Emotion Detection and Classification in a Multigenre Corpus with Joint Multi-Task Deep Learning

Shabnam Tafreshi
George Washington University
Department of Computer Science
GWU NLP Lab
shabnamt@gwu.edu

Mona Diab
George Washington University
Department of Computer Science
GWU NLP Lab
mtdiab@gwu.edu

Abstract

Detection and classification of emotion categories expressed by a sentence is a challenging task, due to subjectivity of emotion. To date, most of the models are trained and evaluated on single genre and when used to predict emotion in different genre, their performance drops by a large margin. To address the issue of robustness, we model the problem within a joint multi-task learning framework. We train this model with a multigenre emotion corpus to predict emotions across various genres. Each genre is represented as a separate task, we use soft parameter shared layers across the various tasks. our experimental results show that this model improves the results across the various genres, compared to a single genre training in the same neural net architecture.

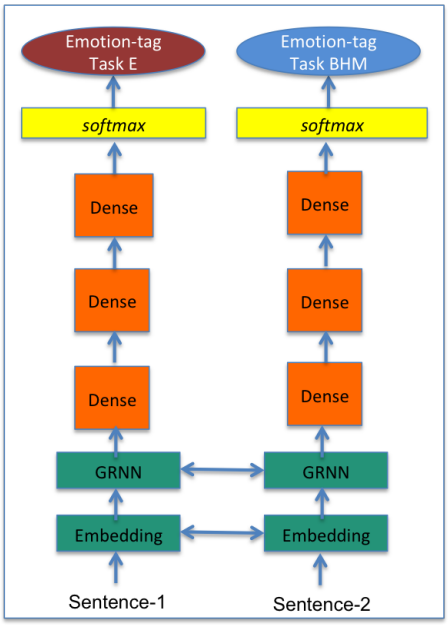

Figure 1: Joint Multi-Task Emotion neural net architecture. In this multi-task framework, Task-E and Task-BHM (explained in section 3) demonstrate different genres.

1 Introduction

Sentence-level emotion detection is garnering a lot of attention recently. To date, most systems are trained and evaluated on a single genre resource. However, the problem is robustness, when such models are applied to new genres, the performance expectedly drops significantly. In this paper, we propose a model to address the genre robustness issue.

Proceedings of the 27th International Conference on Computational Linguistics, pages 2905–2913
Santa Fe, New Mexico, USA, August 20-26, 2018.

Dataset	joy	trust	anti	surprise	sad	fear	anger	disgust
BLG	689	43	260	150	312	132	192	255
HLN	106	6	56	31	83	68	28	55
MOV	4875	26	119	255	258	63	20	5145
EmoNet	5670	75	435	436	653	263	240	5455
BHMT	11340	150	870	872	1306	526	480	10910

Table 1: Data statistics illustrating the distributions of the various emotion tags from PL8 across the different genres.

We frame the problem as a joint multi task learning problem, where each of the genres is represented as a separate task that shares information with other tasks (genres). Our model, the Joint Multi-Task Emotion (JMTE), illustrated in figure 1, trains on emotions in four genres of data: Blog Posts, News Headlines, Movie Reviews, and Tweets. Experimental results show that using JMTE achieves better results, over a single model trained on a single of these different genres. We also study the impact of specifically adding Twitter data that is distantly supervised to the training models.
The rest of this paper is organized as follows: We describe our multigenre corpus in section 2; We present our devised JMTE model in section 3; Experimental conditions and results are presented in section 4; Discussion is presented in section 5; We review related studies in section 6; Conclusions are described in section 7.

2 Data

We create a unified multigenre data set annotated on the sentence and clause level using the 8 emotions from Plutchik (Plutchik, 1962), which includes the following 8 emotions: *joy, trust, anticipation, surprise, anger, fear, sadness, disgust* (PL8) and a *no-emotion* category (Tafreshi and Diab, 2018). [1]
Our combined multigenre corpus includes the following data sets: The emotional blog post (BLG) (Saima and Stan, 2007) comprising 4,115 sentences; The headlines dataset (HLN) (Carlo and Rada, 2007) comprising 1,250 sentences; a movie review dataset (MOV) (Bo and Lillian, 2005) where people express their opinions about movies, sound tracks, and casts. The MOV dataset contains 11,855 sentences. Both BLG and HLN were originally annotated using the 6 basic emotion categories from (Paul, 1992), *happiness, sadness, fear, anger, surprise and disgust* (EK6), while the MOV data set was annotated for sentiment and sentiment intensity. The three corpora resulted in 17,220 sentences annotated with the PL8 emotion tag set, from which 3993 sentences are annotated with *no-emotion*. That corpora annotated using the crowd sourcing platform CrowdFlower[2]. The inter-annotator agreement (IAA) achieved was 79.95%. We refer to this data set as BHM corresponding to BLG, HLN, and MOV, respectively.
We further experiment with a fourth dataset from Twitter. We added 13,227 randomly selected tweets (to match the BHM collection size) from the EmoNet (Muhammad and Ungar, 2017) tweet collection, which has a total of 547,555 tweets tagged with PL8 and its extension into 16 fine grained emotion tags. EmoNet is labeled using distant supervision, namely relying on hashtag information to render the emotion tag. We only selected for our corpus tweets that had emotion tags corresponding to the PL8 emotions. This collection has no *no-emotion* tag. We refer to this data set as TweetEN. Accordingly, our annotated multigenre corpus includes: BLG, HLN, MOV, and TweetEN, comprising a total of 26,454 annotated sentences with PL8 labels. We refer to this combined corpus as BHMT. Table-1 shows data statistics.

3 Proposed Approach

We model the problem as a joint multitask learning architecture. We use a Gated Recurrent Neural Network architecture (GRU), inspired by (Muhammad and Ungar, 2017). We create two tasks in JMTE,

[1] You can download BHM dataset and JMTE python code from `https://github.com/shabnamt/jointMultitaskEmo`

[2] `https://www.crowdflower.com`

one trained on the TweetEN data set (Task-E) and the other trained on the BHM data (Task-BHM). The reason to model them separately is that they are annotated in very different ways: TweetEN is annotated using distant supervision relying on hashtags, vs. BHM which is annotated completely manually. We balance the distribution of labeled data per emotion across the two tasks.

Recurrent Neural Network (RNN)- has been widely used in the literature to model sequential problems. RNN applies the same set of weights recursively as follow:

$$h_t = f(W_{x_t} + Uh_{t-1} + b) \tag{1}$$

RNN input vector $x_t \in R^n$ at time step t is calculated based on a hidden state and an input from the current state based on Eq. 1. The function f is a nonlinearity such as tanh or ReLU. Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Neural Nets- (Cho, 2014; Chung, 2015) are implementations of RNNs that circumvent some of the major issues with RNNs as they are much better at capturing long-term dependencies, and dealing with the vanishing gradient problem (Bengio et al., 1994; Pascanu et al., 2013).

Gated Recurrent Neural Nets- (Cho, 2014; Chung, 2015) is very similar to LSTM with the following equations:

$$r_t = \sigma(W^r_{x_t} + U^r h_{t-1} + b^r) \tag{2}$$

$$z_t = \sigma(W^z_{x_t} + U^z h_{t-1} + b^z) \tag{3}$$

$$\hat{h_t} = \tanh(W_{x_t} + r_t \times U^{\hat{h}} h_{t-1} + b^{\hat{h}}) \tag{4}$$

$$h_t = z_t \times h_{t-1} + (1 - z_t) \times \hat{h_t} \tag{5}$$

GRU has two gates, a reset gate r_t, and an update gate z_t. Intuitively, the reset gate determines how to combine the new input with the previous memory, and the update gate defines how much of the previous memory to keep around. We use Keras[3] GRNN implementation to setup our experiments. We note that GRU units are a concatenation of GRU layers in each task.

Shared layers - these two tasks, Task-E and Task-BHM share the word embedding (*i.e.* Keras embedding) layer. We setup this layer in two different ways: a) initiating random weights for each word vector and tune these weights using the training set, b) initiating the weights from pre-trained word embedding model and let the embedding layer to tune the weights using the training set. The task specific GRU layers share soft parameters between each of the two layers, we observed that soft parameter sharing creates better weights for this layer. Shared GRU takes advantage of the underlying emotion cues structure shared among the genres (*e.g.* presence of adjective and adverbs) as well as semantic and syntactic emotion features that improve emotion detection and classification in different genres.

Task Specific Layers - each task has 3 hidden dense layers and a *softmax* layer for prediction, this setup allows freedom for each task to have auxiliary input layers and be optimized per task. We optimize these layers per task.

3.1 Training JMTE Model

We set the dimensionality of the input embedding layer to 300 and the hidden GRU to 70. We concatenate clause feature, explained at 4.2 to Task-BHM embedding layer. At each training epoch, we train the model in the following order: shared embedding, shared soft parameters of the hidden GRU, 3 specified hidden dense layers for each task. We experimented with different number of units in dense layers and our results on dev set suggest the following setup for dense layers: 300 units for Task-E and 200 units for Task-BHM. We use a *softmax* layer for predicting emotion tags. Further, we use an input maximum length of 70, 10 epochs, and Adam (Kingma and Ba, 2014) optimizer with a learning rate 0.001. We use

[3] https://keras.io/

Condition	Training set	TweetEN	BLG+HLN	MOV
LL1	TweetEN	*21.9%*	6.9%	35.8%
LL2	BLG+HLN	37.0%	***38.3%***	62.4%
LL3	MOV	25.2%	17.0%	*32.1%*
LL4	BHMT	**43.9%**	21.7%	**79.0%**

Table 2: LIBLINEAR weighted F1 scores for different experimental conditions where we train on various training data set combinations and test within and across genres. Within genre is marked in italics. It should be noted that we did not balance the size of the training data across the different experimental conditions.

dropout (Graves et al., 2013) for regularization, with a dropout rate: 0.3 for both tasks. The loss function is a categorical-cross-entropy function. We use a mini batch (Cotter, 2011) of size 65.

4 Experiments and Results

4.1 Data

We split the data representing each emotion category per genre into 70%,10%,20% for train, dev, and test, respectively. We added dev set to training set after tuning our model parameters on dev set.

4.2 Baseline Models

We compare our proposed models to two baselines: a feature engineering architecture LIBLINEAR from the SVM family. We leverage LIBLINEAR architecture implementation in Weka.[4]; and a single GRU model architecture where we use all the data from both TweetEN and BHM in a single model.

Feature Engineering Baseline: For the LIBLINEAR setting, we build our model combining a number of features: character and word n-grams (uni-gram and bi-gram); POS: presence of POS tags taken from PennTreebank; syntactic features like presence of adjective, adverbs, or negation; and semantic features like presence of emotion words based on EmoNet lexicon (Mohammad, 2012), and clause feature, which we explain below. We report weighted F1 scores across the 8 PL8 emotion tags.
Clause feature - for this feature, we study the distribution of clauses emotion tags in multi-clausal sentences. We note that the majority of those sentences with multiple clauses tend to have clauses with specific emotion labels (e.g. sentence emotion tag *joy*, have clauses with tags *trust, anticipation, no-emotion, and surprise*). We model this feature as an 8-dimension vector, where each dimension represent one emotion tag with a binary value: 1 indicates the presence of sub-sentential emotion clause tag and 0 otherwise.

Table 2 illustrates the results of the LIBLINEAR models. We combine the test sets for BLG and HLN as they are relatively small. In LL1, training with TweetEN yields the worst results on all 3 data sets even on the TweetEN test set (within genre setting). In LL3, training with MOV yields the lowest within genre results compared to the other sets. In LL4, training with a combination of BLG+HLN+MOV and TweetEN yields the best results on TweetEN and MOV data sets. In addition, in this condition we combine gold annotated set, BHM, with distant supervision set TweetEN. In LL2, training with BLG+HLN, which is the smallest training set yields the best results on BLG+HLN, and compare to LL1, yields better results on TweetEN and MOV. Hence, size is not a factor in the performance and the feature engineering approach is not robust towards genre variation.
Single Task NN Learning Baseline: We experiment with a GRU architecture as a single task. The architecture is similar to the JMTE framework with an embedding layer, followed by a GRU, then 3 dense layers followed by a softmax classification layer. Both the GRNN and JMTE are implemented using Keras and Tensorflow[5] in the backend. Table 3 present the results for single task GRU baseline. We mentioned earlier about embedding layer setup in our model, we experimented with two different

[4] https://www.cs.waikato.ac.nz/ml/weka/
[5] https://www.tensorflow.org/

Condition	training set	TweetEN	BLG+HLN	MOV
GRNN1	TweetEN	61.9%	27.2%	36.5%
GRNN1-pt		65.5%	29.9%	45.7%
GRNN2	BLG+HLN	33.7%	31.2%	27.4%
GRNN2-pt		40.2%	37.2%	38.2%
GRNN3	MOV	45.3%	26.4%	66.2%
GRNN3-pt		46.6%	28.02%	69.6%
GRNN5	BHMT	76.2%	81.2%	89.8%
GRNN5-pt		**78.1%**	**83.6%**	**91.0%**

Table 3: Single task GRNN F-score results on all the test data sets. Using pre-train (GRNN-pt) word embedding to initiate the weights for embedding layers, creates better results across all conditions. Colored results are out-of-genre evaluations.

Model	TweetEN	BLG+HLN	MOV
LL5	43.9%	21.7%	79.0%
GRNN5-pt	78.1%	83.6%	91.0%
JMTE	78.5%	82.3%	92.2%
JMTE-pt	**80.0%**	**84.0%**	**92.6%**

Table 4: Weighted macro F1-scores yielded by baseline models using all the training data BHMT compared against the JMTE model.

setup, a) we initiate random weights for word vectors b) we initiate the weights using pre-trained word embedding; in both of these settings we tuned the word vectors using the training set. We experimented with different word embedding models, mainly to have a better coverage for all these 4 genres in our corpus. Common training set for word embedding models are wiki+news, news, tweets, and common crawl, and the methods are Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fasText (Bojanowski et al., 2016). Our results indicate that common crawl corpus with 2 million words, trained using fastText model has the most word coverage among these genres. [6] We experimented with google news (trained using word2vec), wikipedia+Gigaword (trained using GloVe), Twitter (trained using GloVe).[7]

The best results are obtained using all the data in condition GRNN5-pt which trains on all the labeled data, and we used pre-trained word embedding to initialize the wights for embedding layer. The results yielded by the Deep learning model surpass those of the LIBLINEAR baseline by a significant margin comparing GRNN5-pt (F1 scores of: 78.1%, 83.6%, 91.0%, for TweetEN, BLG+HLN, MOV, respectively) compared with LL5 (F1 scores of: 43.9%, 21.7%, 79%, for TweetEN, BLG+HLN, MOV, respectively). GRNN1-pt and GRNN3-pt indicate that within genre training and testing yields the best results. Even when there is more data available for training, comparing GRNN1-pt (more training data) compared to GRNN3-pt condition, GRNN3-pt yields higher results on the MOV (within genre) test data at 69.9% vs. 45.7% F1 score as yielded from GRNN1-pt condition. GRNN1-pt consistently beats LL1 across all test sets, likewise for GRNN3-pt and LL3. The pattern is consistent.

Joint MultiTask Learning of Emotion Model: Table 4 shows the performance of the JMTE model proposed in this paper against the two baselines GRNN (GRNN5-pt) and LIBLINEAR (LL5) when using all the data for training. Overall, the JMTE-pt yields the best results across all test data sets, significantly outperforming the LIBLINEAR baseline as well as beating the single task GRNN architecture.

5 Discussion

It is worth noting that we could not compare our results against other systems in the literature since available systems are typically trained on the EK6 tag set. The only system we know that is trained and tested on the PL8 tag set is that of Muhammad and Ungar (2017) but the test set is twitter data which is

[6] https://fasttext.cc/docs/en/english-vectors.html
[7] https://nlp.stanford.edu/projects/glove/

Test set	Model	joy	trust	anti	surprise	sad	fear	anger	disgust
TweetEN	JMTE-pt	86.2%	31.3%	57.5%	52.1%	69.8%	29.2%	26.3%	86.6%
	GRNN5-pt	85.2%	22.1%	54.9%	48.9%	53.2%	39.9%	39.9%	85.6%
	LIBLINEAR	42.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	63.0%
MOV	JMTE-pt	93.5%	29.0	68.8%	50.4%	66.2%	64.2%	0.0%	93.2%
	GRNN5-pt	92.5%	29.0	68.1%	50.4%	65.8%	63.8%	0.0%	93.1%
	LIBLINEAR	88.8%	0.01%	34.8%	39.0%	48.8%	20.0%	0.0%	90.3%
BLG+HLN	JMTE-pt	88.7%	53.2%	83.6%	87.3%	84.2%	86.4%	83.3%	83.7%
	GRNN5-pt	87.7%	52.2%	83.6%	87.3%	84.2%	86.4%	82.1%	82.7%
	LL5	78.4%	28.6%	55.8%	45.5%	54.5%	51.9%	66.7%	50.0%

Table 5: F-score results with LIBLINEAR and JMTE across different emotion tags, per genre. Emotion tags *anger* and *trust* are minority in MOV, hence, the model could not predict correct instances in these two emotion categories.

not a stable data set.[8] Furthermore, in an attempt to understand the performance of the various models per emotion tag. Table 5 compares F-score per emotion tag between LIBLINEAR and JMTE, which indicates JMTE is able to learn each emotion tag better than LIBLINEAR and GRNN models. We built JMTE to generalize emotion detection and classification across different genres.
Certainly, adding more data would create better results, but this is not the aim of our study. We observe that LIBLINEAR has shortcoming when we add TweetEN to the training set, and when we train LIBLINEAR with different genre than TweetEN. JMTE and GRNN overcome this shortcoming across all genres, and for some genres the improvement is significant (*i.e.* BLG+HLN and TweetEN).
GRNN produces close results to JMTE, however JMTE has the advantage to be specialized per genre by adding auxiliary layers and genre specific layers.

LIBLINEAR trained on BLG+HLN+MOV performed poorly on tweetsEN, this model can only classify *joy* and *disgust*, with f-score 42.0% and 63.0% respectively and these two emotion tags are the most populated tags in this set.
Although JMTE is able to generalize emotion classification across genres and create the best results, however, GRNN trained on BLG+HLN+MOV created better results for HLN, which indicates JMTE needs auxiliary (*i.e.* specific) layer for different genres, particularly if the genre has lower amount of test and train data (HLN) or genre is very different compare to other genres (MOV). In our setup HLN has the minimum number of instances in multigenre corpus (training: 825 and test: 425) and MOV is very different compare to tweets, BLG, and HLN, hence, adding specific layers for these genres can improve the results.
Other challenge here is pre-trained word embedding model. We observed that pre-trained word embedding creates better results, however, none to the best of our knowledge are trained using different genres to correspond to our need. We observe the results in Table 3 using pre-trained word embedding, and the significance of better coverage for all the genres. Comparing GRNN4 vs GRNN4-pt on TweetEN 31.7% vs. 32.9%, which has only 1% improvement, vs., GRNN2 vs. GRNN2-pt on TweetEN 33.7% vs. 40.2% which has 7% improvement, which, is an indication that an effective word embedding can improve the results by a large margin, as GRNN2-pt has less training data compare to GRNN-4.
Further, we observed that most of the instances that are misclassified in BLG, HLN, and MOV, are the ones with lower confidence score annotation, as these sets are manually annotated. In addition, observing the confusion matrix in JMTE suggests that in both tasks data points are separable, since misclassified data points are not skewed towards particular or the most popular classes. However, emotion tag *surprise* is mainly misclassified with *joy* and *discussed*, we further observed the data points to address this issue and noticed most of these data points imply surprise and even as human it was difficult for us to categorize them as surprise.

[8]Unfortunately due to licensing issues, exact tweets can't be shared, only IDs, hence when retrieving the actual tweets, we noted that we can only retrieve 75% of the exact tweets.

6 Related Work

6.1 Emotion Detection and Classification

Emotion detection has attracted several NLP applications like chatbots, stock market, and human personality analysis. Several studies investigated the problem in various genres. We present some of the studies most relevant to this paper. In the literature, emotion detection is cast as a classification problem, hence, the objective is to effectively learn emotion cues. Among the feature engineering approaches, we review the following works:Gilad (2005) collected a set of blog posts - online diary entries - which include an indication of the writers' mood. Carlo and Rada (2007) collected and manually labeled 1,250 headlines (HLN) for emotion classification and valence (*i.e.* positive, negative) using the 6 basic emotions identified by Ekman (Paul, 1992) (EK6) tags. Saima and Stan (2007) collected and labeled a blog posts corpus (BLG) using EK6 tags on both the sentence and the phrase levels, they annotated emotion categories, emotion intensity, and identifying emotion phrases in blog posts. Diman et al. (2010) experimented with hierarchical classification for emotion analysis which considers the relation between neutrality, polarity and emotion of a text. Diana and Carlo (2010) presented a categorical model and dimensional model for recognition of affective states (emotion cues). Mohammad (2012) investigated word-level affect lexicons features on sentence-level emotion detection. Özbal and Daniele (2013) showed the effect of incorporating different levels of syntactic and semantic information on sentence level emotion detection.
In recent years, unlimited access to social media data such as *Twitter and Facebook*, enabled the community to have access to large amount of data. In these works researchers have used supervised learning model trained on lexical, semantic, and stylistic features to classify emotion in Twitter (Wenbo et al., 2012; Roberts et al., 2012; Ashequl and Ellen, 2013; Yan, 2014; Saif and Svetlana, 2015; Yan and Howard, 2016; Svitlana and Yoram, 2016). Muhammad and Ungar (2017) proposed a gated recurrent neural network architecture to classify emotion in tweets.

6.2 Multi-Task Learning in Deep Neural Net

Multi-Task learning is inspired by human learning. As human, when we learn new tasks, often we apply the knowledge we have gathered from related tasks. In the literature, multi-task learning comes in different forms: joint learning, learning to learn, and learning with auxiliary tasks. The following studies show the benefit of multi-task learning in closely-related or different type of tasks (R, 1993; Ronan and Jason, 2008; Collobert et al., 2011; Xiao et al., 2011; Seltzer and Droppo, 2013; Devries et al., 2014; Xia and Liu, 2015; Luong et al., 2015; Anders and Goldberg, 2016; Kazuma et al., 2016; Andor et al., 2016; Jonathan et al., 2016; Makoto and Bansal, 2016).
To the best of our knowledge multi-task learning has not been studied to detect emotion in multigenre text input. The closest work to ours is the work of (Xia and Liu, 2015). In the latter study of (Xia and Liu, 2015), they proposed a multi-task learning framework that leverages activation and valence information for acoustic emotion recognition.
Our work contributes the following: a) we empirically illustrate that emotion cues can be learned robustly across genres by framing the problem as a Joint Multi-Task learning problem.

7 Conclusion

Combination of different genre datasets can improve and generalize emotion detection in sentences. We showed multi-task deep neural net models are able to successfully classify emotion in multigenre and across genres. We showed that unified annotation is beneficial for emotion detection through combining different genres to augment and create larger training sets. We discuss the impact of pre-trained word embedding in emotion classification and the challenges involve finding a proper word embedding that has the most coverage among different genres in our corpus.
Our future direction is to increase number of instances for minority genres and experiment with formal and informal text to represent different tasks. We aim to experiment with more robust tuning method to create better pre-trained word embedding. Further, we aim to add genre specific layer to improve results across different genres.

References

Sgaard Anders and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Vol. 2*.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*.

Qadir Ashequl and Riloff Ellen. 2013. Bootstrapped learning of emotion hashtags hashtags4you. *WASSA, NAACL-HLT*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks 5.2*, pages 157–166.

Pang Bo and Lee Lillian. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

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

Kyunghyun Cho. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung. 2015. Gated feedback recurrent neural networks. *ICML*, pages 2067–2075.

Ronan Collobert, Jason Weston, Lon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, pages 2493–2537.

Andrew Cotter. 2011. Speech recognition with deep recurrent neural networks. *Better mini-batch algorithms via accelerated gradient methods.*

Terrance Devries, Kumar Biswaranjan, and Graham W. Taylor. 2014. Multi-task learning of facial landmarks and expression. *Computer and Robot Vision (CRV), 2014 Canadian Conference on. IEEE*.

Inkpen Diana and Strapparava Carlo. 2010. Evaluation of unsupervised emotion models to textual affect recognition. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 62–70. Association for Computational Linguistics.

Ghazi Diman, Inkpen Diana, and Szpakowicz Stan. 2010. Hierarchical versus flat classification of emotions in text. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 140–146. Association for Computational Linguistics.

Mishne Gilad. 2005. Experiments with mood classification in blog posts. In *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access*, volume 19, pages 321–327.

Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. *Acoustics, speech and signal processing (icassp), ieee international conference on. IEEE*.

Sepp Hochreiter and Jorgen Schmidhuber. 1997. Long short-term memory. *Neural computation 9.8*, pages 1735–1780.

Godwin Jonathan, Stenetorp Pontus, and Riedel Sebastian. 2016. Deep semi-supervised learning with linguistically motivated sequence labeling task hierarchies. *arXiv preprint arXiv:1612.09113*.

Hashimoto Kazuma, Xiong Caiming, Tsuruoka Yoshimasa, and Socher Richard. 2016. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Miwa Makoto and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

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

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

Abdul-Mageed Muhammad and Ungar Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. *ACL*.

Gözde Özbal and Pighin Daniele. 2013. Evaluating the impact of syntax and semantics on emotion recognition from text. In *Computational Linguistics and Intelligent Text Processing*, pages 161–173. Springer.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *International Conference on Machine Learning*.

Ekman Paul. 1992. An argument for basic emotions. *Cognition and emotion*, 6.3-4:169–200.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

Robert Plutchik. 1962. The emotions: Facts, theories, and a new model. *New York: Random House*.

Caruna R. 1993. Multitask learning: A knowledge-based source of inductive bias. *Machine Learning: Proceedings of the Tenth International Conference*.

Kirk Roberts, Michael A. Roach, Joseph Johnson, Josh Guthrie, and Sanda M. Harabagiu. 2012. Empatweet: Annotating and detecting emotions on twitter. *Vol. 12. LREC*.

Collobert Ronan and Weston Jason. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Mohammad Saif and Kiritchenko Svetlana. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence 31.2*, pages 301–326.

Aman Saima and Szpakowicz Stan. 2007. Identifying expressions of emotion in text. In *International Conference on Text, Speech and Dialogue*, pages 196–205. Springer.

Michael L. Seltzer and Jasha Droppo. 2013. Multi-task learning in deep neural networks for improved phoneme recognition. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*.

Volkova Svitlana and Bachrach Yoram. 2016. Inferring perceived demographics from user emotional tone and user-environment emotional contrast. *ACL (1)*.

Shabnam Tafreshi and Mona Diab. 2018. Sentence and clause level emotion annotation, detection, and classification in a multi-genre corpus. In *Proceedings of LREC 2018*, pages 1246–1251.

Wang Wenbo, Chen Lu, Thirunarayan Krishnaprasad, and Sheth Amit. 2012. Harnessing twitter" big data" for automatic emotion identification. *Privacy, Security, Risk and Trust (PASSAT), International Conference on and 2012 International Confernece on Social Computing (SocialCom). IEEE*.

Rui Xia and Yang Liu. 2015. A multi-task learning framework for emotion recognition using 2d continuous space. *IEEE Transactions on Affective Computing*.

Li Xiao, Ye-Yi Wang, and Gokhan Tur. 2011. Multi-task learning for spoken language understanding with shared slots. *Twelfth Annual Conference of the International Speech Communication Association*.

Liew Jasy Suet Yan and Turtle Howard. 2016. Exploring fine-grained emotion detection in tweets. *NAACL-HLT*.

Liew Jasy Suet Yan. 2014. Expanding the range of automatic emotion detection in microblogging text. *EACL*.