2018 2nd Int. Conf. on Innovations in Science, Engineering and Technology (ICISET)
27-28 October 2018, Chittagong, Bangladesh

Emotion Detection from Text Using Skip-thought Vectors

Maruf Hassan
Department of Computer Science and Engineering
International Islamic University Chittagong
Chittagong, Bangladesh
marufxhassan@gmail.com

Md. Sakib Bin Alam
Department of Computer Science and Engineering
International Islamic University Chittagong
Chittagong, Bangladesh
sakibsba@gmail.com

Tanveer Ahsan
Department of Computer Science and Engineering
International Islamic University Chittagong
Chittagong, Bangladesh
tanveer.ahsan@gmail.com

***Abstract*—Emotion detection from natural language has become a popular task because of the primary role of emotions in human-machine interaction. It has a wide variety of applications ranging from developing emotional chatbots to better understanding people and their lives. The problem of finding emotion from text has been handled by using lexical approaches and machine learning techniques. In recent years neural network models have become increasingly popular for text classification. Especially, the emergence of word embeddings within deep learning architectures has recently drawn a high level of attention amongst researchers. In this research, we apply a recently proposed deep learning model named skip-thought, an approach to learning fixed length representations of sentences, to face the problem of emotion detection from text. We propose a new framework that takes advantage of the pre-trained model and pre-trained word vectors. We found that skip-thought vectors are well suited for emotion detection task. The results of the performance evaluation are encouraging and comparable to related research.**

Keywords— emotion detection; machine learning; deep learning; recurrent neural network, word embedding; sentence embedding;

I. Introduction

Emotion is an essential component of human behavior, a complex experience of consciousness that reflects the personal criticalness of a thing, an occasion, or a situation. Emotion detection is the process of recognizing a person's emotional state, e.g. as stated in Ekman's basic emotions; specifically, anger, disgust, fear, happiness, sadness, and surprise [1]. Accurately detecting emotion from natural language has application of developing emotional chatbots which extent to better understanding persons and their behavior, developing intelligent personal assistants to detecting the emotions of social media users to understand their psychological and physical health.

Computer system has not yet utilized emotion detection properly to facilitate humans in performing their assignments. Linguistic knowledge still becomes the focus on the continued efforts in emotion analysis. Language is inherently ambiguous and complex by nature. It is the most important challenges researcher encounter in emotion analysis. Also, presently there aren't any established classifications of "all human emotions" because of the complicated characteristics of human minds.

Deep Learning architectures and algorithms have made outstanding advancements in the field of Natural Language Processing (NLP) in recent years. The idea of the usage of vector representation of words has been there around for a few times. But the interest in word embedding has been hovering in the recent years. Word embedding is a technique that produces vectors and maps them into corresponding words. Tomas Mikolov's word2vec [2] model which is trained on a large collection of corpora produce high-dimensional (50 - 300 dimensional) vector representation corresponding to words that can capture the syntactic and semantic knowledge. This representation seems to capture much linguistic information. In word2vec model, they showed fixed length representations of individual words. However, fixed length representation of sentences was an issue there.

Skip-thought vectors or simply skip-thought [3] is a name given to a simple neural networks model for learning fixed length vector representations of sentences. It was inspired by the skip-gram structure used in word2vec model, abstract the skip-gram model to the sentence level. Fixed length vector representation of sentences makes it easy to replace any sentence with equivalent vectors of numbers. This makes the process of understanding and responding to natural language mathematically straightforward.

To get good performance in these techniques, availability of large-scale data is needed to train. But, in our placing, we have only small datasets. Deep learning model is incapable of giving good result on small datasets. That's why we make the decision to take a pre-trained model. We propose a framework that utilizes the pre-trained model for finding a sentence level feature representation using the word vectors and evaluate how it performs well for emotion detection task.

The rest of this paper is structured as follows. Section II discusses related work. Section III provides a brief introduction to the skip-thought model that we used for our proposed framework. Section IV introduces our proposed approach to emotion detection. Section V carries the two data sets. We report on the experiments and results in Section VI and conclude the paper in Section VII.

II. Related Work

Emotion detection is closely connected to sentiment analysis, they are often used interchangeably. By proposing the

978-1-5386-8524-2/18/$31.00 ©2018 IEEE

role of emotions in human-computer interaction, the concept of affective computing was introduced by Picard [4].

In our literature review, we face two approaches that have been occurred frequently to face the problem of emotional text analysis task. Lexicon-based methods use a lexicon of affective words or phrases to detect emotions in text. The primary focus of lexical-based methods has been on the employment of lexicons of emotion words. The second one depends on machine learning tools and technologies.

A. Lexicon-based Approach

Lexicon-based method is a straightforward method and considerably easier to implement than other methods. In this method, each emotional word carries emotional valences. These valences are calculated to get the emotional context in the whole document [5]. The capability of lexicon-based systems relies on the coverage and quality of the chosen emotion lexicon [6] [7]. Lexical-based method ends up in poor performance and generalization during unavailability of emotional words in text [8]. Also, it showed bad performance in handling more complex linguistic structures of text.

B. Learning-based Approach

Learning based methods are being employed to face the problem differently. Machine learning algorithms use computational methods that depend on various classifiers. Performance comparison of four different classifiers (Bayesian, Random Forest, Logistic Regression, and SVM) can be found in [9]. One of the major problems of learning-based approach is that it results in blurred borders between emotion categories and an absence of context analysis. They are domain (dataset topic) dependent and datasets collected at different time periods require different features. Also, the classifier needs to be trained on large feature sets. As a result, long training sessions are needed and because of the enormous number of features, the classifiers take much more time and memory to produce result.

C. Deep Learning for NLP

Neural network [10] has come out as an undefeated category of techniques that has the ability of spontaneously finding the representations required for recognition or classification and has been efficiently carried out a couple of NLP assignments. Named entity recognition (NER), a part of speech (POS) tagging or sentiment analysis are a few the issues wherever neural network models have outperformed ancient approaches. A line of research [11] [12] [13] within the literature target to learn sentiment-specific word and phrase embeddings from neighboring textual content. On the other hand, some analysis concentrates on learning linguistics composition, as well as they extend to phrases and sentences using recurring neural networks [14] [15]. There exist two types of Recurrent Neural Networks (RNN), Gated Recurrent Unit (GRU) [16] [17] and Long-short term memory (LSTM) [18] for managing sequential data like speech or character recognition. They are also being used effectively for task such as sentiment analysis [13] [19].

To generate high-quality word vectors from domain-specific dataset, we need a large volume of data and heavy computation resources. All the datasets available for emotion detection task are not large enough to train for good vectors. Skip-thought model has already generated high-quality word vectors [3]. Inspired by the good results of the skip-thought vectors on different tasks such as semantic relatedness, sentiment analysis etc., we intent to use this technique for our emotion detection task.

One of the previous researches showed how to utilize a pre-trained model to improve the performance of emotion detection task [20]. Our research is different from their work in a way that they focused on word embedding as a feature vector. In our work, we try to apply recurrent neural network (RNN) based sentence embedding for emotion detection task as it preserves much more context.

III. Skip-thought Model Overview

In this section, we give a brief introduction to the skip-thought model that we used for our proposed framework. We describe how skip-thought model was developed and how the skip-thought vectors were generated.

Let's say, we are given three sequential sentences (s_{i-1}, s_i, s_{i+1}). Here s_i is the i-th sentence from the corpus.

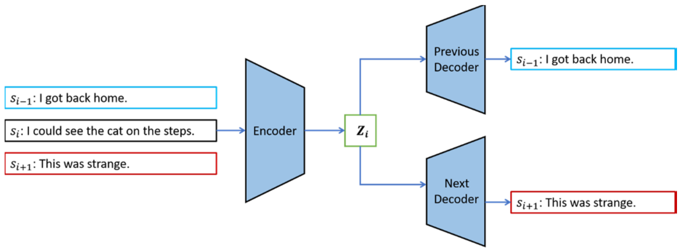

Fig. 1. Skip-thought Model Overview.

In general, the encoder generates the fixed-length representation of the input sentence s_i and gives the embedding z_i. The decoders are trained to reduce the reconstruction error of the preceding sentence s_{i-1} and succeeding sentence s_{i+1} given the embedding z_i. This reconstruction error is back-propagated to the encoder which maps the relationship of the input sentence to its surrounding sentence. This mapping is shown in Figure 1.

The end product of skip-thought model is the encoder because it captures both syntactic and semantic properties. The decoders are thrown away after training. The lookup table is shared among encoder and decoders allows high-quality word embeddings for not only input sentences, but also their context. After training, the trained encoder along with the lookup table can then be used as a generic feature extractor for new input sentences.

IV. Proposed Framework for Emotion Detection

In this section, we give the detailed description of our proposed framework. Our objective is to entitle appropriate emotion label to a text document based on its context. The overview of our proposed framework for emotion detection task is depicted in Figure 2.

In the beginning, our system fetches all the text from the user and queue them for further processing Next stages are described in the following.

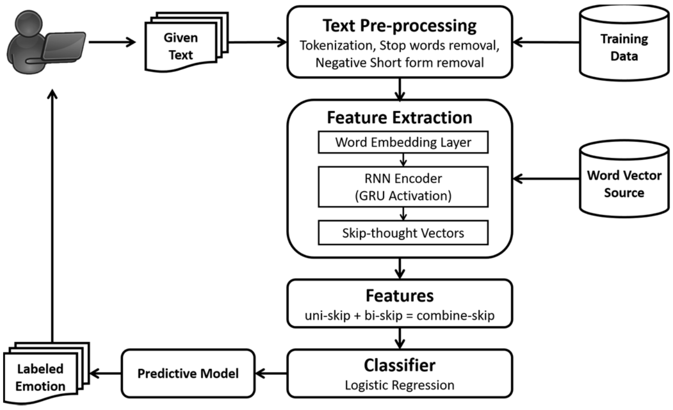

Fig. 2. Our proposed emotion detection framework

A. Text Preprocessing

Since our goal is to evaluate how skip-thought vectors perform well for emotion detection task, we keep text preprocessing to a minimum. To encode a new sentence, we applied simple word tokenization, stop words removal and conversion of negative short form to long form e.g. "won't" is replaced by "will not". This is carried out to check the effectiveness of the skip-thought vectors for emotion detection task. We used nltk[1] library, a natural language toolkit, for text preprocessing.

B. Feature Extraction

We embed the trained encoder from the skip-thought model along with the lookup table layer as depicted in Figure 3. We use this layer as feature extractor for our emotion detection task.

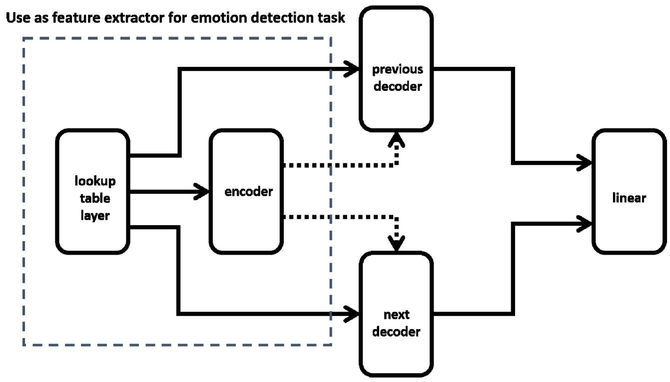

Fig. 3. Feature extraction from the skip-thought model for our framework.

The goal is to generate a vector that summarizes the whole input text. To do this, each word in the input sentence is represented as a one-hot encoded vector. The encoder linearly projects the one-hot encoded vector w_i with a parameter matrix E. This matrix is initialized from the lookup table. This projection $s_i = \mathrm{E}w_i$ (s_i is the continuous representation of word), results in a continuous vector for each input word. Now we will use an RNN with GRU activations to convert the sequence of continuous vectors corresponding the words into sentence vectors. It is shown in the previous research that on language modeling tasks, GRU and LSTM show analogous performance, also they are conceptually similar [21]. To encode a sentence, we iterate the following sequence of equations as suggested in [3].

$$r^t = \sigma(W_r x^t + U_r h^{t-1}) \quad (1)$$

$$z^t = \sigma(W_z x^t + U_z h^{t-1}) \quad (2)$$

$$\overline{h^t} = \tanh\left(W x^t + U(r^t \odot h^{t-1})\right) \quad (3)$$

$$h^t = (1 - z^t) \odot h^{t-1} + z^t \odot \overline{h^t} \quad (4)$$

Here, r^t= reset gate, z^t= update gate, x^t= embedding for the word, $\overline{h^t}$ = proposed state update at time t, **W**, **U** = parameter matrices, $\odot$ = element-wise product, σ = sigmoid function, $\tanh$ = hyperbolic tangent function.

We use two separate technique for the encoder to extract features as suggested in [3]. These are uni-skip and bi-skip. For uni-skip, the encoder contains a unidirectional forward GRU with 2400 dimension. For bi-skip, the encoder contains a forward and a backward GRU with 1200 dimension of each, forward GRU is offered the sentence in correct order, backward GRU is offered the sentence in reverse order. One output is appended with other to shape a 2400-dimensional vector. We concatenate uni-skip and bi-skip features which we refer to as combine-skip. This is 4800-dimensional vectors.

Now that we have a summary vector that represents a whole input text. This is our feature vector. We feed this feature vector to the machine learning classifier.

C. Classifier

We use Logistic Regression classifier for classification task. We select it for two reasons. The first is, it is a simple linear classifier and fast to train. The second reason is to directly evaluate the performance of the skip-thought vectors on emotion detection task. It is possible to use a non-linear classifier such as Support Vector Machine (SVM) with some additional parameter tuning but that's beyond the scope of this study. Our goal is to analyze the strength of the computed skip-thought vectors on emotion detection task. We used scikit-learn[2] for Logistic Regression implementation.

After running the system, set of emotion labeled text is returned to the user.

V. Datasets

To analyze the performance of skip-thought vectors on emotion detection task, we have explored two publicly available research data (summarized in Table 1).

[1] https://www.nltk.org/

[2] http://scikit-learn.org/

- **ISEAR**[3] [22] (International Survey on Emotion Antecedents and Reactions) contains 7,666 labeled sentences where 1096 participants from different cultural backgrounds accomplished a set of written questions with a choice of answers about the experiences and reactions to seven emotions. This is a well-balanced dataset meaning that the dataset is not biased towards any specific emotion label.
- **Fairy Tales**[4][23] dataset includes annotated sentences from fairy tales. The dataset is not well-balanced. In this dataset, there exist related semantics between anger and disgust. The author of the dataset merges these two emotions to represent one class.

TABLE 1 DATASET CHARACTERISTICS

Dataset	# of data	# of class	Emotion Classes
ISEAR	7,666	7	Angry, Disgust, Fear, Joy, Sad, Shame, Surprise
Fairy Tale	1207	5	Angry-Disgusted, Fearful, Happy, Sad, Surprise

VI. EXPERIMENTS

A. Experimental Setup

We run our framework for both ISEAR and Fairy Tale dataset separately. The pre-trained word vector source from skip-thought model are used as word embedding source in our framework. The dimension of word embedding in skip-thought vectors is 620. To load the pre-trained model, we use several python libraries such as cPickle[5], numpy[6]. For the RNN GRU implementation, Theano[7], a numerical computation library for python, is used. We used 10-fold cross-validation technique to evaluate the performance of our framework.

The performance of Logistic Regression depends on the parameter C. We need to be careful about selecting the value of C. For each fold, we arbitrarily choose the value [1, 2, 4, 8, 16, 32, 64, 128, 256] for parameter C, run another nested 10-fold cross-validation and check which value gives the best scores. We select that value of C for the outer cross-validation checking. During 10-fold cross validation running, we save the scores from each fold and take the mean of all the scores.

Sample input and output from our framework using ISEAR dataset are shown below in Table 2. Here, the input is emotion text and output are the predicted emotion label. The feature extracted from the input text is 4800-dimensional.

TABLE 2 SAMPLE INPUT AND OUTPUT FROM ISEAR DATASET

Input	Extracted Feature	Output
“I received a letter from a distant friend”	[0.418, 0.04492, 0. 32111, ….. ,-0.46765]	Joy
“Close relative is dying of cancer - found out last week”	[0.625, -0.8659, -0. 37282, ….. 0.8372]	Sadness
“I lied to one of my best friends”	[-0.837, 0.37282, -0. 352, ….. ,0.5847]	Guilt

B. Classification Report

Based on the several experimental setups, the summarized results are presented in Table 3 in terms of several evaluation measures including Recall (R), Precision (P), F1 Score (F1).

TABLE 3 CLASSIFICATION REPORT FOR BOTH ISEAR AND FAIRY TALE DATASET

	ISEAR			**Fairy Tale**		
Emotion	***P***	***R***	***F1***	***P***	***R***	***F1***
Joy/Happy	0.700	0.792	0.742	0.732	0.832	0.778
Fear/Fearful	0.692	0.726	0.710	0.636	0.568	0.598
Anger	0.474	0.476	0.474	0.656	0.646	0.644
Sadness	0.630	0.596	0.612	0.652	0.650	0.650
Disgust	0.610	0.598	0.606	-	-	-
Shame	0.524	0.490	0.504	-	-	-
Guilt	0.526	0.496	0.510	-	-	-
Surprise	-	-	-	0.534	0.332	0.406
Average / Total	**0.594**	**0.596**	**0.594**	**0.672**	**0.672**	**0.666**

C. Performance Analysis

When our proposed framework was employed individually on ISEAR dataset, it showed an average precision of 59.4%, average recall of 59.6%, and average F1-measure up to 59.4%. Our framework exhibits high performance for two emotional classes, namely, for the joy and fear classes. Joy appears to be the only category on the positive side of the emotional categories, contrasted with the other negative emotions fear, anger, sadness, disgust, shame, guilt. Anger category showed lower precision of only 47.4%. This is probably because what causes someone angry could be the reason of satisfaction to another.

On Fairy Tales dataset, it showed an average precision of 67.2%, average recall of 67.2%, and average F1-measure up to 66.6%. Fairy Tale dataset was not so well balanced, this dataset is biased to happy emotion, a number of data in surprise category are sparse. We can see the effect of such inconsistency in the classification report. Happy category showed a high recall of 83.2%. On the other side, Surprise category showed a lower recall of 33.2%.

[3] https://www.gelbukh.com/emosenticnet/
[4] http://people.rc.rit.edu/~coagla/affectdata/
[5] https://docs.python.org/2.3/lib/module-cPickle.html
[6] http://www.numpy.org/
[7] http://deeplearning.net/software/theano/

D. Word Embedding Methods Comparison

We compared the performance of skip-thought vectors using our proposed framework with two publicly accessible pre-trained word vector sources, based on GloVe[8] [24] and word2vec[9], regarding emotion detection performance. The dimension of word embedding in both sources are 300.

Emotion detection datasets have been assigned to multiple emotions and are not balanced. That's why we select macro average F1-score for comparative performance analysis of the proposed framework with the related work. We take Continuous bag of words (CBOW) classifier as our baseline from [20].

TABLE 4 COMPARISON OF EACH PRE-TRAINED WORD VECTOR SOURCE WORD2VEC, GLOVE, AND SKIP-THOUGHT IN TERMS OF MACRO F1-SCORES

Dataset	Word2vec	Glove	**Skip-thought**
	CBOW	CBOW	
ISEAR	0.566	0.577	**0.594**
Fairy Tales	0.617	0.661	**0.666**

Table 4 shows the macro F1-scores for both datasets and each text representation technique that is based on word vectors and sentence vectors. The highest scores are printed in bold font. From the result, we can take the decision that for both dataset skip-thought vectors performed better than using word2vec and Glove based vectors. For instance, on an average, skip-thought vectors sources showed a 2 - 5% improvement in F1-score relative to word2vec.

VII. CONCLUSION AND FUTURE WORK

In this research, we presented a recently proposed deep learning model named skip-thought for representing sentences for emotion detection task. We demonstrated the use of a pre-trained sentence embedding model for classifying emotions in user's text. We proposed a framework for emotion detection task which embeds the trained encoder from the skip-thought model along with the lookup table layer. To validate the framework, we executed a set of experiments on two datasets ISEAR and Fairy Tales to explore how effective this framework is at detecting emotions. The RNN sentence embedding technique with skip-thought vectors has exhibited satisfactory results across all performance metrics. We have shown how the problem of emotion detection and classification at the sentence level can be tackled as one of supervised classification, even with relatively small labeled datasets.

In the future, we want to investigate how other word vector sources such as word2vec and Glove perform using RNN sentence embedding technique in terms of emotion detection task. We also plan to train the skip-thought model on domain adapted pseudo-labeled large-scale datasets.

REFERENCES

[1] P. Ekman, "An argument for basic emotions," *Cognition & emotion,* vol. 6, pp. 169-200, 1992.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013.

[3] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015.

[4] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz and J. G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine,* vol. 18, pp. 32-80, 2001.

[5] L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Computing attitude and affect in text: Theory and applications*, Springer, 2006, pp. 1-10.

[6] D. Ghazi, D. Inkpen and S. Szpakowicz, "Prior versus Contextual Emotion of a Word in a Sentence," in *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, 2012.

[7] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics,* vol. 37, pp. 267-307, 2011.

[8] A. Koumpouri, I. Mporas and V. Megalooikonomou, "Evaluation of Four Approaches for Sentiment Analysis on Movie Reviews: The Kaggle Competition," in *Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)*, 2015.

[9] R. Meo and E. Sulis, "Processing affect in social media: a comparison of methods to distinguish emotions in tweets," *ACM Transactions on Internet Technology (TOIT),* vol. 17, p. 7, 2017.

[10] I. Goodfellow, Y. Bengio, A. Courville and Y. Bengio, Deep learning, vol. 1, MIT press Cambridge, 2016.

[11] I. Labutov and H. Lipson, "Re-embedding words," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2013.

[12] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, 2011.

[13] D. Tang, B. Qin and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015.

[14] J. Bradbury, S. Merity, C. Xiong and R. Socher, "Quasi-recurrent neural networks," *arXiv preprint arXiv:1611.01576,* 2016.

[15] O. Irsoy and C. Cardie, "Deep recursive neural networks for compositionality in language," in *Advances in neural information processing systems*, 2014.

[16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078,* 2014.

[17] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Gated feedback recurrent neural networks," in *International Conference on Machine Learning*, 2015.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, pp. 1735-1780, 1997.

[19] M. Zhang, Y. Zhang and D.-T. Vo, "Gated Neural Networks for Targeted Sentiment Analysis.," in *AAAI*, 2016.

[20] J. Herzig, M. Shmueli-Scheuer and D. Konopnicki, "Emotion Detection from Text via Ensemble Classification Using Word Embeddings," in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, 2017.

[21] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555,* 2014.

[8] https://nlp.stanford.edu/data/glove.840B.300d.zip

[9] https://code.google.com/archive/p/word2vec/

[22] K. R. Scherer and H. G. Wallbott, "Evidence for universality and cultural variation of differential emotion response patterning.," *Journal of personality and social psychology,* vol. 66, p. 310, 1994.

[23] E. C. O. Alm, Affect in* text and speech, University of Illinois at Urbana-Champaign, 2008.

[24] J. Pennington, R. Socher and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.