A Sentiment-and-Semantics-Based Approach for Emotion Detection in Textual Conversations

Umang Gupta
Microsoft, Hyderabad, India
umangup@microsoft.com

Ankush Chatterjee*
IIT Kgp, Kharagpur, India
ankushchatterjee@iitkgp.ac.in

Radhakrishnan Srikanth
Microsoft, Hyderabad, India
rsrikan@microsoft.com

Puneet Agrawal
Microsoft, Hyderabad, India
punagr@microsoft.com

ABSTRACT

Emotions are physiological states generated in humans in reaction to internal or external events. They are complex and studied across numerous fields including computer science. As humans, on reading "Why don't you ever text me!" we can either interpret it as a sad or angry emotion and the same ambiguity exists for machines. Lack of facial expressions and voice modulations make detecting emotions from text a challenging problem. However, as humans increasingly communicate using text messaging applications, and digital agents gain popularity in our society, it is essential that these digital agents are emotion aware, and respond accordingly.

In this paper, we propose a novel approach to detect emotions like happy, sad or angry in textual conversations using an LSTM based Deep Learning model. Our approach consists of semi-automated techniques to gather training data for our model. We exploit advantages of semantic and sentiment based embeddings and propose a solution combining both. Our work is evaluated on real world conversations and significantly outperforms traditional Machine Learning baselines as well as other off-the-shelf Deep Learning models.

1 INTRODUCTION

Emotions are basic human traits and have been studied by researchers in the fields of psychology, sociology, medicine, computer science etc. for the past several years. Some of the prominent work in understanding and categorizing emotions include Ekman's six

---

*Work done during Research Internship at Microsoft, India

Figure 1: A sample 3-turn conversation from our dataset.

class categorization [9] and Plutchik's "Wheel of Emotion" which suggested eight primary bipolar emotions [28]. Given the vast nature of study in this field, there is naturally no consensus on the granularity of emotion classes. In this paper we consider 3 emotion classes, Happy, Sad, Angry along with an Others category; and classify a textual conversation into one of the above four.

Problem Definition: *Given a textual user utterance along with 2 turns of context in a conversation, classify the emotion of user utterance as Happy, Sad, Angry or Others.*

Detecting emotions in textual conversations can be a challenging problem in absence of facial expressions and voice modulations. Figure 1 provides an example where it is difficult, even as a human, to detect the emotion of user utterance solely on the basis of text of the conversation. The emotion of the user whose messages are on the left, could be interpreted as angry or sad. The challenge of detecting emotions is further compounded by difficulty in understanding context, sarcasm, class size imbalance, natural language ambiguity and rapidly growing Internet slang.

However, with the growing prominence of messaging platforms like WhatsApp and Twitter as well as digital agents, it is essential that machines are able to understand emotions in textual conversations and avoid responding inappropriately [23]. Emotion detection technology can find several applications in today's online world. In domain of customer service, social media platforms like Twitter are gaining prominence where customers expect quick responses. In case of heavy flow of tweets, turn-around time for responses increase. If tweets can be prioritized according to their emotional content and responded to in that order, it will increase customer satisfaction. For example, responding to an angry tweet prior to a basic inquiry. Also, in this era of text messaging, users are constantly texting and may send inappropriately angry messages to others. If
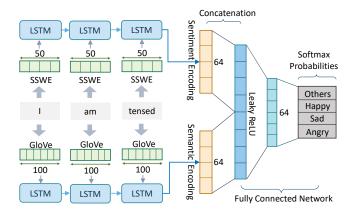
Figure 2: The architecture of Sentiment and Semantic LSTM (SS-LSTM) Model.

emotion detection is implemented, in such cases, the application can take appropriate action such as popping up a warning to the user before sending a message.

In this paper, we propose a deep learning approach for detecting emotions in textual conversations. We use sentiment and semantic representation of text to create a unified LSTM architecture called "Sentiment and Semantic LSTM (SS-LSTM)" to detect emotions. Our model SS-LSTM does not require hand-crafted features and is trained as a single unified model. We evaluate SS-LSTM on real world textual conversations and it outperforms traditional Machine Learning approaches and other Deep Learning based approaches. The main contributions of our paper are as follows:

- We propose a novel deep learning approach called "Sentiment and Semantic LSTM (SS-LSTM)" to detect emotions in textual conversations.
- We evaluate various Deep Learning techniques and embeddings, along with Machine learning algorithms (such as SVM, Decision Trees, Naive Bayes), on real world textual conversations and compare their effectiveness for the task of detecting emotions.

The rest of the paper is organized as follows: Section 2 provides a summary of related work. Section 3 describes our approach (SS-LSTM) in detail. Our experimental setup is discussed in Section 4 and our results are in Section 5. Section 6 concludes the paper and finally Section 7 has acknowledgements.

2 RELATED WORK

The field of sentiment analysis has been extensively studied. However, limited research exists in classifying textual conversations based on emotions. In a bid to gather annotated data on a large scale, some researchers have used automated methods such as emoticons, sentiment analysis and hashtags to label the data [13, 14, 30, 35, 39]. Our method relies on a combination of Deep Learning based data expansion, heuristics and human judgment to create a large corpus of training data for the model.

Emotion-detection algorithms can broadly be categorized into 3 classes:

(a) *Rule based approaches* - Some methods exploit the usage of keywords in a sentence and their co-occurrence with other keywords with explicit emotional/affective value [4, 6, 18, 33, 36]. To that effect, several lexical resources are used, some of the most popular ones being WordNet-Affect [34] and SentiWordNet [10]. Part-of-Speech taggers like the Stanford Parser are also used to exploit the structure of keywords in a sentence. Such methods often need hand-crafting and have good precision, but suffer from low recall since many sentences do not contain affective words despite conveying emotions, e.g. "Trust me! I am never gonna order again". Our method differs from such Rule based approaches as it does not require any hand-crafted features, which are often unable to capture all possible representations of emotions.

(b) *Non-neural Machine Learning approaches* - For sentiment analysis as well as emotion detection, most methods rely on extracting features such as presence of frequent n-grams, negation, punctuation, emoticons, hashtags etc. to form a feature representation of the sentence, which is then used as input by classifiers such as Decision Trees, Naive Bayes, SVMs among others to predict the output [2, 3, 8, 20, 35, 40]. More detailed analysis have been provided in [5]. Vosoughi et al. [38] extract tweets based on location, time and author and uses context to model prior in Bayesian models. These methods often require extensive feature engineering and do not achieve high recall due to diverse ways of representing emotions.

(c) *Deep Learning approaches* - Deep Neural networks have enjoyed considerable success in varied tasks in text, speech and image domains. Variations of Recurrent Neural Networks, such as LSTM [15] and BiLSTM [31] have been effective in modeling sequential information. Also, Convolutional Neural Networks [19] have been a popular choice in the image domain. The lower layers of the network capture local features whereas higher layers unravel more abstract task based features for the image. Their introduction to the text domain has proven their ability to decipher abstract concepts from raw signals [17, 29]. One of the approaches employs CNNs to classify emotion features [24]. The vast success of Deep Neural Nets and their ability to perform tasks without hand-crafting features is our motivation to try these techniques for detecting emotions. We combine both sentiment and semantic features from user utterance to improve emotion detection.

3 OUR APPROACH

We model the task of detecting emotions as a multi-class classification problem where given a user utterance, the model outputs probabilities of it belonging to four output classes - Happy, Sad, Angry and Others. The architecture of our proposed SS-LSTM model is shown in Figure 2. The input user utterance is fed into two LSTM layers using two different word embedding matrices. One layer uses a semantic word embedding, whereas the other layer uses a sentiment word embedding. These two layers learn semantic and sentiment feature representation and encode sequential patterns in the user utterance. These two feature representations are then concatenated and passed to a fully connected network with one hidden layer which models interactions between these features and outputs probabilities per emotion class. Further details of training data used to train the model, sentiment and semantic embeddings, and model training are provided below.

Label	Happy	Sad	Angry	Others	*Total*
#	109	107	90	1920	2226
%	4.90	4.81	4.04	86.25	100

Table 1: Emotion class label distribution in evaluation dataset.

3.1 Training Data Collection

Given the potentially diverse representation of emotions, we collected a large amount of training data using a semi-automated approach. We constructed a dataset of 17.62 million tweet conversational pairs i.e. tweets (Twitter-Qs) and their responses (Twitter-As; collectively referred to as Twitter Q-A pairs below), extracted from the Twitter Firehose, covering the four year period from 2012 through 2015. This data was further cleaned to remove twitter handles and served as the base data for our two training data collection techniques.

Technique 1: In this technique, we start with a small set (approximately 300) of annotated utterances per emotion class obtained by showing a randomly selected sample from Twitter-Qs and Twitter-As to human judges. Using a variation of the model described in [25], we created sentence embeddings for these annotated utterances as well as Twitter-Qs and Twitter-As. We identified potential candidate utterances for each emotion class using the threshold-based cosine similarity between annotated utterances and Twitter-Qs and Twitter-As. Various heuristics like presence of opposite emoticons (example ":'(" in a potential candidate set for Happy emotion class), length of utterances etc. are used to further prune the candidate set. The candidate set is then shown to human judges to determine whether or not they belong to the emotion class. Using this method we cut down the amount of human judgments required by five times when compared to showing a random sample of utterances and choosing emotion class utterances from them.

Technique 2: Once we obtain utterances belonging to an emotion class by the method described above, we take all the utterances that belonged to Twitter-Qs and find their corresponding Twitter-As. These Twitter-As are then further aggregated by their frequency and top Twitter-As are chosen. For example in the Angry emotion class "There, there"[1] was a popular response in Twitter-As. Twitter-Qs corresponding to these top Twitter-As per emotion class are picked as potential utterances in that class and are further shown to human judges for pruning.

Negative data (belonging to class Others) is collected by randomly selecting utterances from both Twitter-Qs and Twitter-As. Those which have a high cosine score (using Technique 1) with any of the utterances in emotion classes (Happy, Sad, Angry) are discarded.

We finally obtained 456k utterances in the Others category, 28k for Happy, 34k for Sad, and 36k for Angry.

---

[1]A phrase frequently used in popular American sitcom, "The Big Bang Theory"

	Happy F1	**Sad** F1	**Angry** F1	Avg. F1
Word2Vec	64.44	74.71	59.28	66.14
FastText	64.58	76.68	59.98	67.08
GloVe	66.11	78.99	63.79	69.63
SSWE	65.64	78.22	63.22	69.32

Table 2: Comparison of results obtained from different embeddings using an LSTM network.

Word1, Word2	GloVe	SSWE
depression, :'(	0.23	0.63
happy, sad	0.59	-0.42
best, great	0.78	0.15

Table 3: Comparison of GloVe and SSWE embeddings w.r.t cosine similarity of word pairs.

3.2 Emoticon Handling and Normalization

Emoticons are frequently used in textual conversations. In Twitter Q-A pairs we found 21% of textual conversations contain emoticons. We used several heuristics and normalization techniques to specifically deal with emoticons. For example, we converted the following utterance "Yeah! :(((My plan is cancelled 😖😖" into "Yeah! :(My plan is cancelled :| :(". This helps us deal with Out of Vocabulary (OOV) issues for infinitely many possible combinations of emoticons, and convert various forms of emoticons which represent similar feelings to a singular form.

3.3 Choosing Input Embeddings

For each word in the input utterance we obtain word embeddings using several techniques. We try Word2Vec [22], GloVe [27], FastText [16] as well as Sentiment Specific Word Embedding (SSWE) [37]. SSWE aims at encoding sentiment information in the continuous representation of words. To test the effectiveness of these embeddings for emotion detection, we train a simple Long-Short Term Memory (LSTM) model using each of these embeddings. LSTMs are variants of Recurrent Neural Networks (RNN) [15] and have the ability to capture long-term dependencies present in the input sequence, and thus are helpful for our task. We use cross validation to determine the effectiveness of different embeddings. Our results, as depicted in Table 2, indicate that GloVe gives the best average F1 score which is slightly better than SSWE F1 score. However, we also observe that GloVe and SSWE behave very differently; a few examples can be seen in Table 3. SSWE embeddings give a high cosine similarity when calculated for "depression" and ":'(" whereas GloVe gives a low score even though the two words have similar sentiment. For the "happy" and "sad" pair, SSWE rightly gives a low score but GloVe outputs a reasonably high score. However, semantically similar words like "best" and "great" have a low cosine similarity with SSWE but high score from GloVe. Based on these observations, we choose GloVe as our embedding for the Semantic LSTM layer and SSWE as our embedding for the Sentiment LSTM layer.

| | **Happy** | | | **Sad** | | | **Angry** | | | |
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Avg. F1
NB	41.35	50.46	45.45	70.87	68.22	69.52	38.16	32.22	34.94	49.97
SVM	66.67	25.69	37.09	86.49	59.81	70.71	85.42	45.56	59.42	55.74
GBDT	75.76	22.94	35.21	89.47	63.55	74.31	86	47.78	61.43	56.98
CNN-NAVA	63.32	42.29	50.71	79.37	68.69	73.64	67.42	45.79	54.54	59.63
CNN-SSWE	67.69	40.37	50.57	77.45	73.83	75.6	80.95	37.77	51.51	59.23
CNN-GloVe	52.29	52.29	52.29	93.72	67.29	74.61	67.82	65.55	66.66	64.52
LSTM-SSWE	70.69	37.61	49.1	83.87	72.89	78	73.24	57.77	64.6	63.9
LSTM-GloVe	64.18	39.45	48.86	72.88	80.37	76.44	72.15	63.33	67.45	64.25
SS-LSTM	69.51	52.29	**59.68**	85.42	76.63	**80.79**	87.69	63.33	**73.55**	**71.34**

Table 4: Comparison of various models on evaluation dataset. SS-LSTM results are statistically significant with p < 0.005

3.4 Model Training

We use the Microsoft Cognitive Toolkit[2] for training SS-LSTM. The parameters of SS-LSTM are trained to maximize prediction accuracy given the target labels in the training set. We split our training data in a 9:1 ratio to create sets for training and validation respectively. We train the model using the training set and tune the hyper-parameters using the validation set. We use Cross Entropy with Softmax as our loss function [12], and Stochastic Gradient Descent (SGD) as our learner. We found the optimal batch size to be 4000 with a learning rate of 0.005. It is worth noting that when training sequence classification models, the Microsoft Cognitive Toolkit uses the sum of the length of sequences across utterances (not the number of utterances) when picking up data of a particular batch size.

4 EXPERIMENTAL SETUP

In this section we describe details of evaluation dataset used to compare various techniques and baseline methods used for comparison.

4.1 Evaluation Dataset

We are aware of two datasets in this domain: (a) The ISEAR dataset[3] and (b) The SemEval2007 Affective Text Dataset[4]. However, both these datasets are unsuitable for evaluating our task. ISEAR dataset consists of user reactions when they were asked to remember a circumstance which aroused certain emotions in them. For example "When my mother slapped me in the face, I felt anger at that moment." is one of the statements in ISEAR dataset and has a different form than what one would expect in a conversation. On the other hand, SemEval2007 dataset consists of news headlines which are again not similar to conversations.

To overcome these challenges we sample 3-turn conversations from Twitter i.e. User 1's tweet; User 2's response to the tweet,

and User 1's response to User 2. We used the Twitter Firehose to extract these 3 turn conversations covering the year of 2016. We sampled from conversations where the last turn was the third turn as well as from those where the third turn was in the middle of the conversation. Our dataset finally comprised of 2226 3-turn conversations along with their emotion class labels (Happy, Sad, Angry, Others) provided by human judges. The details of the dataset along with emotion class label statistics is shown in Table 1. To gather the emotion class labels, we showed the third turn of the conversation along with the context of the previous 2 turns to human judges and asked them to mark the emotion of the third turn after considering the context. To gather high quality judgments each conversation was shown to 5 judges, and a majority vote was taken to decide the emotion class. After several rounds of training and auditing of mock sets, the final inter-annotator agreement based on fleiss' kappa value [32] was found to be 0.59. This kappa value, while slightly less then desirable, indicates the difficulty in judging textual conversations due to ambiguities discussed earlier in Section 1.

Our evaluation dataset is unseen at time of training. SS-LSTM and all baseline approaches are evaluated on this dataset.

4.2 Baseline Approaches

We compare our approach against two classes of baselines. (a) Machine Learning based baselines and (b) Deep Learning based baselines.

For Machine Learning based baselines we used a Support Vector Machine (SVM) classifier [7], a Gradient Boosted Decision Tree (GBDT) classifier [11] and a Naive Bayes (NB) classifier [11]. SVM, GBDT and NB classifiers were trained using Scikit Learn [26]. One of the salient features of our approach is the lack of need for feature engineering. Hence we kept the feature set small for SVM, GBDT and NB. We used 1,2,3 n-grams as features along with a hand-crafted emoticon feature set. This emoticon feature set is a 3 dimensional vector where the first dimension is the count of Happy emoticons like ":)" in the utterance. Similarly the second and third dimension are for Sad and Angry Emotions. After tuning parameters using the

---

[2]https://www.microsoft.com/en-us/cognitive-toolkit/
[3]http://www.affective-sciences.org/en/home/research/materials-and-online-research/research-material/
[4]http://nlp.cs.swarthmore.edu/semeval/tasks/task14/data.shtml

#	True Label	User 1's tweet	User 2's response	User 1's response	Comment
1	Angry	It will be arranged within two business days?	It will be done at the earliest	This is getting very annoying now, no pickup yet!	LSTM-SSWE and SS-LSTM predict correctly, probably because of the keyword 'annoying' which represents negative sentiment. LSTM-GloVe fails.
2	Sad	Man even food delivery apps in bangalore won't deliver till 6:(	Yea well it is a bandh	Yeah well i do not have anything at home :/	LSTM-SSWE fails as there is no keyword with an obvious negative polarity but LSTM-GloVe and SS-LSTM are correct.
3	Angry	:) Good for both of us!	It's better not to interact with a girl with so much ego. Attitude is still fine	It is not an ego or attitude. U started first! U asked me stupid ques! :/	SS-LSTM is only model which could correctly predict this rather complicated user utterance.
4	Sad	3 gone 2 more to go :3	Crores? :D	Haha no ya. My kittens. One by one they are all leaving :'(	Presence of 'Haha' and ":'(" make this case difficult to predict and all models including SS-LSTM fail
5	Happy	I just qualified for the Nabard internship	WOOT! That's great news. Congratulations!	I started crying	All models predicted it as Sad, however, when one takes into account context, true emotion is Happy.

Table 5: Qualitative Analysis of SS-LSTM results and other baseline approaches.

validation set as described in Section 3.4 we found SVM to give the best performance with linear Kernel and regularization constant 0.005. In case of GBDT, the best performance was achieved with 50 trees and a minimum of 10 samples per leaf.

For deep learning based baseline we implemented the approach defined in [24]. To the best of our knowledge this work is the only other deep learning based approach attempted to detect emotion classes. We call this approach CNN-NAVA. We trained emotion vectors as defined in [1] and used them as input to a CNN model. We also trained individual CNN and LSTM models with different embeddings like GloVe and SSWE.

We used Precision, Recall, F1 score, and Average F1 score (where average is taken across F1 scores of emotion classes i.e. happy, sad and angry) to evaluate different approaches.

5 RESULTS

A summary of results from various techniques on the dataset described in Section 4.1 is presented in Table 4. SS-LSTM gives the best performance on F1 score for each emotion class as well as on Average F1. The performance of SS-LSTM over all other models is particularly significant (p < 0.005) as measured by McNemar's test [21]. Our results thus indicate that combining sentiment and semantic features in SS-LSTM outperforms individual LSTM-SSWE and LSTM-GloVe. SS-LSTM was also significantly better than CNN based approaches including CNN-NAVA. Also, when comparing across models using Average F1 score, Deep Learning based models outperform NB, SVM and GBDT.

5.1 Qualitative Analysis

Table 5 highlights some examples from evaluation set and compares the performance of our models across these examples. We observe that if user utterance had keywords or emoticons with a certain sentiment polarity associated with them, LSTM-SSWE usually works well even if LSTM-GloVe does not. The absence of the same affected

#	User 1	User 2	User 1
1	Good morning! weekend	Good morning. :) :) :) :)	Happy Morning
2	What r the birthday plans? ;)	going to hills with friends.	Oh great!
3	I had a match today.	And did you win?	Yes!! And I am super happy :)

Table 6: Sample conversations indicating challenges in Happy emotion class

LSTM-SSWE's performance. SS-LSTM, by combining both the feature sets, is able to accurately predict examples #1-3 in Table 5. Specifically, in #3, all baseline approaches fail, but SS-LSTM is able to harness the advantage of combining both semantic and sentiment features to predict it correctly. However, SS-LSTM still needs further improvement. For example in #4 presence of keywords like "Haha" and ":'(" make it difficult for all models to predict it correctly. In some utterances like in #5 context of the conversation plays an important role to determine underlying emotion, SS-LSTM does not consider context and hence fails as do all other models.

5.2 Discussion on Ambiguity in Happy Class

On comparing the F1 scores of several models in Table 4 we observe that the Happy emotion class performs significantly worse than other emotion classes. We found inter-judge agreement to be particularly low for the Happy emotion class, which indicates variation in how a user utterance is interpreted by different human judges. In example #1 of Table 6, User 1's second utterance is interpreted as Happy by some judges and just as a greeting by some other judges who mark it as Others. Similarly in example #2, User 1's second utterance is considered a comment by some and happy statement by others due to the keyword "great". While in example #3 User 1 is visibly happy, which is marked Happy by most judges. We

thus believe that predicting utterances for the Happy class on basis of textual conversation alone is a challenging problem and hence, understanding context becomes even more important for this class.

6 CONCLUSION

We proposed a Deep Learning based approach called "Sentiment and Semantic LSTM (SS-LSTM)" to detect emotions in textual conversations. Our approach combines sentiment and semantic features from user utterance using SSWE and GloVe embeddings respectively and do not require any hand-crafted features. Evaluation on real world textual conversation shows that our approach outperforms CNN and LSTM baselines, in addition to other Machine Learning baselines. We observe that our approach can benefit from the ability to handle context. As part of future work, we plan to extend this approach to train models that are context aware.

7 ACKNOWLEDGMENTS

REFERENCES

[1] A. Agrawal and A. An. Unsupervised emotion detection from text using semantic and syntactic relations. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, pages 346–353, 2012.

[2] C. O. Alm, D. Roth, and R. Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. ACL, 2005.

[3] R. C. Balabantaray, M. Mohammad, and N. Sharma. Multi-class twitter emotion classification: A new approach. *International Journal of Applied Information Systems*, Vol. 4, pages 48–53, 2012.

[4] A. Balahur, J. M. Hermida, and A. Montoyo. Detecting implicit expressions of sentiment in text based on commonsense knowledge. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 53–60. ACL, 2011.

[5] L. Canales and P. Martínez-Barco. Emotion detection from text: A survey. *Processing in the 5th Information Systems Research Working Days (JISIC)*, page 37, 2014.

[6] F.-R. Chaumartin. Upar7: A knowledge-based system for headline sentiment tagging. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 422–425. ACL, 2007.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, Vol. 20, pages 273–297, 1995.

[8] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd international conference on computational linguistics: posters*, pages 241–249. ACL, 2010.

[9] P. Ekman. An argument for basic emotions. *Cognition & emotion*, Vol. 6, pages 169–200, 1992.

[10] A. Esuli and F. Sebastiani. Sentiwordnet: A high-coverage lexical resource for opinion mining. *Evaluation*, pages 1–26, 2007.

[11] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[13] M. Hasan, E. Agu, and E. Rundensteiner. Using hashtags as labels for supervised learning of emotions in twitter messages. In *ACM SIGKDD Workshop on Health Informatics, New York, USA*, 2014.

[14] M. Hasan, E. Rundensteiner, and E. Agu. Emotex: Detecting emotions in twitter messages. 2014.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, Vol. 9, pages 1735–1780, 1997.

[16] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[17] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[18] Z. Kozareva, B. Navarro, S. Vázquez, and A. Montoyo. Ua-zbsa: a headline emotion classification through web information. In *Proceedings of the 4th international workshop on semantic evaluations*, pages 334–337. ACL, 2007.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] F. Kunneman, C. Liebrecht, and A. van den Bosch. The (un) predictability of emotional hashtags in twitter. *In European Chapter of the Association for Computational Linguistics*, pages 26–34, 2014.

[21] Q. McNemar. *Psychological statistics*. Wiley New York, 1969.

[22] 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*, pages 3111–3119, 2013.

[23] A. S. Miner, A. Milstein, S. Schueller, R. Hegde, C. Mangurian, and E. Linos. Smartphone-based conversational agents and responses to questions about mental health, interpersonal violence, and physical health. *JAMA internal medicine*, Vol. 176, pages 619–625, 2016.

[24] S. Mundra, A. Sen, M. Sinha, S. Mannarswamy, S. Dandapat, and S. Roy. Fine-grained emotion detection in contact center chat utterances. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 337–349. Springer, 2017.

[25] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, Vol. 24, pages 694–707, 2016.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, Vol. 12, pages 2825–2830, 2011.

[27] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume Vol. 14, pages 1532–1543, 2014.

[28] R. Plutchik and H. Kellerman. *Emotion: theory, research and experience*. Academic press New York, 1986.

[29] A. Prakash, C. Brockett, and P. Agrawal. Emulating human conversations using convolutional neural network-based ir. *arXiv preprint arXiv:1606.07056*, 2016.

[30] M. Purver and S. Battersby. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. ACL, 2012.

[31] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, Vol. 45, pages 2673–2681, 1997.

[32] P. E. Shrout and J. L. Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, Vol. 86, page 420, 1979.

[33] C. Strapparava and R. Mihalcea. Learning to identify emotions in text. In *2008 ACM symposium on Applied computing*, pages 1556–1560, 2008.

[34] C. Strapparava, A. Valitutti, et al. Wordnet affect: an affective extension of wordnet. In *The 4th International Conference on Language Resources and Evaluation*, volume Vol. 4, pages 1083–1086, 2004.

[35] J. Suttles and N. Ide. Distant supervision for emotion classification with discrete binary values. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 121–136. Springer, 2013.

[36] M. D. Sykora, T. Jackson, A. O'Brien, and S. Elayan. Emotive ontology: Extracting fine-grained emotions from terse, informal messages. 2013.

[37] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565, 2014.

[38] S. Vosoughi, H. Zhou, and D. Roy. Enhanced twitter sentiment classification using contextual information. In *6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, page 16, 2015.

[39] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth. Harnessing twitter "big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust, 2012 International Conference on Social Computing*, pages 587–592. IEEE, 2012.

[40] J. L. S. Yan and H. R. Turtle. Exploring fine-grained emotion detection in tweets. In *The North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 73–80, 2016.