Emotion Detection from Text via Ensemble Classification Using Word Embeddings

Jonathan Herzig
IBM Research - Haifa
Haifa 31905, Israel
hjon@il.ibm.com

Michal Shmueli-Scheuer
IBM Research - Haifa
Haifa 31905, Israel
shmueli@il.ibm.com

David Konopnicki
IBM Research - Haifa
Haifa 31905, Israel
davidko@il.ibm.com

ABSTRACT

Emotion detection from text has become a popular task due to the key role of emotions in human-machine interaction. Current approaches represent text as a sparse bag-of-words vector. In this work, we propose a new approach that utilizes pre-trained, dense word embedding representations. We introduce an ensemble approach combining both sparse and dense representations. Our experiments include five datasets for emotion detection from different domains and show an average improvement of 11.6% in macro average F1-score.

1 INTRODUCTION

Emotions are an important element of human nature and detecting them in the textual messages written by users has many applications in information retrieval [17] and human-computer interaction [6]. A common approach to emotion analysis and modeling is categorization, e.g., according to Ekman's basic emotions; namely, anger, disgust, fear, happiness, sadness, and surprise [9]. Approaches to categorical emotion classification based on text often employ supervised machine learning classifiers, which require labeled training data.

Currently, two types of datasets labeled with emotions are publicly available: manually labeled, and pseudo-labeled. Manual annotation requires high cognitive capabilities of multiple human annotators per sample. As a result, the quality of these datasets is usually high. However, the task is tedious, time-consuming, and expensive [4], and thus, these datasets are usually small (in the order of thousands of annotated samples). Manual annotations are usually applied to domain specific datasets (e.g., news headlines). To overcome these limitations, pseudo-labeled datasets are gathered from social media platforms where social media posts are explicitly tagged by the author by using the hashtag symbol (#) or by adding emoticons [1]. This tagged data can be used to create large-scale training data labeled with emotions in a non-specific domain as in [20].

Given such a dataset (manually or pseudo labeled), it is then common to train a linear classifier based on bag-of-words (BOW) representation: representing text samples as sparse vectors, where each vector entry corresponds to the presence of a specific feature (such as n-grams, punctuation and other) [21].

As reported recently by [11], deep learning is a promising approach for solving NLP tasks including text classification. While the aforementioned approach utilizes BOW representation and linear classifiers, neural network methods are based on dense vector representations of text samples (word embedding) and are non-linear. Such word embedding representation captures syntactic and semantic knowledge, which can improve the emotion detection task. For example, in such a representation the words "awful" and "terrible" are expected to have similar vector representations.

Generating high quality word vectors requires large-scale data and computing power. When generation is not an option, one can utilize pre-trained representations; the most popular pre-trained representations are based on word2vec [18] and GloVe [23] algorithms, and were trained on large corpora.

Inspired by the good results of the deep learning approach, we aim at using these techniques for emotion detection. Pseudo-labeled large-scale data is suitable for deep learning techniques, however, it exhibits low accuracy when classifying domain specific data, even after domain adaptation [20] (using linear models). On the other hand, the manually annotated data is highly accurate, but it is not large enough to form an appropriate base for deep learning techniques, as these models tend to overfit small size datasets. In order to utilize the high quality but small size datasets, we propose an *ensemble* approach that combines both a linear model based on BOW, and a non-linear model based on the pre-trained word vectors. In addition, we propose a new method for realizing a sentence level representation from the single words vectors.

To our knowledge, this is the first research that shows how to utilize pre-trained word vectors to improve emotion detection from text in domain-specific datasets.

2 RELATED WORK

Approaches to categorical emotion classification often employ one-vs-rest machine learning classifiers (a binary classifier for each emotion), typically SVM [19, 25] or logistic regression [32], using textual features such as n-grams, lexicon based, and POS features.

In the domain of sentiment analysis, which is closely related to emotion detection, Tang et al. [30] showed that encoding sentiment information into the word embeddings using Twitter data, as a part of a neural network based system, outperformed previous approaches. [16, 29] focused on generating document level embeddings in sentiment classification. These works all require the availability of large-scale data. In our setting, we only have small datasets, thus, we used pre-trained vectors.

[1] http://techcrunch.com/2013/04/09/facebook-mood/

ICTIR'17, October 1–4, 2017, Amsterdam, The Netherlands.

DOI: 10.1145/3121050.3121093

Pre-trained word vectors were used as an input to neural networks to improve sentiment analysis classification [14, 27]. This approach also requires large-scale data for the neural network training. Forgues [10] used pre-trained word vectors and a linear classifier to classify user intents in dialog systems, however their task and methodology is different than ours.

3 EMOTION CLASSIFIERS

In this section we describe the two different classifiers we created and an ensemble method that combines their output. The two classifiers are based on different document representations. Depending on the dataset being used, the classification task can be represented as a multi-class or a multi-label problem. For both types of problems we used a one-vs-rest SVM classifier. Thus, given some test sample, a classifier outputs the decision function value for each emotion that appears in the training data. The classes associated with the test sample are then taken to be the emotion with the highest decision function value (for multi-class) or the set of emotions with a positive decision function value (for multi-label).

3.1 BOW Classifier

In our first approach we used an SVM classifier with a linear kernel, and represented every document as a BOW. We extracted various n-grams (after lemmatization), punctuation and social media features. Namely, unigrams, bigrams, NRC lexicon features (number of terms in a post associated with each affect label in NRC lexicon [22]), and presence of exclamation marks, question marks, usernames, links, happy emoticons, and sad emoticons.

We used TFIDF weights for the values of n-gram features (this was experimentally superior to using binary weights), and removed n-grams that appeared in less than two documents. We further removed 10% of the remaining features that got the lowest scores in a χ^2 statistical test. We handled negation similarly to [24].

3.2 Word Embedding-Based Classifier

Word embedding based vectors can be combined to represent a document into a fixed size vector. We have experimented with several document representations, combining the word vectors, following the notation:

$$d_{we}(t_1, ..., t_k) = \frac{1}{\sum_{i=1}^{k} a_i} \sum_{i=1}^{k} a_i \cdot v(t_i), \quad (1)$$

where d_{we} is the word embedding based vector representation for document d with k terms, v is some pre-trained word-to-vector mapping (described in Section 5), and a_i is some weight indicating the relative importance of term t_i. The document representations we experimented with include:

CBOW (continuous bag of words) [18]: in this case, $\forall i, a_i = 1$, which means uniform weights for all terms.

TFIDF weights: a_i is the TFIDF weight for term t_i. In case t_i was not present in the training data, we smoothed its IDF weight as if it appeared in one document (this yielded better performance than discarding the term).

Classifier weights (CLASS): in this approach we calculated a weight function, $w(t, e)$ for each term t in the training data which indicates its importance in classifying a document as expressing emotion e. We did this by first representing the documents in a BOW binary vector representation where we only extracted unigram features. Then, for each e we trained an SVM model with a linear kernel and took $m(e, t)$ to be the weight associated by the model with each term t in the training data. Motivated by Guyon [12] who showed that $|m(e, t)|$ is an indicative feature selection criterion, we define:

$$w(t, e) = \frac{|m(e, t)| - \mu_e}{\sigma_e}, \quad (2)$$

where μ_e and σ_e are the corresponding average and standard deviation of model weights in absolute value. We now define:

$$a_i = \begin{cases} 1 & t_i \notin V \text{ or } w(t_i, e) < 1 \\ w(t_i, e) & \text{else} \end{cases} \quad (3)$$

, where V is the vocabulary generated from the training data. In words, a low constant weight is assigned to terms which did not appear in the training data, or were found to be less discriminative. For other terms, the weight is proportional to the words discriminative power. This method captures the notion that some terms in a document are more important in its embedded representation since they are more informative given the classification task. Note that, in this method, a different document representation is used for every emotion e since the discriminative power of every word (and hence its weight a_i) is different for each emotion. This is a novel embedded document representation method, and as detailed in Section 5, it is superior in comparison to the other representation methods we experimented with.

Similarly to our BOW classifier, we used an SVM classifier in this approach as well, but since we represented a document as a low-dimensional vector, we allowed non-linearity by using an RBF kernel. This yielded better results than using a linear kernel.

3.3 Ensemble Methods

Ensembles tend to achieve better results when there is a significant diversity among the classifiers [15]. Thus, we utilized the above classifiers that are based on different document representations, to form an ensemble model. As a preliminary step, we transformed the above classifiers' decision function value output to represent probabilities, using softmax transformation for multi-class problems [8], and sigmoid transformation for multi-label problems. The ensemble methods we experimented with, follow the notation:

$$m_{en}(d) = \alpha \cdot (m_{bow}(d)) + (1 - \alpha) \cdot (m_{we}(d)), \quad (4)$$

where $m_{en}(d)$ is the output probability vector for the ensemble classifier given a test document d, $m_{bow}(d)$ and $m_{we}(d)$ are the output probability vectors of the BOW and the word embedding based classifiers respectively, and α is a parameter which corresponds to the specific ensemble method used. We have experimented with the following weighed average probabilities methods: equal weights ($\alpha = 0.5$), stacking (α is learned by an additional classifier) and precision-based weighting [3] (α reflects the ratio between the macro precision scores for the two classifiers over the training data). We have found in our setting that precision-based weighting achieved the best performance, thus we report results using this method only.

Dataset	Size	Emotion Classes	Problem	# Annotators
ISEAR	7666	anger, disgust, fear, guilt, joy, sadness, and shame	multi-class	1
SemEval	1250	anger, disgust, fear, happiness, sadness, and surprise	multi-class	6
Fairy Tales	1207	angry-disgusted , fear, happiness, sadness, and surprise	multi-class	6
Blog Posts	4090	anger, disgust, fear, happiness, sadness, surprise, and neutral	multi-class	4
Twitter Dialogs	1056	confusion, frustration, anger, sadness, happiness, hopefulness, disappointment, gratitude, politeness, and neutral	multi-label	5

Table 1: Datasets characteristics.

Dataset	Previous Work	Performance	BOW	Metric
ISEAR	[7]	0.702	**0.712**	accuracy
SemEval	[19]	**0.516**	0.464	macro F1
Fairy Tales	[5]	0.619	**0.691**	accuracy
Blog Posts	[2]	0.739	**0.779**	accuracy
Twitter Dialogs	[13]	0.440	**0.472**	macro F1
Average	-	0.603	**0.624**	-

Table 2: BOW baseline performance comparison with published results using the published metric.

Dataset	Word2Vec			GloVe		
	CBOW	TFIDF	CLASS	CBOW	TFIDF	CLASS
ISEAR	.566	.543	.610	.577	.562	**.620**
SemEval	.477	.486	.482	.495	**.502**	.499
Fairy Tales	.617	.630	.688	.661	.680	**.710**
Blog Posts	.506	.528	.634	.505	.533	**.635**
Twitter Dialogs	.441	.450	.447	.469	**.480**	.479
Average	.521	.528	.572	.541	.551	**.589**

Table 3: Macro F1-scores for each pre-trained word vector source and embedded representation method.

4 DATASETS

We experimented with the following five emotion detection datasets from different domains (summarized in Table 1):

- **ISEAR** [26] contains labeled sentences where participants who have different cultural backgrounds reported experiences and reactions for seven emotions.
- **SemEval** [28] contains newspaper headlines labeled with the six Ekman emotions by six annotators. For our experiments, we considered the most dominant emotion as the headline label as in [19].
- **Fairy Tales** [1] includes sentences from fairy tales, labeled with five emotions by six annotators. For our experiments, we used only sentences with high annotation agreement of four identical emotion labels, as in [5].
- **Blog Posts** [2] consists of emotion-rich sentences collected from blogs labeled with emotions by four annotators. We considered only sentences for which the annotators agreed on the emotion category, as in [2].
- **Customer Support Dialogs in Twitter** [13] consists of customer turns from customer support dialogs in Twitter, labeled by five annotators with nine emotions relevant to customer care.

5 EXPERIMENTS

5.1 Experimental Setup

We tuned the hyper-parameters for our classifiers using a grid search over a validation dataset collected from Twitter, presented in [33]. This process yielded the following hyper-parameter values that we used in all of our experiments below. Penalty parameter for BOW classifier: $C = 0.5$; Penalty and kernel parameters for the word embedding based classifier: $C = 4$, $\gamma = 0.1$.

We evaluated our methods for each dataset by using 10-fold cross-validation. Our baseline is the BOW classifier described above. This was used as a state-of-the-art approach for emotion detection in short texts in many cases, e.g., [19, 31] and more.

Emotion detection datasets are labeled with multiple emotions and are imbalanced. Thus, we evaluated the classification performance for all emotion classes by using macro average F1-score. We used scikit-learn [2] for an SVM implementation and spaCy [3] for n-grams extraction.

5.2 BOW Classifier Comparison

We compared our BOW classifier baseline with previous work which presented cross-validation results on the datasets we experimented with. Table 2 shows results presented in the original work that introduced the dataset, or in an advanced work. The table shows that our BOW classifier results correspond to previous work, which validates our baseline's implementation.

5.3 Word Embedding Methods Comparison

We compared the quality of two publicly available pre-trained word vector sources, based on GloVe[4] and Word2Vec (GoogleNews)[5], in terms of emotion detection performance. Vectors from both sources are 300 dimensional.

Table 3 depicts the macro F1-scores for each dataset and each document representation method that is based on word vectors, as detailed in Section 3.2. Results show that for all datasets and representation methods, word vectors trained by GloVe achieved higher performance than using Word2Vec based vectors. For example, on average, CBOW representation for GloVe source showed a 4% improvement in F1-score relative to Word2Vec. Thus, we used GloVe based pre-trained word vectors below. Also, the CLASS method we

[2] http://scikit-learn.org/stable/
[3] https://spacy.io/
[4] http://nlp.stanford.edu/data/glove.840B.300d.zip
[5] https://code.google.com/archive/p/word2vec/

Dataset	BOW	EN-CBOW	EN-TFIDF	EN-CLASS
ISEAR	0.617	0.626	0.625	**0.641***
SemEval	0.464	0.516	**0.525***	0.517*
Fairy Tales	0.638	0.700	0.729*	**0.733***
Blog Posts	0.560	0.590	0.612*	**0.663***
Twitter Dialogs	0.472	0.512	**0.522***	0.516*
Average	0.550	0.589	0.603	**0.614**

Table 4: Macro F1-scores for all datasets using ensembles. "*" marks a significantly better model ($p < 0.001$) than the baseline model, using McNemar's test.

proposed outperformed the other embedded document representation methods.

5.4 Ensemble Results

Table 4 depicts the macro F1-scores for each dataset, and for the different models: BOW is our baseline, presented in Section 3.1. EN-CBOW, EN-TFIDF and EN-CLASS are the ensemble models of BOW and the corresponding embedded representations presented in Section 3.2.

Our ensemble methods outperformed the baseline for each document representation method. The best result, which is also significantly better for each dataset, is of EN-CLASS model that achieved an average relative improvement of 11.6% in F1-score over all datasets. These results indicate the advantage in combining both BOW and embedded document representations for emotion detection from text.

6 CONCLUSIONS

This work studied the use of pre-trained word vectors for emotion detection. We presented CLASS, a novel method for representing a document as a dense vector based on the importance of the document's terms in respect to emotion classification. Our results show that an ensemble that combines BOW and embedded representations using our CLASS method, outperforms previous approaches for domain-specific datasets. In comparison to other deep-learning methods, our approach fits a small number of model parameters and requires little computing power.

For Future work we plan to investigate the use of deep learning models trained on domain adapted pseudo-labeled large-scale datasets. We also plan to investigate transfer learning for multi-domain emotion detection.

REFERENCES

[1] Ebba Cecilia Ovesdotter Alm. 2008. *Affect in text and speech.* ProQuest.

[2] Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*. Springer, 196–205.

[3] Alina Andreevskaia and Sabine Bergler. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging.. In *ACL*. 290–298.

[4] Lea Canales and Patricio Martinez-Barco. 2014. Emotion Detection from text: A Survey. *Processing in the 5th Information Systems Research Working Days (JISIC 2014)* (2014), 37.

[5] Soumaya Chaffar and Diana Inkpen. 2011. Using a heterogeneous dataset for emotion analysis in text. In *Advances in Artificial Intelligence*. Springer, 62–67.

[6] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. 2001. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE* 18, 1 (2001), 32–80.

[7] Taner Danisman and Adil Alpkocak. 2008. Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, Vol. 1. 53.

[8] Kaibo Duan, S Sathiya Keerthi, Wei Chu, Shirish Krishnaj Shevade, and Aun Neow Poo. 2003. Multi-category classification by soft-max combination of binary classifiers. In *Multiple classifier systems*. Springer, 125–134.

[9] Paul Ekman. 1992. An argument for basic emotions. *Cognition & Emotion* 6, 3-4 (1992), 169–200.

[10] Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. 2014. Bootstrapping Dialog Systems with Word Embeddings. (2014).

[11] Yoav Goldberg. 2017. Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies* 10, 1 (2017), 1–309.

[12] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research* 3 (2003), 1157–1182.

[13] Jonathan Herzig, Guy Feigenblat, Michal Shmueli-Scheuer, David Konopnicki, Anat Rafaeli, Daniel Altman, and David Spivak. 2016. Classifying Emotions in Customer Support Dialogues in Social Media. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 64.

[14] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *CoRR* abs/1408.5882 (2014).

[15] Ludmila I Kuncheva and Christopher J Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning* 51, 2 (2003), 181–207.

[16] Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053* (2014).

[17] Irene Lopatovska and Ioannis Arapakis. 2011. Theories, methods and current research on emotions in library and information science, information retrieval and human–computer interaction. *Information Processing & Management* 47, 4 (2011), 575–592.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[19] Saif Mohammad. 2012. Portable features for classifying emotional text. In *Proceedings of NAACL HLT*. 587–591.

[20] Saif M. Mohammad. 2012. #Emotional Tweets. In *Proceedings of 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 '12)*.

[21] Saif M Mohammad. 2015. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. *Emotion Measurement* (2015).

[22] Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence* 29, 3 (2013), 436–465.

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[24] Nataliia Plotnikova, Micha Kohl, Kevin Volkert, Andreas Lerner, Natalie Dykes, Heiko Ermer, and Stefan Evert. 2015. KLUEless: Polarity classification and association. In *Proceedings of the 9th International Workshop on Semantic Evaluation*. 619–625.

[25] Ashequl Qadir and Ellen Riloff. 2014. Learning Emotion Indicators from Tweets: Hashtags, Hashtag Patterns, and Phrases. In *Proceedings of EMNLP*. 1203–1209.

[26] Klaus R Scherer and Harald G Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology* 66, 2 (1994), 310.

[27] Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. 464–469.

[28] Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, 70–74.

[29] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1422–1432.

[30] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification.. In *ACL (1)*. 1555–1565.

[31] Bincy Thomas, KA Dhanya, and P Vinod. 2014. Synthesized feature space for multiclass emotion classification. In *Networks & Soft Computing (ICNSC), 2014 First International Conference on*. IEEE, 188–192.

[32] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth. 2012. Harnessing Twitter Big Data for Automatic Emotion Identification. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. 587–592.

[33] Yichen Wang and Aditya Pal. 2015. Detecting emotions in social media: a constrained optimization approach. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 996–1002.