets. We believe that the reason why SentiWordNet is not achieving good results is its lack of informal English expressions. SentiStrength on the other hand, is a strong baseline for Twitter sentiment analysis. This is because of two reasons: 1) it is based on a lexicon formed by both formal and informal English words, and 2) it includes linguistic rules for handling negations and intensifiers. We observe that most of our expanded lexicons are at least competitive with SentiStrength according to the statistical tests. Moreover, there are several cases in Sanders and SemEval in which the expanded lexicons achieve statistically significant improvements over SentiStrength, especially for AUC. This is noteworthy, considering that the features we calculate from the expanded lexicons are based on simple additions of prior sentiment scores in contrast to the linguistic rules that

SentiStrength uses for aggregating its lexicon's words. Most of the cases where the expanded lexicons are statistically significantly worse than SentiStrength occur for the kappa measure in *6HumanCoded*.

Regarding the lexicons built from emoticon-annotated data, the performance of STS is slightly better than that of ED.EM. This pattern was also observed in the word-level classification performance shown in Table 8. This suggests that the two different ways of evaluating the lexicon expansion, one at the word level and the other at the message level, are consistent with each other. Regarding the lexicons built from the model transfer annotation approach, the results are competitive with the ones achieved with the emoticon-annotated data. Moreover, the lexicon built from hard transfered labels appears to be slightly better than the one built using soft labels, especially in the kappa value for the *SemEval* dataset.

We can also observe, in the majority of the cases, that the best performance is obtained by the ensemble of expanded lexicons. Therefore, we can conclude that lexicons expanded using either data from different collections or by applying different annotation approaches can be combined to improve message-level classification of tweets.

As was discussed in the previous section, the seed lexicon does not provide POS-tagged entries. Therefore, words exhibiting multiple POS-tags were labelled with the same polarity in the word-level training data. We also mentioned that this assumption could make the classifier learn spurious patterns and erroneously classify unlabelled words. For example the word *ill* together with the POS tag *nominal+verb* receives a negative label. However, when *ill* is used with the part-of-speech tag that refers to the contraction of the pronoun *I* and the verb *will*, it should be labelled as neutral instead. Moreover, by inspection we realised that *ill* is the only labelled entry with the POS tag *nominal+verb*. Considering that the POS tag is also used as a feature in the word-level classifiers, we observed that most of the words exhibiting the POS tag *nominal+verb* were classified into the negative class in all expanded lexicons. Common sense suggests that these words should be expanded to the neutral class.

In order to avoid learning this type of spurious pattern, we re-trained the word-level classifiers using an outlier removal technique. More specifically, we clean out the instances from the word-level training data that are misclassified by a classifier evaluated using 10-fold cross-validation on this data, again using an RBF SVM. Afterwards, we retrain the word-level classifiers on the cleaned data and create new versions of all the expanded lexicons. We inspected the new versions of the expanded lexicons observing that words exhibiting the *nominal+verb* POS tag are classified to the neutral class as common sense suggests. This shows that removing outliers is a successful way of tackling ambiguities such as the one produced by the word *ill* in the seed lexicon.

AUC			
Dataset	6HumanCoded	Sanders	SemEval
STS	0.82 ± 0.03 + + =	0.82 ± 0.04 + + + ↓	0.82 ± 0.02 + + + ↓
ED.EM	0.84 ± 0.02 + + + ↑	0.83 ± 0.04 + + +	0.83 ± 0.02 + + + ↑
ED.SL	0.82 ± 0.03 + + = ↑	0.83 ± 0.04 + + +	0.82 ± 0.02 + + = ↑
ED.T07	0.81 ± 0.03 + + =	0.83 ± 0.04 + + +	0.82 ± 0.02 + + +
ENS	**0.84** ± 0.02 + + + ↑	**0.84** ± 0.04 + + +	**0.84** ± 0.02 + + + ↑
Kappa			
Dataset	6HumanCoded	Sanders	SemEval
STS	0.48 ± 0.05 + + = ↑	0.52 ± 0.08 + + = ↓	0.36 ± 0.03 = + = ↓
ED.EM	0.51 ± 0.05 + + = ↑	**0.53** ± 0.08 + + + ↓	0.41 ± 0.03 + + = ↑
ED.SL	0.48 ± 0.05 + + = ↑	**0.53** ± 0.08 + + + ↓	0.37 ± 0.04 = + = ↑
ED.T07	0.48 ± 0.06 + + = ↑	**0.53** ± 0.08 + + +	0.39 ± 0.04 + + = ↓
ENS	**0.52** ± 0.05 + + = ↑	**0.53** ± 0.07 + + + ↓	**0.42** ± 0.03 + + +

Table 14: Message-level polarity classification performance with outlier removal. Best result per columns is given in bold.

The message-level classification results obtained by the expanded lexicons with outlier removal are shown in Table 14. The improvements or degradations over the previous expanded lexicons are denoted with symbols ↑ and ↓ respectively. In relation to *6HumanCoded*, we observe that the AUC metric is improved for almost all the different setups, and that all the setups outperform the previous kappa results. It is noteworthy to mention that the kappa value achieved by the ensemble of lexicons in this dataset exceeds the previous value by 0.03. On the other hand, we

see degradations in the kappa values for *Sanders*. Regarding *SemEval*, we see a degradation in STS, a substantial improvement in ED.EM, and a minor improvement in ED.SL. Another interesting result is that the removal of outliers creates lexicons that are always equal or better than SentiStrength according to the statistical tests.

The fact that the removal of outliers can also produce degradations in the quality of the expanded lexicons, as in the case of the *Sanders* dataset, indicates that words that are useful for learning the word-level classifier have also been removed. This suggests that the problem of reducing the noise in the seed lexicon is hard to address in an automatic fashion. A simple but labour-intensive approach to overcome this problem would be to manually clean the labelled POS-disambiguated words.

5. Conclusions

In this article, we have presented a method for opinion lexicon expansion in the context of tweets. The method exploits information from three types of information sources, all of which are relatively cheap to obtain: emoticon-annotated tweets, unlabelled tweets, and hand-annotated lexicons. The method creates a lexical resource with disambiguated POS entries and a probability distribution for positive, negative, and neutral classes. To the best of our knowledge, our method is the first approach for creating a Twitter opinion lexicon with these characteristics. Considering that these characteristics are very similar to those of SentiWordNet, a well-known publicly available lexical resource, we believe that several sentiment analysis methods that are based on SentiWordNet can be easily adapted to Twitter by relying on our expanded lexicons[21]. Moreover, our expanded resources have shown to outperform the tweet-level polarity classification performance achieved by SentiWordNet and SentiStrength in most cases.

The word-level experimental results show that the supervised fusion of POS tags, SGD-SO, and PMI-SO, produces a significant improvement for three-dimensional word-level polarity classification compared to using PMI semantic orientation alone. We can also conclude that attributes describing the central location of SGD-SO and PMI-SO time series tend to be more informative than the last values of the series because they smooth the temporal fluctuations in the sentiment pattern of a word.

There are many domains, such as politics, in which emoticons are not frequently used to express positive and negative opinions. This is an important limitation of previous approaches for domain-specific lexicon expansion that are based solely on emoticon-annotated tweets. The proposed model transfer annotation approach tackles this problem and enables inference of opinion words from any collection of unlabelled tweets.

We have also proposed a novel way for computing word-level attributes from data with soft labels. The proposed soft version of PMI-SO based on partial counts can be used for expanding lexicons from any collection of tweets in an unsupervised fashion. In contrast to a threshold approach, soft PMI-SO is parameter free and avoids discarding tweets that may contain valuable words.

Our supervised framework for lexicon expansion opens several directions for further research. For instance, this approach could be used for creating a concept-level opinion resource for Twitter by relying on word clustering techniques such as the Brown Clustering method [9]. We could build the same time series we have built for words for word-clusters, and use the trained classier for estimating a sentiment distribution for each word cluster or concept.

We believe that unlabelled words and their feature values could provide valuable information that is not being exploited so far. Semi-supervised methods such as the EM algorithm [31] could be used to include unlabelled words as part of the training process.

Because our word-level features are based on time series, they could be easily calculated in an on-line fashion from a stream of time-evolving tweets. Based on this, we could study the dynamics of opinion words. New opinion words could be discovered because the change of the distribution in certain words could be tracked. This approach could be used for online lexicon expansion in specific domains, and potentially be useful for high-impact events on Twitter, such as elections and sports competitions.

6. Acknowledgment

Felipe Bravo-Marquez was supported by a doctoral scholarship from The University of Waikato.

[21] The expanded lexicons and the source code used to generate them are available for download at `http://www.cs.waikato.ac.nz/ml/sa/lex.html#kbs16`.

References

[1] Amir, S., Ling, W., Astudillo, R., Martins, B., Silva, M. J., & Trancoso, I. (2015). Inesc-id: A regression model for large scale twitter sentiment lexicon induction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (pp. 613–618). Denver, Colorado: Association for Computational Linguistics.

[2] Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation* LREC'10 (pp. 2200–2204). Valletta, Malta.

[3] Bahrainian, S. A., Liwicki, M., & Dengel, A. (2014). Fuzzy subjective sentiment phrases: A context sensitive and self-maintaining sentiment lexicon. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01* WI-IAT '14 (pp. 361–368). Washington, DC, USA: IEEE Computer Society.

[4] Becker, L., Erhart, G., Skiba, D., & Matula, V. (2013). Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises* SemEval'13 (pp. 333–340).

[5] Bifet, A., & Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. In *Proceedings of the 13th international conference on Discovery science* DS'10 (pp. 1–15). Berlin, Heidelberg: Springer-Verlag.

[6] Bravo-Marquez, F., Frank, E., & Pfahringer, B. (2015). From unlabelled tweets to twitter-specific opinion words. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* SIGIR '15 (pp. 743–746). New York, NY, USA: ACM.

[7] Bravo-Marquez, F., Frank, E., & Pfahringer, B. (2015). Positive, negative, or neutral: Learning an expanded lexicon from emoticon-annotated tweets. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence* IJCAI '15 (pp. 1229–1235). AAAI Press.

[8] Bravo-Marquez, F., Mendoza, M., & Poblete, B. (2014). Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, *69*, 86 – 99.

[9] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, *18*, 467–479.

[10] Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems*, *31*, 102–107.

[11] Cambria, E., & Hussain, A. (2015). *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Cham, Switzerland: Springer International Publishing.

[12] Esuli, A., & Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* CIKM '05 (pp. 617–624). New York, NY, USA: ACM.

[13] Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation* LREC'06 (pp. 417–422).

[14] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recogn. Lett.*, *27*, 861–874.

[15] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., & Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2* (pp. 42–47). Association for Computational Linguistics.

[16] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, .

[17] Hatzivassiloglou, V., & Wiebe, J. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *COLING* (pp. 299–305).

[18] Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* KDD '04 (pp. 168–177). New York, NY, USA: ACM.

[19] Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent Twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 151–160). Association for Computational Linguistics.

[20] Kamps, J., Marx, M., Mokken, R. J., & De Rijke, M. (2004). Using WordNet to Measure Semantic Orientation of Adjectives. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)* (pp. 1115–1118). European Language Resources Association (ELRA) volume 4.

[21] Kim, S.-M., & Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics* COLING '04 (pp. 1367–1373). Stroudsburg, PA, USA: Association for Computational Linguistics.

[22] Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, (pp. 723–762).

[23] Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, *11*, 538–541.

[24] Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management* CIKM '09 (pp. 375–384). New York, NY, USA: ACM.

[25] Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

[26] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 3111–3119). Curran Associates, Inc.

[27] Mohammad, S., & Turney, P. D. (2013). Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, *29*, 436–465.

[28] Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises* SemEval'13 (pp. 321–327).

[29] Nadeau, C., & Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, *52*, 239–281.

[30] Nielsen, F. r. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages* #MSM2011 (pp. 93–98).

[31] Nigam, K., McCallum, A., & Mitchell, T. (2006). Semi-supervised text classification using em. *Semi-Supervised Learning*, (pp. 33–56).

[32] Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)* (pp. 1320–1326). European Language Resources Association (ELRA).

[33] Petrović, S., Osborne, M., & Lavrenko, V. (2010). The Edinburgh Twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media* WSA '10 (pp. 25–26). Stroudsburg, PA, USA: Association for Computational Linguistics.

[34] Poria, S., Gelbukh, A. F., Cambria, E., Hussain, A., & Huang, G. (2014). EmoSenticSpace: A novel framework for affective common-sense reasoning. *Knowledge-Based Systems*, *69*, 108–123.

[35] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, *37*, 267–307.

[36] Tang, D., Wei, F., Qin, B., Zhou, M., & Liu, T. (2014). Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland* (pp. 172–182).

[37] Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *JASIST*, *63*, 163–173.

[38] Tsai, A.-R., Wu, C.-E., Tsai, R.-H., & Hsu, J. (2013). Building a concept-level sentiment dictionary based on commonsense knowledge. *Intelligent Systems, IEEE*, *28*, 22–30.

[39] Turney, P. D. (2002). 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 (pp. 417–424). Stroudsburg, PA, USA: Association for Computational Linguistics.

[40] Valitutti, R. (2004). Wordnet-affect: an affective extension of wordnet. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation* (pp. 1083–1086).

[41] Weichselbraun, A., Gindl, S., & Scharl, A. (2014). Enriching semantic knowledge bases for opinion mining in big data applications. *Knowledge-Based Systems*, *69*, 78–85.

[42] Wilks, Y., & Stevenson, M. (1998). The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Natural Language Engineering*, *4*, 135–143.

[43] Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* HLT '05 (pp. 347–354). Stroudsburg, PA, USA: Association for Computational Linguistics.

[44] Wu, C., & Tsai, R. T. (2014). Using relation selection to improve value propagation in a conceptnet-based sentiment dictionary. *Knowledge-Based Systems*, *69*, 100–107.

[45] Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning* ICML '04 (pp. 919–926). New York, NY, USA: ACM.

[46] Zhou, Z., Zhang, X., & Sanderson, M. (2014). Sentiment analysis on twitter through topic-based lexicon expansion. In H. Wang, & M. Sharaf (Eds.), *Databases Theory and Applications* (pp. 98–109). Springer International Publishing volume 8506 of *Lecture Notes in Computer Science*.

[47] Zirn, C., Niepert, M., Stuckenschmidt, H., & Strube, M. (2011). Fine-grained sentiment analysis with structural features. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)* (pp. 336–344).