Generating a Word-Emotion Lexicon from #Emotional Tweets

Anil Bandhakavi[1] Nirmalie Wiratunga[1] Deepak P[2] Stewart Massie[1]

[1] IDEAS Research Institute, Robert Gordon University, Scotland, UK
[2] IBM Research - India, Bangalore, India
{a.s.bandhakavi,n.wiratunga}@rgu.ac.uk
deepaksp@acm.org, s.massie@rgu.ac.uk

Abstract

Research in emotion analysis of text suggest that emotion lexicon based features are superior to corpus based n-gram features. However the static nature of the general purpose emotion lexicons make them less suited to social media analysis, where the need to adopt to changes in vocabulary usage and context is crucial. In this paper we propose a set of methods to extract a word-emotion lexicon automatically from an emotion labelled corpus of tweets. Our results confirm that the features derived from these lexicons outperform the standard Bag-of-words features when applied to an emotion classification task. Furthermore, a comparative analysis with both manually crafted lexicons and a state-of-the-art lexicon generated using Point-Wise Mutual Information, show that the lexicons generated from the proposed methods lead to significantly better classification performance.

1 Introduction

Emotion mining or affect sensing is the computational study of natural language expressions in order to quantify their associations with different emotions (e.g. anger, fear, joy, sadness and surprise). It has a number of applications for the industry, commerce and government organisations, but uptake has arguably been slow. This in part is due to the challenges involved with modelling subjectivity and complexity of the emotive content. However, use of qualitative metrics to capture emotive strength and extraction of features from these metrics has in recent years shown promise (Shaikh, 2009). A general-purpose emotion lexicon (GPEL) is a commonly used resource that allows qualitative assessment of a piece of emotive text. Given a word and an emotion, the lexicon provides a score to quantify the strength of emotion expressed by that word. Such lexicons are carefully crafted and are utilised by both supervised and unsupervised algorithms to directly aggregate an overall emotion score or indirectly derive features for emotion classification tasks (Mohammad, 2012a), (Mohammad, 2012b).

Socio-linguistics suggest that social media is a popular means for people to converse with individuals, groups and the world in general (Boyd et al., 2010). These conversations often involve usage of non-standard natural language expressions which consistently evolve. Twitter and Facebook were credited for providing momentum for the 2011 Arab Spring and Occupy Wall street movements (Ray, 2011),(Skinner, 2011). Therefore efforts to model social conversations would provide valuable insights into how people influence each other through emotional expressions. Emotion analysis in such domains calls for automated discovery of lexicons. This is so since learnt lexicons can intuitively capture the evolving nature of vocabulary in such domains better than GPELs.

In this work we show how an emotion labelled corpus can be leveraged to generate a word-emotion lexicon automatically. Key to this is the availability of a labelled corpus which may be obtained using a distance-supervised approach to labelling (Wang et al., 2012). In this paper we propose three lexicon generation methods and evaluate the quality of these by deploying them in an emotion classification task. We show through our experiments that the word-emotion lexicon generated using the proposed methods in this paper significantly outperforms GPELs such as WordnetAffect, NRC word-emotion association lexicon and a leaxicon learnt using Point-wise Mutual Information (PMI). Additionally, our lexicons also outperform the traditional Bag-of-Words representation.

The rest of the paper is organised as follows: In

*Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 12–21,
Dublin, Ireland, August 23-24 2014.

Section 2 we present the related work. In Section 3 we outline the problem. In Section 4 we formulate the different methods proposed to generate the word-emotion lexicons. In Section 5 we discuss experimental results followed by conclusions and future work in Section 6.

2 Related Work

Computational emotion analysis, draws from cognitive and physiology studies to establish the key emotion categories; and NLP and text mining research to establish features designed to represent emotive content. Emotion analysis has been applied in a variety of domains: fairy tales (Francisco and Gervas, 2006; Alm et al., 2005); blogs (Mihalcea and Liu, 2006; Neviarouskaya et al., 2010), novels (John et al., 2006), chat messages (E.Holzman and William M, 2003; Ma et al., 2005; Mohammad and Yang, 2011) and emotional events on social media content(Kim et al., 2009). Comparative studies on emotive word distributions on micro-blogs and personal content (e.g. love letters, suicide notes) have shown that emotions such as *disgust* are expressed well in tweets. Further, expression of emotion in tweets and love letters have been shown to have similarities(K. Roberts and Harabagiu, 2012).

Emotion classification frameworks provide insights into human emotion expressions (Ekman, 1992; Plutchik, 1980; Parrott, 2001). The emotions proposed by (Ekman, 1992) are popular in emotion classification tasks (Mohammad, 2012b; Aman and Szpakowicz, 2008). Recently there has also been interest in extending this basic emotion framework to model more complex emotions (such as politeness, rudeness, deception, depression, vigour and confusion) (Pearl and Steyvers, 2010; Bollen et al., 2009). A common theme across these approaches involves the selection of emotion-rich features and learning of relevant weights to capture emotion strength (Mohammad, 2012a; Qadir and Riloff, 2013).

Usefulness of a lexicon: Lexicons such as Wordnet Affect (Strapparava and Valitutti, 2004) and NRC (Saif M. Mohammad, 2013)) are very valuable resources from which emotion features can be derived for text representation. These are manually crafted and typically contain emotion-rich formal vocabulary. Hybrid approaches that combine features derived from these static lexicons with n-grams have resulted in better performance than either alone (Mohammad, 2012b),(Aman and Szpakowicz, 2008). However the informal and dynamic nature of social media content makes it harder to adopt these lexicons for emotion analysis. An alternative strategy is to derive features from a dynamic (i.e., learnt) lexicon. Here association metrics such as Pointwise Mutual Information (PMI) can be used to model emotion polarity between a word and emotion labelled content (Mohammad, 2012a). Such approaches will be used as baselines to compare against our proposed lexicon generation strategies. There are other lexicon generation methods proposed by Rao .et. al (Yanghui Rao and Chen, 2013) and Yang .et. al (Yang et al., 2007). We do not consider these in our comparative evaluation since these methods require rated emotion labels and emoticon classes respectively.

Lexicon generation, relies on the availability of a labelled corpus from which the word-emotion distributions can be discovered. For this purpose we exploit a distance-supervised approach where indirect cues are used to unearth implicit (or distant) labels that are contained in the corpus (Alec Go and Huang, 2009). We adopt the approach as in (Wang et al., 2012) to corpus labelling where social media content, and in particular Twitter content is sampled for a predefined set of hashtag cues (P. Shaver, 1987) . Here each set of cues represent a given emotion class. Distant-supervision is particularly suited to Twitter-like platforms because people use hashtags to extensively convey or emphasis the emotion behind their tweets (e.g., That was my best weekend ever.#happy!! #satisfied!). Also given that tweets are length restricted (140 characters), modelling the emotional orientation of words in a Tweet is easier compared to longer documents that are likely to capture complex and mixed emotions. This simplicity and access to sample data has made Twitter one of the most popular domains for emotion analysis research (Wang et al., 2012; Qadir and Riloff, 2013).

3 Problem Definition

We now outline the problem formally. We start with a set of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ where each document d_i has an associated label C_{d_i} indicating the emotion class to which d_i belongs. We consider the case where the documents are tweets. For example, a tweet d_i *nice sunday*

#awesome may have a label *joy* indicating that the tweet belongs to the *joy* emotion class. We also assume that the labels C_{d_i} come from a pre-defined set of six emotion classes *anger, fear, joy, sad, surprise, love*. Since our techniques are generic and do not depend on the number of emotion classes, we will denote the emotion classes as $\{C_j\}_{j=1}^{N}$. Let there be K words extracted from the training documents, denoted as $\{w_i\}_{i=1}^{K}$. Our task is to derive a lexicon Lex that quantifies the emotional valence of words (from the tweets in $\mathcal{D}$) to emotion classes. In particular, the lexicon may be thought of as a 2d-associative array where $Lex[w][c]$ indicates the emotional valence of the word w to the emotion class c. When there is no ambiguity, we will use $Lex(i, j)$ to refer to the emotional valence of word w_i to the emotion class C_j. We will quantify the goodness of the lexicons that are generated using various methods by measuring their performance in an emotion classification task.

4 Lexicon Generation Methods

We now outline the various methods for lexicon generation. We first start off with a simple technique for learning lexicons based on just term frequencies (which we will later use as a baseline technique), followed by more sophisticated methods that are based on conceptual models on how tweets are generated.

4.1 Term Frequency based Lexicon

A simple way to measure the emotional valence of the word w_i to the emotion class C_j is to compute the probability of occurrence of w_i in a tweet labelled as C_j, normalized by its probability across all classes. This leads to:

$$Lex(i,j) = \frac{p(w_i|C_j)}{\sum_{k=1}^{N} p(w_i|C_k)} \tag{1}$$

where the conditional probability is simply computed using term frequencies.

$$p(w_i|C_j) = \frac{freq(w_i, C_j)}{freq(C_j)} \tag{2}$$

where $freq(w_i, C_j)$ is the *number of times w_i occurs in documents labeled with class C_j*. $freq(C_j)$ is the *total number of documents in C_j*.

4.2 Iterative methods for Lexicon Generation

The formulation in the previous section generates a word-emotion matrix L by observing the term frequencies within a class. However term frequencies alone do not capture the term-class associations, because not all frequently occurring terms exhibit the characteristics of a class. For example, a term *sunday* that occurs in a tweet *nice sunday #awesome* labelled *joy* is evidently not indicative of the class *joy*; however, the frequency based computation increments the weight of *sunday* wrt the class *joy* by virtue of this occurrence. In the following sections, we propose generative models that seek to remedy such problems of the simple term frequency based lexicon.

4.2.1 Generative models for Documents

As discussed above, though a document is labelled with an emotion class, not all terms relate strongly to the labelled emotion. Some documents may have terms conveying a different emotion than what the document is labelled with, since the label is chosen based on the most prominent emotion in the tweet. Additionally, some words could be emotion-neutral (e.g., *sunday* in our example tweet) and could be conveying non-emotional information. We now describe two generative models that account for such considerations, and then outline methods to learn lexicons based on them.

Mixture of Classes Model: Let L_{C_k} be the unigram language model (Liu and Croft, 2005) that expresses the lexical character for the emotion class C_k; though microblogs are short text fragments, language modeling approaches have been shown to be effective in similarity assesment between them (Deepak and Chakraborti, 2012). We model a document d_i to be generated from across the emotion class language models:

1. For each word w_j in document d_i,
   (a) Lookup the unit vector $[\lambda_{d_{ij}}^{(1)}, \ldots, \lambda_{d_{ij}}^{(N)}]$; This unit vector defines a probability distribution over the language models.
   (b) Choose a language model L from among the K LMs, in accordance with the vector
   (c) Sample w_j in accordance with the multinomial distribution L

If d_i is labelled with the emotion class C_{d_i}, it is likely that the value of $\lambda_{d_{ij}}^{(n)}$ is high for words in d_i since it is likely that majority of the words are sampled from the $L_{C_{d_i}}$ language model. The posterior probability in accordance with this model can then be intuitively formulated as:

$$P(d_i, C_{d_i}|\theta) = \prod_{w_j \in d_i} \sum_{x=1}^{N} \lambda_{d_{ij}}^{(x)} \times L_{C_x}(w_j) \quad (3)$$

where θ is the parameters $\{L_{C_j}\}_{j=1}^{N}$, λ and C_{d_i} is the class label for document d_i.

Class and Neutral Model: We now introduce another model where the words in a document are assumed to be sampled from either the language model of the corresponding (i.e., labelled) emotion class or from the *neutral language model*, L_C. Thus, the generative model for a document d_i labelled with emotion class C_{d_i} would be as follows:

1. For each word w_j in document d_i,
   (a) Lookup the weight $\mu_{d_{ij}}$; this parameter determines the mix of the labelled emotion class and the neutral class, for w_j in d_i
   (b) Choose L_{C_k} with a probability of $\mu_{d_{ij}}$, and L_C with a probability of $1.0 - \mu_{d_{ij}}$
   (c) Sample w_j in accordance with the multinomial distribution of the chosen language model

The posterior probability in accordance with this model can be intuitively formulated as :

$$P(d_i, C_{d_i}|\theta) = \prod_{w_j \in d_i} \mu_{d_{ij}} \times L_{C_{d_i}}(w_j) + (1 - \mu_{d_{ij}}) \times L_C(w_j) \quad (4)$$

where θ is the parameters $\{L_{C_j}\}_{j=1}^{N}, L_C, \mu$.

Equation 3 models a document to exhibit characteristics of many classes with different levels of magnitude. Equation 4 models a document to be a composition of terms that characterise one class and other general terms; a similar formulation where a document is modeled using a mix of two models has been shown to be useful in characterizing problem-solution documents (Deepak et al., 2012; Deepak and Visweswariah, 2014). The central idea of the expectation maximization (EM) algorithm is to maximize the probability of the data, given the language models $\{L_{C_j}\}_{j=1}^{N}$ and L_C. The term weights are estimated from the language models (E-step) and the language models are re-estimated (M-step) using the term weights from the E-step. Thus the maximum likelihood estimation process in EM alternates between the E-step and the M-step. In the following sections we detail the EM process for the two generative models separately. We compare and contrast the two variants of the EM algorithm in Table 1.

4.2.2 EM with Mixture of Classes Model

We will use a matrix based representation for the language model and the lexicon, to simplify the illustration of the EM steps. Under the matrix notation, $L^{(p)}$ denotes the $K \times N$ matrix at the p^{th} iteration where the i^{th} column is the language model corresponding to the i^{th} class, i.e., L_{C_i}. The p^{th} E-step estimates the various $\lambda_{d_{ij}}$ vectors for all documents based on the language models in $L^{(p-1)}$, whereas the M-step re-learns the language models based on the λ values from the E-step. The steps are detailed as follows:

E-Step: The $\lambda_{d_{ij}}^{(n)}$ is simply estimated to the fractional support for the j^{th} word in the i^{th} document (denoted as w_{ij}) from the n^{th} class language model:

$$\lambda_{d_{ij}}^{(n)} = \frac{L_{C_n}^{(p-1)}(w_{ij})}{\sum_x L_{C_x}^{(p-1)}(w_{ij})} \quad (5)$$

M-Step: As mentioned before in Table 1 this step learns the language models from the λ estimates of the previous step. As an example, if a word w is estimated to have come from the *joy* language model with a weight (i.e., λ) 0.5, it would contribute 0.5 as its count to the *joy* language model. Thus, every occurrence of a word is split across language models using their corresponding λ estimates:

$$L_{C_n}^{(p)}[w] = \frac{\sum_i \sum_j I(w_{ij} = w) \times \lambda_{d_{ij}}^{(n)}}{\sum_i \sum_j \lambda_{d_{ij}}^{(n)}} \quad (6)$$

where the indicator function $I(w_{ij} = w)$ evaluates to 1 if $w_{ij} = w$ is satisfied and 0 otherwise.

After any M-Step, the lexicon can be obtained by normalizing the $L^{(p)}$ language models so that the weights for each word adds up to 1.0. i.e.,

$$Lex^{(p)}(i,j) = \frac{L_{C_j}^{(p)}[w_i]}{\sum_{x=1}^{K} L_{C_x}^{(p)}[w_i]} \quad (7)$$

In the above equation, the suffix (i, j) refers to the i^{th} word in the j^{th} class, confirming to our 2d-array representation of the language models.

Table 1: EM Algorithm variants

States	EM with mixture of classes model	EM with class and neutral model
INPUT	Training data T	Training data T
OUTPUT	Word-Emotion Lexicon	Word-Emotion Lexicon
Initialisation	Learn the initial language models $\{L_{C_j}\}_{j=1}^{N}$	Learn the initial language models $\{L_{C_j}\}_{j=1}^{N}$ and L_C
Convergence	While not converged or #Iterations $< \delta$, a threshold	While not converged or #Iterations $< \delta$, a threshold
E-step	Estimate the $\lambda_{d_{ij}}s$ based on the current estimate of $\{L_{C_j}\}_{j=1}^{N}$ (Sec 4.2.2)	Estimate $\mu_{d_{ij}}$ based on the current estimate of $\{L_{C_j}\}_{j=1}^{N}$ and L_C (Sec 4.2.3)
M-step	Estimate the language models $\{L_{C_j}\}_{j=1}^{N}$ using $\lambda_{d_{ij}}s$ (Sec 4.2.2)	Estimate the language models $\{L_{C_j}\}_{j=1}^{N}$ and L_C using $\mu_{d_{ij}}$ (Sec 4.2.3)
Lexicon Induction	Induce a word-emotion lexicon from $\{L_{C_j}\}_{j=1}^{N}$ (Sec 4.2.2)	Induce a word-emotion lexicon from $\{L_{C_j}\}_{j=1}^{N}$ and L_C (Sec 4.2.3)

4.2.3 EM with Class and Neutral Model

The main difference in this case, when compared to the previous is that we need to estimate a neutral language model L_C in addition to the class specific models. We also have fewer parameters to learn since the $\mu_{d_{ij}}$ is a single value rather than a vector of N values as in the previous case.

E-Step: $\mu_{d_{ij}}$ is estimated to the relative weight of the word w_{ij} from across the language model of the corresponding class, and the neutral model:

$$\mu_{d_{ij}} = \frac{L_{C_{d_i}}^{(p-1)}(w_{ij})}{L_{C_{d_i}}^{(p-1)}(w_{ij}) + L_C^{(p-1)}(w_{ij})} \quad (8)$$

Where C_{d_i} denotes the class corresponding to the label of the document d_i.

M-Step: In a slight contrast from the M-Step for the earlier case as shown in Table 1, a word estimated to have a weight (i.e., μ value) of 0.2 would contribute 20% of its count to the corresponding class' language model, while the remaining would go to the neutral language model L_C. Since the class-specific and neutral language models are estimated differently, we have two separate equations:

$$L_{C_n}^{(p)}[w] = \frac{\sum_{i,label(d_i)=C_n} \sum_j I(w_{ij} = w) \times \mu_{d_{ij}}}{\sum_{i,label(d_i)=C_n} \sum_j \mu_{d_{ij}}} \quad (9)$$

$$L_C^{(p)}[w] = \frac{\sum_i \sum_j I(w_{ij} = w) \times (1.0 - \mu_{d_{ij}})}{\sum_i \sum_j (1.0 - \mu_{d_{ij}})} \quad (10)$$

where $label(d_i) = C_n$ As is obvious, the class-specific language models are contributed to by the documents labelled with the class whereas the neutral language model has contributions from all documents. The normalization to achieve the lexicon is exactly the same as in the mixture of classes case, and hence, is omitted here.

4.2.4 EM Initialization

In the case of iterative approaches like EM, the initialization is often considered crucial. In our case, we initialize the unigram class language models by simply aggregating the scores of the words in tweets labelled with the respective class. Thus, the *joy* language model would be the initialized to be the maximum likelihood model to explain the documents labelled *joy*. In the case of the *class and neutral* generative model, we additionally build the neutral language model by aggregating counts across all the documents in the corpus (regardless of what their emotion label is).

5 Experiments

In this section we detail our experimental evaluation. We begin with the details about the Twitter data used in our experiments. We then discuss how we created the folds for a cross validation experiment. Thereafter we detail the classification task used to evaluate the word-emotion lexicon. Finally we discuss the performance of our proposed methods for lexicon generation in comparison with other manually crafted lexicons, PMI based method for lexicon generation and the standard BoW in an emotion classification task.

5.1 Twitter Dataset

The data set used in our experiments was a corpus of emotion labelled tweets harnessed by (Wang et al., 2012). The data set was available in the form of tweet ID's and the corresponding emotion label. The emotion labels comprised namely : *anger, fear, joy, sadness, surprise, love and thankfulness*. We used the Twitter search API[1] to obtain the tweets by searching with the corresponding tweet ID. After that we decided to consider only tweets that belong to the primary set of emotions defined by Parrott (Parrott, 2001). The emotion classes in our case included *anger, fear, joy, sadness, surprise and love*. We had a collection of 0.28 million tweets which we used to carry out a 10 fold corss-validation experiment.

We decided to generate the folds manually,in order to compare the performance of the different algorithms used in our experiments. We split the collection of 0.28 million tweets into 10 equal size sets to generate 10 folds with different training and test sets in each fold. Also all the folds in our experiments were obtained by stratified sampling, ensuring that we had documents representing all the classes in both the training and test sets. We used the training data in each fold to generate the word-emotion lexicon and measured the performance of it on the test data in an emotion classification task. Table 2 shows the average distribution of the different classes namely: *anger, fear, joy, sadness, surprise and love* over the 10 folds. Observe that emotions such as *joy* and *sadness* had a very high number of representative documents . Emotions such as *anger*,*love* and *fear* were the next most represented emotions. The emotion *surprise* had very few representative documents compared to that of the other emotions.

5.2 Evaluating the word-emotion lexicon

We adopted an emotion classification task in order to evaluate the quality of the word-emotion lexicon generated using the proposed methods. Also research in emotion analysis of text suggest that lexicon based features were effective compared to that of n-gram features in an emotion classification of text (Aman and Szpakowicz, 2008; Mohammad, 2012a). Therefore we decided to use the lexicon to derive features for text representation. We followed a similar procedure as in (Mohammad, 2012a) to define integer valued features for text representation. We define one feature for each emotion to capture the number of words in a training/test document that are associated with the corresponding emotion. The feature vector for a training/test document was constructed using the word-emotion lexicon. Given a training/test document d we construct the corresponding feature vector $d^{'} =< count(e_1), count(e_2), \ldots, count(e_m)) >$ of length m (in our case m is 6), wherein $count(e_i)$ represents the *number of words* in d that exhibit emotion e_i. $count(e_i)$ is computed as:

$$count(e_i) = \sum_{w \in d} I(\max_{j=1,\ldots,m} Lex(w,j) = C_i) \tag{11}$$

where $I(\ldots)$ is the indicator function as used previously. For example if a document has 1 joy word, 2 love words and 1 surprise word the feature vector for the document would be $(0, 0, 1, 0, 1, 2)$. We used the different lexicon generation methods discussed in sections 4.1, 4.2.2 and 4.2.3 to construct the feature vectors for the documents. In the case of the lexicon generated as in section 4.2.3 the max in equation 11 is computed over $m + 1$ columns. We also used the lexicon generation method proposed in (Mohammad, 2012a) to construct the feature vectors. PMI was used in (Mohammad, 2012a) to generate a word-emotion lexicon which is as follows :

$$Lex(i,j) = \log \frac{freq(w_i, C_j) * freq(\neg C_j)}{freq(C_j) * freq(w_i, \neg C_j)} \tag{12}$$

Table 2: Average distribution of emotions across the folds

Emotion	Training	Test
Anger	58410	6496
Fear	13692	1548
Joy	74108	8235
Sadness	63711	7069
Surprise	2533	282
Love	31127	3464
Total	243855	27095

[1] https://dev.twitter.com/docs/using-search

where $freq(w_i, C_j)$ is the number of times n-gram w_i occurs in a document labelled with emotion C_j, $freq(w_i, \neg C_j)$ is the number of times n-gram w_i occurs in a document not labelled with emotion C_j. $freq(C_j)$ and $freq(\neg C_j)$ are the number of documents labelled with emotion C_j and $\neg C_j$ respectively.

Apart from the aforementioned automatically generated lexicons we also used manually crafted lexicons such as WordNet Affect (Strapparava and Valitutti, 2004) and the NRC word-emotion association lexicon (Saif M. Mohammad, 2013) to construct the feature vectors for the documents. Unlike the automatic lexicons, the general purpose lexicons do not offer numerical scores. Therefore we looked for presence/absence of words in the lexicons to obtain the feature vectors. Furthermore we also represented documents in the standard BoW representation. We performed feature selection using the metric Chisquare[2], to select the top 500 features to represent documents. Since tweets are very short we incorporated a binary representation for BoW instead of term frequency. For classification we used a multiclass SVM classifier [3] and all the experiments were conducted using the data mining software Weka[2]. We used standard metrics such as Precision, Recall and F-measure to compare the performance of the different algorithms. In the following section we analyse the experimental results for TF-lex (Sec 4.1), EMallclass-lex (Sec 4.2.2), EMclass-corpus-lex (Sec 4.2.3), PMI-lex (Mohammad, 2012a), WNA-lex (Strapparava and Valitutti, 2004), NRC-lex (Saif M. Mohammad, 2013) and BoW in an emotion classification task. Also in the case of EM based methods we experimented with different threshold limits δ shown in Table 1. We report the results only w.r.t $\delta = 1$ due to space limitations.

5.3 Results and Analysis

Table 3 shows the F-scores obtained for different methods for each emotion. Observe that the F-score for each emotion shown in Table 3 for a method is the average F-score obtained over the 10 test sets (one per fold). We carried a two tail paired t-test[4] between the baselines and our proposed methods to measure statistical significance for performance on the test set in each fold. From the t-test we observed that our proposed methods are statistically significant over the baselines with a confidence of 95% (i.e with p value 0.05). Also note that the best results obtained for an emotion are highlighted in bold. It is evident from the results that the manually crafted lexicons Worndnet Affect and the NRC word-emotion association lexicon are significantly outperformed by all the automatically generated lexicons for all emotions. Also the BoW model significantly outperforms the manually crafted lexicons suggesting that these lexicons are not sufficiently effective for emotion mining in a domain like Twitter.

When compared with BoW the PMI-lex proposed by (Mohammad, 2012a) achieves a 2% gain w.r.t emotion *love*, a 0.6% gain w.r.t emotion *joy* and 1.28% gain w.r.t emotion *sadness*. However in the case of emotions such as *fear* and *surprise* BoW achieves significant gains of 11.17% and 20.96% respectively. The results suggest that the PMI-lex was able to leverage the availability of adequate training examples to learn the patterns about emotions such as *anger, joy, sadness* and *love*. However given that not all emotions are widely expressed a lexicon generation method that relies heavily on abundant training data could be ineffective to mine less represented emotions.

Now we analyse the results obtained for the lexicons generated from our proposed methods and compare them with BoW and PMI-lex. From the results obtained for our methods in Table 3 it suggests that our methods achieve the best F-scores for 4 emotions namely *anger, fear, sadness and love* out of the 6 emotions. In particular the EM-class-corpus-lex method obtains the best F-score for 3 emotions namely *anger, sadness and love*. When compared with BoW and PMI-lex, EM-class-corpus-lex obtains a gain of 0.85% and 0.93% respectively w.r.t emotion *anger*, 1.85% and 0.57% respectively w.r.t emotion *sadness*, 18.67% and 16.88% respectively w.r.t emotion *love*. Our method TF-lex achieves a gain of 5.47% and 16.64% respectively over BoW and PMI-lex w.r.t emotion *fear*. Furthermore w.r.t emotion *surprise* all our proposed methods outperform PMI-lex. However BoW still obtains the best F-score for emotion *surprise*.

When we compared the results between our own methods EM-class-corpus-lex obtains the best F-scores for emotions *anger, joy, sadness and love*. We expected that modelling a document

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

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

[4] http://office.microsoft.com/en-gb/excel-help/ttest-HP005209325.aspx

Table 3: Emotion classification results

Method	Average F-Score					
	Anger	**Fear**	**Joy**	**Sadness**	**Surprise**	**Love**
Baselines						
WNA-lex	25.82%	6.61%	12.94%	8.76%	0.76%	2.67%
NRC-lex	21.37%	3.97%	16.04%	8.87%	1.54%	7.22%
Bow	56.5%	13.56%	63.34%	50.57%	**21.65**%	20.52%
PMI-lex	56.42%	2.39%	**63.4**%	50.57%	0.69%	22.31%
Our Learnt Lexicons						
TF-lex	55.85%	**19.03**%	62.01%	50.54%	11.29%	37.69%
EMallclass-lex	56.64%	14.53%	61.89%	50.48%	12.33%	38.13%
EMclass-corpus-lex	**57.35**%	16.1%	62.74%	**51.14**%	12.05%	**39.19**%

to exhibit more than one emotion (EM-allclass-lex) would better distinguish the class boundaries. However given that tweets are very short it was observed that modelling a document as a mixture of emotion terms and general terms (EM-class-corpus-lex) yielded better results. However we expect EM-allclass-lex to be more effective in other domains such as blogs, discussion forums wherein the text size is larger compared to tweets.

Table 4 summarizes the overall F-scores obtained for the different methods. Note that the F-scores shown in Table 4 are the average overall F-scores over the 10 test sets. Again we conducted a two tail paired t-test[4] between the baselines and our proposed methods to measure the performance gains. It was observed that all our proposed methods are statistically significant over the baselines with a confidence of 95% (i.e with p value 0.05). In Table 4 we italicize all our best performing methods and highlight in bold the best among them. From the results it is evident that our proposed methods obtain significantly better F-scores over all the baselines with EM-class-corpus achieving the best F-score with a gain of 3.21%, 2.9%, 39.03% and 38.7% over PMI-lex, BoW, WNA-lex and NRC-lex respectively. Our findings reconfirm previous findings in the literature that emotion lexicon based features improve over corpus based n-gram features in a emotion classification task. Also our findings suggest that domain specific automatic lexicons are significantly better over manually crafted lexicons.

Table 4: Overall F-scores

Method	**Avg Overall F-score**
Baselines	
WNA-lex	13.17%
NRC-lex	13.50%
Bow	49.30%
PMI-lex	48.99%
Our automatic lexicons	
TF-lex	*51.45%*
EMallclass-lex	*51.38%*
EMclass-corpus-lex	**52.20**%

6 Conclusions and Future Work

We proposed a set of methods to automatically extract a word-emotion lexicon from an emotion labelled corpus. Thereafter we used the lexicons to derive features for text representation and showed that lexicon based features significantly outperform the standard BoW features in the emotion classification of tweets. Furthermore our lexicons achieve significant improvements over the general purpose lexicons and the PMI based automatic lexicon in the classification experiments. In future we intend to leverage the lexicons to design different text representations and also test them on emotional content from other domains. Automatically generating human-interpretable models (e.g., (Balachandran et al., 2012)) to accompany emotion classifier decisions is another interesting direction for future work.

References

Richa Bhayani Alec Go and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Processing*.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceed-*

ings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, pages 579–586, Stroudsburg, PA, USA. Association for Computational Linguistics.

S. Aman and S. Szpakowicz. 2008. Using roget's thesaurus for fine-grained emotion recognition. In *International Joint Conference on Natural Language Processing*.

Vipin Balachandran, Deepak P, and Deepak Khemani. 2012. Interpretable and reconfigurable clustering of document datasets by deriving word-based rules. *Knowl. Inf. Syst.*, 32(3):475–503.

Johan Bollen, Alberto Pepe, and Huina Mao. 2009. Modelling public mood and emotion : Twitter sentiment and socio-economic phenomena. In *CoRR*.

Danah Boyd, Scott Golder, and Gilad Lotan. 2010. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences, Washington, DC, USA*.

P. Deepak and Sutanu Chakraborti. 2012. Finding relevant tweets. In *WAIM*, pages 228–240.

P. Deepak and Karthik Visweswariah. 2014. Unsupervised solution post identification from discussion forums. In *ACL*.

P. Deepak, Karthik Visweswariah, Nirmalie Wiratunga, and Sadiq Sani. 2012. Two-part segmentation of text documents. In *CIKM*, pages 793–802.

Lars E.Holzman and Pottenger William M. 2003. Classification of emotions in internet chat : An application of machine learning using speech phonemes. Technical report, Technical report, Leigh University.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6(3):169–200.

Virginia Francisco and Pablo Gervas. 2006. Automated mark up of affective information in english text. *Text, Speech and Dialouge*, volume 4188 of Lecture Notes in Computer Science:375–382.

David John, Anthony C. Boucouvalas, and Zhe Xu. 2006. Representing emotinal momentum within expressive internet communication. In *In Proceedings of the 24th IASTED international conference on Internet and multimedia systems and applications, pages 183-188, Anaheim, CA, ACTA Press*.

J. Johnson J. Guthrie K. Roberts, M.A. Roach and S.M. Harabagiu. 2012. "empatweet: Annotating and detecting emotions on twitter",. In *in Proc. LREC, 2012, pp.3806-3813*.

Elsa Kim, Sam Gilbert, J.Edwards, and Erhardt Graeff. 2009. Detecting sadness in 140 characters: Sentiment analysis of mourning of michael jackson on twitter.

Xiaoyong Liu and W Bruce Croft. 2005. Statistical language modeling for information retrieval. Technical report, DTIC Document.

Chunling Ma, Helmut Prendinger, and Mitsuru Ishizuka. 2005. Emotion estimation and reasoning based on affective textual interaction. In *First International Conference on Affective Computing and Intelligent Interaction (ACII-2005), pages 622-628, Beijing, China*.

Rada Mihalcea and Hugo Liu. 2006. A corpus-based approach for finding happiness. In *In AAAI-2006 Spring Symposium on Computational Approaches to Analysing Weblogs, pages 139-144. AAAI press*.

Saif M. Mohammad and Tony Yang. 2011. Tracking seniment in mail : How genders differ on emotional axes. In *In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis(WASSA 2011), pages 70- 79, Portland, Oregon. Association for Computational Linguistics*.

Saif Mohammad. 2012a. #emotional tweets. In *The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*.

Saif M. Mohammad. 2012b. Portable features for classifying emotional text. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 587-591, Montreal , Canada*.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2010. Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 806–814, Stroudsburg, PA, USA. Association for Computational Linguistics.

D. Kirson P. Shaver, J. Schwartz. 1987. Emotion knowledge: Further exploration of a prototype approach. *Journal of Personality and Social Psychology*, Vol 52 no 6:1061 – 1086.

W Parrott. 2001. Emotions in social psychology. *Psychology Press, Philadelphia*.

Lisa Pearl and Mark Steyvers. 2010. Identifying emotions, intentions and attitudes in text using a game with a purpose. In *In Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, Los Abgeles, California*.

R. Plutchik. 1980. A general psychoevolutionary theory of emotion. *In R. Plutchik & H. Kellerman (Eds.), Emotion: Theory, research, and experience:*, Vol. 1. Theories of emotion (pp. 3-33). New York: Academic:(pp. 3–33).

Ashequl Qadir and Ellen Riloff. 2013. Bootstrapped learning of emotion hashtahs #hashtags4you. In *In the 4th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA 2013)*.

Tapas Ray. 2011. The 'story' of digital excess in revolutions of the arab spring. *Journal of Media Practice*, 12(2):189–196.

Peter D. Turney Saif M. Mohammad. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence, 29 (3), 436-465, Wiley Blackwell Publishing Ltd, 2013*, 29(3):436–465.

Prendinger H. Ishizuka M. Shaikh, M.A.M., 2009. *A Linguistic Interpretation of the OCC Emotion Model for Affect Sensing from Text*, chapter 4, pages 45–73.

Julia Skinner. 2011. Social media and revolution: The arab spring and the occupy movement as seen though three information studies paradigms. *Sprouts: Working papers on Information Systems*, 11(169).

Carlo Strapparava and Alessandro Valitutti. 2004. Wordnet-affect: an affective extension of wordnet. Technical report, ITC-irst, Istituto per la Ricerca Scienti?ca e Tecnologica I-38050 Povo Trento Italy.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. 2012. Harnessing twitter "big data" for automatic emotion identification. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE.*

C. Yang, K. H. Y. Lin, and H. H. Chen. 2007. Emotion classification using web blog corpora. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, WI '07, pages 275–278, Washington, DC, USA. IEEE Computer Society.

Liu Wenyin Qing Li Yanghui Rao, Xiaojun Quan and Mingliang Chen. 2013. Building word-emotion mapping dictionary for online news. In *In Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA 2013.*