How emotional are you? Neural Architectures for Emotion Intensity Prediction in Microblogs

Devang Kulshreshtha*, Pranav Goel*, and Anil Kumar Singh

Indian Institute of Technology (Banaras Hindu University) Varanasi
Varanasi, Uttar Pradesh, India
{devang.kulshreshtha.cse14, pranav.goel.cse14, aksingh.cse}@iitbhu.ac.in

Abstract

Social media based micro-blogging sites like Twitter have become a common source of real-time information (impacting organizations and their strategies), and are used for expressing emotions and opinions. Automated analysis of such content therefore rises in importance. To this end, we explore the viability of using deep neural networks on the specific task of emotion intensity prediction in tweets. We propose a neural architecture combining convolutional and fully connected layers in a non-sequential manner - done for the first time in context of natural language based tasks. Combined with lexicon-based features and transfer learning, our model achieves state-of-the-art performance, outperforming the previous best system by 4.4% Pearson correlation on the WASSA'17 EmoInt shared task dataset. We investigate the performance of deep multi-task learning models trained for all emotions at once in a unified architecture and get encouraging results. Experiments performed on evaluating correlation between emotion pairs offer interesting insights into the relationship between them. The code for our experiments is publicly available.

1 Introduction & Related Work

One can use text in any language not only to express a variety of emotions but also to convey the associated 'intensity' of the emotion. *Emotion intensity* is the degree of an emotion (like anger or fear) expressed by the speaker or author. For example, *'When you lose somebody close to your heart you lose yourself as well. Crying my heart out!!!'* expresses *more* sadness than, say, *'All friends are out of town. Feeling a bit lonely...'*. Automatic detection of emotion intensity can be useful for natural language based systems. An e-commerce firm seeking to publicize positive reviews of its products would want to use the statements expressing high amount of joy or happiness. The intensity of anger expressed in a grievance can be used to automatically decide the priority of addressing complaints. A possible example scenario: based on the tweet *'X box one controller jockey update is the fucking worst! #fuming'* expressing a higher intensity of anger than the tweet *'New madden bundle code for x box one is stupid, takes forever to download lol'*, may point to the need of prioritizing addressing issues with the new jockey stick before looking into the madden bundle of the same product (an Xbox One).

The WASSA'17 workshop conducted the EmoInt shared task (Mohammad and Bravo-Marquez, 2017b) where given a tweet and the emotion it exhibits, systems had to predict the intensity of this given emotion as a real valued score between 0 and 1. We use the shared task setting and the dataset they provide to develop and evaluate all our approaches. The main contributions of our work are:

a) A neural architecture for emotion intensity detection which combines convolutional layers with fully connected layers in a non-sequential or 'parallel' fashion. Such a combination of these neural models or layers has been explored in computer vision (Antol et al., 2015; Vijayanarasimhan et al., 2017) but, to the best of our knowledge, our work is the first to utilize this in Natural Language Processing based tasks. Our results will motivate the NLP community to take an increased interest in such architectures. We also rely on lexicon-based linguistic information in our network, and use transfer learning by incorporating

* The two first authors have equal contributions to the paper

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

activations of a pre-trained convolutional neural network which uses emojis to learn representations for related tasks of sentiment and emotion detection in tweets (Felbo et al., 2017). Our model achieves *state of the art performance* on the EmoInt dataset.

b) Two multi-task neural network based models which successfully handle all emotions jointly, as opposed to having a separate network for each individual emotion. Our best multi-task learning based approach performs relatively well when compared to other approaches and previous systems while benefiting from better generalization, speed and lesser trainable network parameters.

c) A comprehensive set of ablation tests to show how the various components within our models contribute, to help guide the design of future systems for our task as well as other, related tasks.

d) Experiments to gauge the correlation between different pairs of emotion and an effort to link our observations with past findings in linguistic and psychological experiments.

e) Error analysis to assess reasons for erroneous intensity prediction in tweets.

The code for all our approaches is publicly available [1].

Most datasets and systems focus on emotion classification or detecting the presence or absence of an emotion (Brooks et al., 2013; Alm et al., 2005; Aman and Szpakowicz, 2007; Bollen et al., 2011; Wang et al., 2016). To the best of our knowledge, apart from the dataset used for conducting the EmoInt shared task, there is only one resource annotated for the intensity of emotion in text, which was released by Strapparava and Mihalcea (2007) as part of SemEval 2007. Annotators gave a score between 0 and 100 for the degree of emotion in newspaper headlines using a slide bar in a web interface.

We perform our experiments on a Twitter dataset (Mohammad and Bravo-Marquez, 2017a) (Table 4). Users from many backgrounds express emotions via tweets. Increasing activity of bots on Twitter (Gilani et al., 2017) and companies aiming to accurately respond to tweets about their products or services makes Twitter an attractive domain for sentiment analysis (Fan and Gordon, 2014; Nakov et al., 2016).

Twenty two systems participated in the EmoInt shared task. Approaches included random forest regressors, neural approaches and lexicon-based methods (Mohammad and Bravo-Marquez, 2017b).

2 Proposed Neural Framework: LE-PC-DNN

We propose a novel neural network based architecture that combines - (i) Convolutional layers, (ii) Fully-connected layers, (iii) Linguistic features, and (iv) Pretrained CNN activations in a non-sequential fashion (detailed below). The architecture and the hyperparameter setting is consistent across all emotions.

2.1 Embedding Layer ($l^{(1)}$)

The input to our network's first layer is a sequence of tokens $\{w_1, w_2, ...w_n\}$ (padded when necessary). $l^{(1)}$ begins by associating each word w with a feature vector $\mathbf{e}_w$, also called *word embeddings* (Bengio et al., 2003). The obtained embedding matrix $\mathbf{E} \in R^{n*d}$ serves as the input to next layer.

The model consists of two different types of layers - *Parallelly-connected layers* and *Sequential fully-connected layers* (Figure 1).

2.2 Parallelly-connected layers

The output E of the embedding layer ($l^{(1)}$) above is fed to two parallel layers - the **CNN layer** ($l_a^{(2)}$), which applies convolutional operation on the embedding matrix $\boldsymbol{E}$ followed by a max-pooling-over-time operation to get the $l_a^{(2)}$ layer representation (Figure 1) and the **Average Embedding Layer** ($l_b^{(2)}$), which is calculated by taking the mean of word embeddings $\boldsymbol{E}$ across the length of the tweet, giving us a d-dimensional feature vector. $l_b^{(2)}$ is supposed to capture the global context of the tweet. The CNN layer and the average embedding layer are both fully connected to layers $l_a^{(3)}$ and $l_b^{(3)}$ respectively after applying Dropout (Srivastava et al., 2014) in each case to control over-fitting of parameters.

The network consists of two more parallel layers ($l_c^{(2)}$, $l_d^{(2)}$), which process the input tweet $\boldsymbol{T}$ directly. For emotion intensity prediction task, the **Linguistically Informed Layer** ($l_c^{(2)}$) uses the *TweetToLexiconFeatureVector* filter developed in the Affective Tweets[2] package to get 43 tweet-level features, cal-

[1] https://github.com/Pranav-Goel/Neural_Emotion_Intensity_Prediction

[2] https://github.com/felipebravom/AffectiveTweets

culated using several lexicons (details in Mohammad and Bravo-Marquez (2017b)). Sentiment-based lexicons and other resources of linguistic knowledge can capture some aspects of data that are different than those learned by CNN/LSTM relying on word embeddings (Chen et al., 2017) and boost performance (Ebert et al., 2015). There are different ways to use linguistic features with neural architectures (Ebert et al., 2015; Park et al., 2016). We incorporate these external features in a simple manner (Figure 1). The input to the layer $l_c^{(2)}$ is a tweet T. Applying TweetToLexiconFeatureVector filter gives a 43-dimensional feature vector.

We also use **Pretrained CNN features** ($l_d^{(2)}$) by leveraging the activations of a Convolutional Neural Network (CNN) pre-trained on emoji prediction task called DeepMoji (Felbo et al., 2017) [3]. This CNN was trained on 1.3 billion emoji-containing tweets and tested on eight benchmark datasets within emotion, sarcasm and sentiment detection. Since emoji prediction is closely related to emotion intensity prediction, we hypothesize that transferring knowledge via pre-trained CNN activations could help improve performance for our task. Each tweet is converted into a 2304-dimensional feature vector $l_d^{(2)}$ by feeding the tweet into the DeepMoji-CNN and extracting activations of the last hidden layer.

2.3 Sequential Fully-Connected Layers ($l^{(4)}, l^{(5)}, l^{(6)}$)

The layer ($l^{(4)}$) is obtained by concatenating the feature activations of layers $l_a^{(3)}, l_b^{(3)}, l_c^{(2)}, \& l_d^{(d)}$. $l^{(4)}$ is connected to $l^{(5)}$, which is a fully-connected layer. Additionally, a linear hidden layer ($l^{(6)}$) with lesser number of neurons is introduced on top of $l^{(5)}$.

2.4 Output Layer ($\hat{y}$)

Finally, we use a sigmoid neuron after $l^{(6)}$ to compute the intensity score $\hat{y}$ of between 0 and 1.

It is quite common to have fully connected layers or small feedforward neural networks connected in a sequential or hierarchical manner to sequence-to-sequence models like CNNs/LSTMs (Lee and Dernoncourt, 2016; Plahl et al., 2013). The feedforward networks in this case take in the pooled output of CNNs/LSTMs as input. Our architecture (Figure 1), however, combines feedforward, convolutional layers and manually extracted features in a non-hierarchical manner (layer $l_a^{(3)}$) as well as in hierarchical fashion (layer $l_b^{(3)}$). Such a non-sequential combination of fully-connected and convolutional networks is *novel* for NLP tasks, to the best of our knowledge (though this has been used by the vision community (Antol et al., 2015; He et al., 2017; Vijayanarasimhan et al., 2017)). We call this neural framework with a non-hierarchical or *parallel* combination of CNNs/LSTMs, fully-connected layers, lexical features and pretrained CN activations as **PC-DNN** (Parallel Combination of Deep Neural Networks). As it incorporates **L**inguistic features (layer $l_c^{(2)}$) and **E**moji-based pretrained CNN activations (layer $l_d^{(2)}$) as well, we will refer to the network discussed in this section as **LE-PC-DNN** (Figure 1).

3 Deep Multi-Task Learning (DMTL): Handling all Emotions in a Unified Architecture

Multi-task Learning (Caruna, 1993; Caruana, 1998) has resulted in successful systems for various NLP tasks (Collobert and Weston, 2008), especially in cross-lingual settings (Huang et al., 2013). MTL was first applied to Emotion Intensity prediction in (Goel et al., 2017). In their neural architecture, the initial network layers are shared across multiple emotions upto a certain point, after which the architecture gets diverged for each emotion. The objective is to jointly train on different emotion datasets such that initial layers increase generalization and the individual final layers can learn task specific features. The network input features are the concatenation of average word embeddings across the length of the tweet and linguistic features (using *Tweet2LexiconFeatureExtractor*), and the neural layers were fully-connected.

3.1 LE-PC-DMTL for Emotion Intensity Prediction

The above DMTL model only uses feed-forward layers as the input is a 1-dimensional representation of the tweet. Our proposed LE-PC-DNN network, on another note includes convolutional layers and pre-trained CNN activations in the architecture. The model however is separate for each emotion. We

[3] https://github.com/bfelbo/DeepMoji

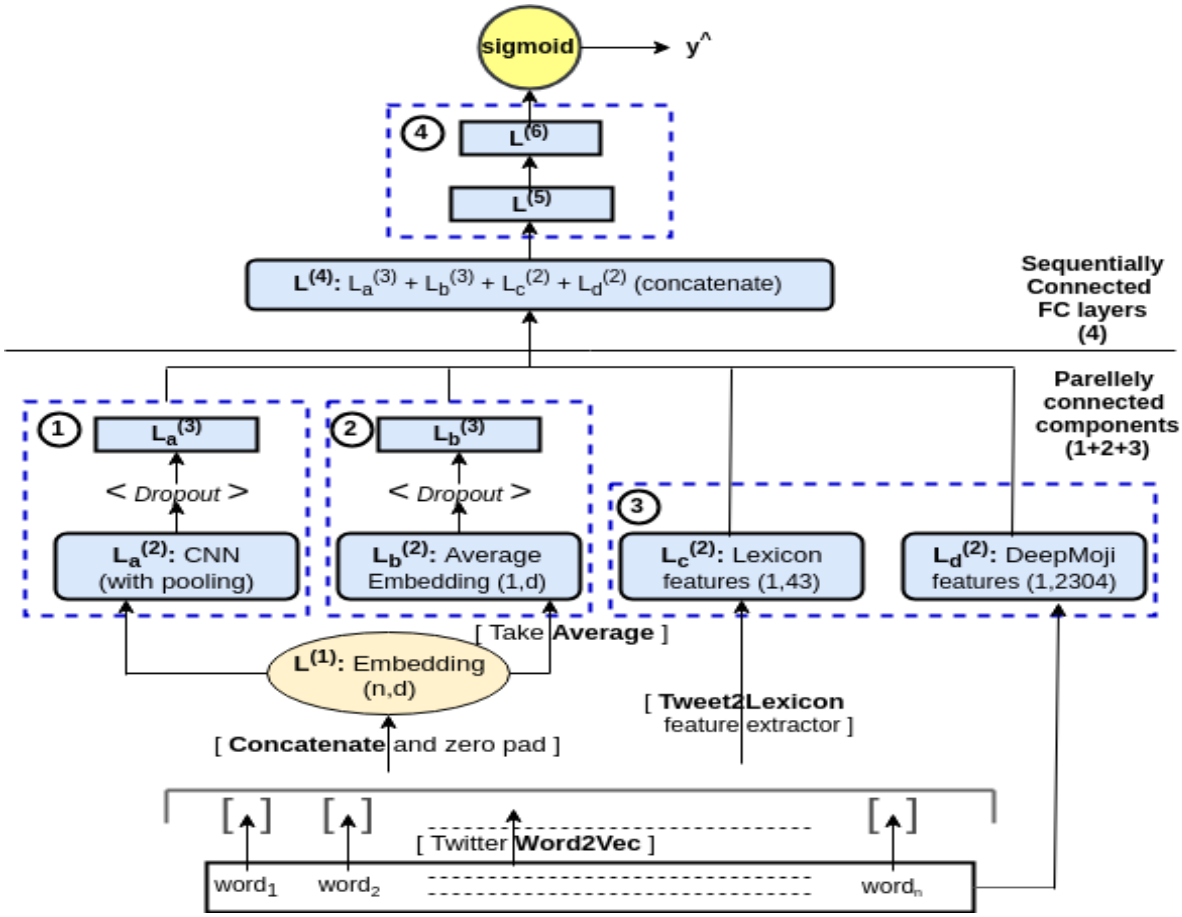

Figure 1: Network architecture for the **LE-PC-DNN** model: Various components - ***L***exicon-based features, an ***E***moji detecting CNN's pre-trained activations, fully connected layers, and a CNN/LSTM undergo a ***P***arallel ***C***ombination (parallelly-connected) to form a ***D***eep ***N***eural ***N***etwork (Section 2).

can combine the different emotion models of LE-PC-DNN architecture into a unified network by using the idea of Multi-Task learning proposed in (Goel et al., 2017). In order to that, we adopt the initial architecture of LE-PC-DNN and then the final fully-connected layers are different for each emotion.

Our basic model (Figure 2a) consists of two different types of layers: *shared representation layers* and *task-specific representation layers*. The network receives a tweet-emotion pair (T, e) and produces an intensity score between 0 and 1.

3.1.1 Shared Representation Layers

These layers receive tweets from all emotions which helps in achieving better generalization as the layers are forced to learn features which work for multiple subtasks. Similar to LE-PC-DNN (Figure 2a), there are four such levels of layers - $l^{(1)}$ (Embedding layer), $l^{(2)}$ (CNN layer, Avg. Embedding layer, Linguistic layer, Pretrained CNN features), $l^{(3)}$ (Fully-connected layers), and $l^{(4)}$ (concatenation layer).

3.1.2 Task-specific Representation Layers

The last two layers are allowed to be different across the different sub-tasks to learn task-specific features.

1. **Fully Connected Layers ($l^{(5)}, l^{(6)}$):** For every emotion, a ReLu transformation will map shared layer $l^{(4)}$ to task-specific layers $l^{(5)}$ ($l_a^{(5)}, l_f^{(5)}, l_j^{(5)}, l_s^{(5)}$ - separate for each emotion), with the separate outputs going to another hidden layers - $l_a^{(6)}, l_f^{(6)}, l_j^{(6)}, l_s^{(6)}$.

2. **Output Layer ($\hat{y}$) :** $l^{(6)}$ is connected to a single sigmoid output neuron – $\hat{y}$, which gives the predicted intensity score between 0 and 1.

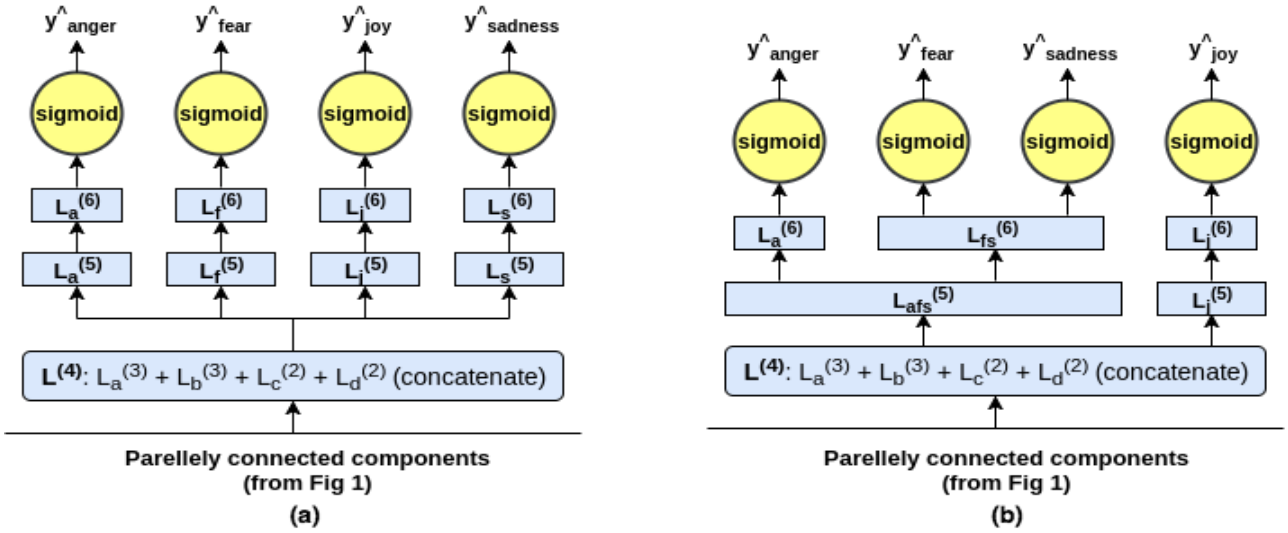

Figure 2: **(a) LE-PC-DMTL**: A deep multi-task learning neural network that uses parallelly connected components - a CNN/LSTM, fully connected layers, a feature vector derived from linguistic lexicons and pre-trained activations of a network trained for Emoji detection (DeepMoji). So, a ***L***exicon and ***E***moji detection based, ***P***arallelly ***C***onnected ***D***eep ***M***ulti-***T***ask ***L***earning neural network and **(b) LE-PC-DMTL-EI**: This network branches differently to exploit the different pairwise correlations between the different emotions (some emotion pairs sharing more layers than others), to optimize performance. So, a ***L***exicon and ***E***moji detection based, ***P***arallelly ***C***onnected ***D***eep ***M***ulti-***T***ask ***L***earning neural network, optimized specifically for the task of ***E***motion ***I***ntensity detection. Both these models handle all the four emotions together in a single architecture (Section 3).

Figure 2a clearly demonstrates that replacement of individual bottom layers with shared layers reduces the network's parameters significantly, less hyper-parameters required to be fine-tuned and speedup during training the network compared to the scenario of having separate networks for each emotion.

3.2 LE-PC-DMTL-EI: A Task-Optimized DMTL Model

The architecture of our proposed LE-PC-DMTL model is that the shared layers will be common across all emotions, *irrespective* of the degree of correlation between any particular pair. The correlations between emotions are investigated via experiments described in section 6. We would like the model to incorporate or *learn* relationships between different emotions and not treat all emotions or subtask in a similar manner since we cannot expect all emotions to be related to each other in the same way.

Inspired by Lu et al. (2016) and some results discussed in section 6, we propose to learn a network architecture which *allows grouping of similar emotions and separate branching of unrelated tasks*. Here we remove the constraint of feature sharing up to a certain level and allow sharing of more parameters between correlated emotions.

Learning the Network Architecture: We initialize the network with our basic LE-PC-DMTL model (Figure 2a). Our objective is to learn an architecture where similar emotions are grouped together and uncorrelated emotions are branched out. After initialization, we refine the architecture step-by-step by experimenting with various combinations. Since manual exploration of such a combinatorially large space is not possible, we follow two heuristics:

1. A model is favored over other if it gives a higher Pearson correlation on the cross-validation set.

2. Models where emotion pairs having high positive correlation (Section 6) share more parameters.

The refinement process is repeated manually for fixed iterations or until no further gain in Pearson correlation is observed (on development set). Since we start from our basic multi-task model, the final model would definitely be an improved version. We apply this heuristic-based algorithm to the EmoInt shared

Emotion	Train	Dev	Test	Total
Anger	857	84	760	1701
Fear	1147	110	995	2252
Joy	823	74	714	1611
Sadness	786	74	673	1533
Total	3613	342	3142	7097

Table 1: The number of instances (tweets) in the EmoInt shared task dataset

Model /	**LE-PC-DNN**					**LE-PC-DMTL**		**LE-PC-DMTL-EI**	
Emotion	$L_a^{(2)}$	$L_a^{(3)}$	$L_b^{(3)}$	$L^{(5)}$	$L^{(6)}$	$L_x^{(5)}$	$L_x^{(6)}$	$L_x^{(5)}$	$L_x^{(6)}$
Anger	CNN (250,Max)	128	256	128	32	64	32	256	64
Fear						75	40		75
Sadness						64	32		
Joy						40	20	125	50

Table 2: Summary of network hyperparameters for our proposed models (Figures 1 and 2).

task dataset (Mohammad and Bravo-Marquez, 2017b). The resulting architecture is shown in Figure 2b. Though the algorithm is general, the resulting architecture is meant to be optimal for the task to which it is applied. We refer to this resulting architecture as **LE-PC-DMTL-EI** (**LE-PC-DMTL** for **E**motion **I**ntensity prediction). Note that the pairwise correlations (heuristic 2) is only a guiding principle - the multi-task model discovers the optimal branching or sharing of layers based on the performance across a lot of variations. The captions of Figures 1 and 2 also serve to summarize the three models we propose.

4 Experimental Setting

4.1 Data

We use the *training*, *development*, and *testing* datasets provided within the WASSA EmoInt shared task (Mohammad and Bravo-Marquez, 2017b) to train and evaluate the various approaches (Table 4). The data files include the tweet ID, the tweet, the emotion of the tweet and the emotion intensity. Details regarding creation and annotation can be found in Mohammad and Bravo-Marquez (2017a).

4.2 Implementation Details

Word Embeddings: We use publicly available pre-trained word embeddings called the Twitter word2vec model (Godin et al., 2015). These were trained on 400 million tweets for the ACL-WNUT 2015 shared task (Baldwin et al., 2015) using the word2vec approach (Mikolov et al., 2013). The large number of tweets in training data makes it a better choice than others available. We choose it over other pre-trained models like Google word2vec (Mikolov et al., 2013) and the GloVe (Pennington et al., 2014) Twitter model. This was backed by cross validation results (not shown for brevity).

Preprocessing: Before forming word vector based embeddings of the Twitter tweets, we employed some preprocessing steps including removal of URLs and user mentions, stripping punctuations from word boundaries, hashtag segmentation ('#wearethebest' to 'we are the best' using the Word S.egment [4] module in Python) and elongation removal ('goooooood' to 'good') (inspired by the work in Akhtar et al. (2017)). Such preprocessing helped in improving cross-validation performance.

Network Hyper-Parameters and Architecture Settings: We show the final hyper-parameter values for the LE-PC-DNN (section 2), and the multi-task models LE-PC-DMTL and LE-PC-DMTL-EI (section 3) in Table 2. These values were chosen on the basis of 7-fold cross validation results on the combined training and validation sets. We changed various network hyper-parameters like number of layers, output dimensionality, the type of pooling for CNNs (max versus average) and dropout value.

[4] https://github.com/grantjenks/wordsegment

Training: The network parameters are learned by directly minimizing the Mean Absolute Error between the actual and predicted intensity values. We optimize the above function by back-propagating through layers via Mini-batch Gradient Descent. We use a batch size of 8, 25-30 training epochs and Adam optimization algorithm (Kingma and Ba, 2014) with the parameters set as $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999\ and\ \epsilon = 10^{-9}$. These settings are common across all our models.

4.3 Evaluation

We evaluate the performance of all systems using the Pearson correlation coefficient (r) between gold ratings and predicted output intensities. This was the metric used in the EmoInt shared task. The Pearson coefficient measures linear correlation between two variables and is always between -1 and 1. A high (positive) value indicates strong (positive) correlation and a value of 0 indicates no correlation. We use following baselines to benchmark our results:

1. *Weka*: The Weka system (Mohammad and Bravo-Marquez, 2017a) served as the baseline for EmoInt shared task (Mohammad and Bravo-Marquez, 2017b) . It uses an SVM regressor on pre-trained word embeddings and lexicon based features.

   The other two baselines are the two top performing systems in the shared task:

2. *Prayas*: (Goel et al., 2017) used weighted ensembling technique to combine 3 different models - feedforward neural network on average word embeddings and sentiment features, CNN/LSTM+max-pooling on concatenated word embeddings, and deep multi-task learning model (described in Section 3). Note that they combine only the outputs (predicted intensities) of their different models, but the fact that combining outputs of different models helped with performance indicates that the architectures are able to complement each other. Our model (Figure 1) exploits this observation by letting the network learn how to best combine different neural models.

3. *IMS*: (Köper et al., 2017) use a random forest regressor on features based on manually created resources, automatically extended lexicons and the output of CNNs/LSTMs.

5 Results and Discussion

We show the Pearson correlation scores for all our proposed neural models and baselines (section 4.3) on the test set in Table 3. The major takeaways from these results and the ablation tests (Figure 3) are:

- **State-of-the-Art Performance** The proposed LE-PC-DNN model (section 2) outperforms all the baselines for all emotions (except for sadness in which our proposed multi-task architecture LE-PC-DMTL-EI fares better), setting a new state of the art score of **0.791** on average. It significantly outperforms the previous best model (Prayas) by 4.4% Pearson score on average.

- **A Solid Case for Multi-Task Learning for Emotion Intensity Prediction**: While the multi-task architecture used in the system Prayas (Goel et al., 2017) used just fully-connected layers, our proposed multi-task models uses CNNs, lexicon-based features and activations of a pre-trained CNN (section 3, Figure 1) to improve performance by a comprehensive 9.6-11.0% Pearson score (the last 3 columns of Table 3) and also gives competitive scores when compared to our state-of-the-art

Emotion	LE-PC-DNN	Prayas	IMS	Weka	LE-PC-DMTL-EI	LE-PC-DMTL	MTL (Prayas)
Anger	**0.767**	0.765	0.767	0.639	0.735	0.731	0.645
Fear	**0.791**	0.732	0.705	0.652	0.760	0.755	0.677
Joy	**0.803**	0.762	0.726	0.654	0.785	0.758	0.654
Sadness	0.803	0.732	0.690	0.648	**0.808**	0.786	0.672
Average	**0.791**	0.747	0.722	0.648	0.772	0.758	0.662

Table 3: Pearson correlations (r) obtained by the systems on the full test sets.

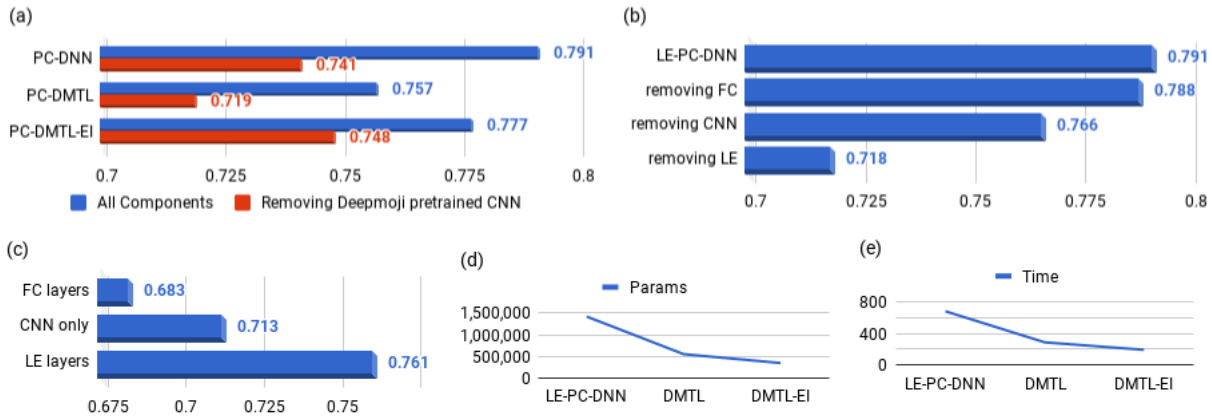

Figure 3: **Ablation test visualizations (scores are average Pearson correlation**: a) Effect of Transfer Learning, b) Effect of removing one of the parallel connected components from LE-PC-DNN architecture (Figure 1), c) Comparing individual parallel components of LE-PC-DNN (section 2.2), d) Comparing number of trainable parameters and e) training time with proposed multi-task learning models.

LE-PC-DNN model trained on and used for each emotion individually. Out of our two multi-task models, LE-PC-DMTL-EI (section 3.2) gives better performance by 1.4% Pearson score. It benefits from being optimized for our task by exploiting the correlation among certain pairs of emotions. The major advantage of using the multi-task models offer is the use of a unified architecture instead of separate models for each emotion, which means *less computational complexity*. This is established in Figures 3d and 3e, which show that our LE-PC-DMTL and LE-PC-DMTL-EI architectures (section 3) have less than one-third number of trainable parameters and take less than one-third the training time compared to our LE-PC-DNN model (section 2). This established the viability of our multi-task approach, which will only increase with the number of different emotions!

- **Effect of Transfer Learning**: Removing the deepmoji *pre-trained CNN activations* (section 2.2) has a significant effect of performance of all our proposed models including the multi-task ones (Figure 3a), with the drop of 5% Pearson score in case of the PC-DNN model (section 2) being the most notable. Transferring features from a related task (like emoji detection) trained on a much larger dataset (1.3 billion tweets compared to about just 1,000 in our case) proves to be very useful in terms of prediction performance. This also means that the representations learned by the Deepmoji model (Felbo et al., 2017) work well for the emotion intensity detection task.

- **Impact of the Different Non-Sequential Components in our Neural Architecture**: From Figures 3b and 3c, it is clear that all of our parallelly connected components (components 1, 2 and 3 in Figure 1) have a positive role in the overall performance. Lexicon-based features and pre-trained activations of a CNN trained for Emoji detection task (section 2; component 3 in Figure 1) contribute the most followed by CNN and then the fully-connected layers.

Note: For the training time results shown in Figure 3e, the experiments were run on a CPU with 4 GB RAM and 1.7 GHz Intel core i5 processor. Also, the time for LE-PC-DNN is the sum of the times taken for training of the individual models (since the architecture is run separately for each emotion unlike multi-task models which handle all emotions in one architecture).

6 Cognitive Implications of Emotions' Pairwise Similarity

To quantify various correlations that different pairs of emotions exhibit in social media microblogs, we perform two experiments. First, we train the best-performing LE-PC-DNN model on one emotion and test on another emotion (Table 4) (also explored in Mohammad and Bravo-Marquez (2017a)). Then, we train the model on the combined datasets of two emotions and see the performance on the test sets of individual emotions (Table 5) to further test how well different pairs of emotions complement each other.

Emotion i/j	anger	fear	joy	sadness
anger	0.766	0.447	-0.526	0.443
fear	0.533	0.791	-0.57	0.712
joy	-0.592	-0.387	0.803	-0.537
sadness	0.586	0.628	-0.567	0.803

Table 4: Pearson correlation scores (trained on emotion *j* (row); tested on emotion *i* (column)).

Emotion i/j	anger	fear	joy	sadness
anger	-	0.779	0.66	0.79
fear	0.73	-	0.687	0.802
joy	0.637	0.675	-	0.728
sadness	0.745	0.787	0.677	-

Table 5: Pearson correlation scores (trained on combined data of emotion *i*, *j*; tested on *i*).

6.1 Training on One Emotion and Testing on Another

Table 4 shows that there exists significant correlations between all the emotions, ranging from absolute Pearson scores of 0.387 (between fear and joy) to 0.803. This indicates that *all emotions are at least somewhat correlated*. This is backed in part by observations made in Wilson-Mendenhall et al. (2013), where the authors claim that 'all human emotions share core affective properties' (note that they relied on arousal and valence of emotions).

The negative emotions ('anger', 'fear' and 'sadness') are *positively correlated with each other* and *negatively correlated with 'joy'*. The *correlations are asymmetric*, *i.e.*, a pair of emotions does not show similar predictive power in either direction. For example, training on anger predicts intensity of sadness with 0.443 correlation with actual sadness intensities (a drop of about 0.36 Pearson score from when training happens on the sadness training set), whereas training on sadness predicts intensity of anger with 0.586 correlation (a drop of about 0.22 Pearson score). These last two observations match well with the results of the experiments carried out in Rutherford et al. (2008). In that work, visual aftereffects of certain emotional scenarios on the participants' faces were observed in various settings. They found that the results of their functional approach suggests (in their own language): "Negative emotions are many and specific. In contrast, positive emotions are few and less specific." The results further backed their hypothesis that "emotions share an asymmetric relationship as a group, in which numerous negative emotions together oppose the relatively small set of positive emotions". Schwartz and Weinberger (1980) also hint at the asymmetric nature of emotions (for fear and anger in particular).

6.2 Training on a Combined Dataset of Two Emotions and Testing on Both of them

Table 5 investigates the impact of combining the datasets of two emotions for training. All negative emotions (anger, fear and sadness) exhibit good performance in this scenario. The Pearson correlation suffers a drop of 0.02-0.03 in case of anger, and training on combined datasets of fear and sadness results in very similar performance for the individual emotions as compared to using just the individual emotion's dataset for training (the results in the diagonal of Table 4). The idea for using combined emotion datasets was motivated in Mohammad and Bravo-Marquez (2017a), who hinted at an increase in performance when combining fear and sadness training sets (but we do not see an increase, although the performance remains basically unaffected). High correlation among the negative emotions was also established in Barrett (2006) and Feldman (1993). However, the specific relationship between fear and sadness seems to be unexplored. It is possible that this relationship is a characteristic of the domain of Twitter tweets or our dataset. These observations comply with our LE-PC-DMTL-EI architecture (Figure 2b), where the emotion 'joy' is the first to get separated while the emotions 'fear' and 'sadness' are learned jointly for many layers in the architecture which gave optimal performance.

7 Error Analysis

We present some examples from the test dataset (Table 6) where the predictions were off by about 0.3 in intensity in either direction. Note that intensities are real-values scores between 0 and 1. To help future systems better predict emotion intensity in tweets, we present plausible explanations for the errors:

- **Incorrect Modeling of the Full Tweet Context**: It can become necessary to correctly model the overall context of the tweet. In Tweet #2, words like 'angry','anger' and 'hurt' indicate high intensity of anger. But the tweet is more of a moral preaching than an angry outburst, which explains an

#	Tweet	Emotion	Actual Intensity	Predicted Intensity
1	Oi @THEWIGGYMESS you've absolutely fucking killed me.. 30 mins later im still crying with laughter.. Grindah.. Grindah...hahahahahahaha	Joy	0.846	0.417
2	People are #hurt and #angry and it's hard to know what to do with that #anger Remember, at the end of the day, we're all #humans #bekind	Anger	0.25	0.580
3	T minus 10 hours till I meet with a designer who wants me to model his new fashion line !!!	Fear	0.667	0.286
4	Ibiza blues hitting me hard already wow	Sadness	0.833	0.533

Table 6: Examples of tweets with absolute difference in predicted and actual intensity greater than 0.3

actual anger intensity of 0.25. In Tweet #1, the latter half of the tweet makes it clear that 'absolutely fucking killed me' is a strong expression of joy and not some negative emotion.

- **Context Outside the tweet**: Tweet #3 requires common sense or world knowledge to know that an important meeting drawing near is generating anxiety with high intensity. This fear or anxiety is not obvious from the text itself which might explain the predicted intensity of 0.291 when the actual intensity is 0.667. Tweet #4 suffers from a similar problem. World knowledge is required to know the relationship between 'Ibiza blues' and sadness.

- **Metaphors**: Metaphorical expressions, like 'crying with laughter' in Tweet #1 or 'Ibiza blues' in Tweet #4, pose a significant challenge to computational models (Ghosh et al., 2015).

We carried out the above analysis after concluding all the experiments. The number of tweets where the predicted intensity was higher than actual and vice-versa were almost equal for any emotion. Hence, our system is not biased towards predicting higher or lower intensities than the annotated ones.

8 Conclusion & Future Work

We proposed various deep neural architectures for the task of emotion intensity prediction on micro-blogging data, specifically, Twitter data. This problem was recently brought to light by the EmoInt shared task (Mohammad and Bravo-Marquez, 2017b). Combining fully-connected layers with CNN in a parallel or non-sequential manner helped performance, and while we see such combinations in works tackling Vision tasks, the NLP community should also try this in their experiments. Transfer learning proved very effective in terms of performance since it helped us utilize the pre-trained activations of a CNN trained on more than a billion tweets for the related task of emoji detection in the same domain of Twitter tweets. These techniques when combined allow us to set the new state-of-the-art performance in emotion intensity detection, considerably outperforming all previous systems. We also proposed two deep neural network based multi-task learning methods with one of them learning to group similar emotions together using a heuristic based algorithm. The models compare well with the best performing system, and give significant gains in terms of computational complexity when compared to having a separate architecture for each emotion. Our multi-task model will be even more useful when the number of emotions are large. The optimized multi-task model and focused correlation tests brought out an especially strong correlation between 'fear' and 'sadness' for our dataset. Our error analysis on the test set using predictions from the best performing system will help future systems aiming to predict emotion intensity in tweets.

For future work, we would like to exploit topic modeling to find out correlations between certain topics and high intensity emotional outbursts, and to see if knowing the topic of a tweet can help with intensity prediction. We also plan to exhibit our models' effectiveness in real-world applications by using them, for example, to rank product reviews in terms of priority on basis of the emotion intensity. Another planned direction is to extend the four emotions used in this task to Ekman's six basic emotions (anger, happiness, surprise, disgust, sadness, and fear) (Ekman and Friesen, 1971).

References

Md Shad Akhtar, Palaash Sawant, Asif Ekbal, Jyoti Pawar, and Pushpak Bhattacharyya. 2017. Iitp at emoint-2017: Measuring intensity of emotions using sentence embeddings and optimized features. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 212–218.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. 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. Association for Computational Linguistics.

Saima Aman and Stan Szpakowicz. 2007. Identifying expressions of emotion in text. In *Text, speech and dialogue*, pages 196–205. Springer.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433.

Timothy Baldwin, Marie Catherine De Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015), Beijing, China*.

Lisa Feldman Barrett. 2006. Are emotions natural kinds? *Perspectives on psychological science*, 1(1):28–58.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Johan Bollen, Huina Mao, and Alberto Pepe. 2011. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *ICWSM*, 11:450–453.

Michael Brooks, Katie Kuksenok, Megan K Torkildson, Daniel Perry, John J Robinson, Taylor J Scott, Ona Anicello, Ariana Zukowski, Paul Harris, and Cecilia R Aragon. 2013. Statistical affect detection in collaborative chat. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 317–328. ACM.

Rich Caruana. 1998. Multitask learning. In *Learning to learn*, pages 95–133. Springer.

R Caruna. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 41–48.

Ruey-Cheng Chen, Evi Yulianti, Mark Sanderson, and W. Bruce Croft. 2017. On the benefit of incorporating external features in a neural architecture for answer sentence selection. In *Proceedings of SIGIR '17*, pages 1017–1020. ACM.

Ronan Collobert and Jason Weston. 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.

Sebastian Ebert, Ngoc Thang Vu, and Hinrich Schütze. 2015. A linguistically informed convolutional neural network. In *WASSA@ EMNLP*, pages 109–114.

Paul Ekman and Wallace V Friesen. 1971. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124.

Weiguo Fan and Michael D Gordon. 2014. The power of social media analytics. *Communications of the ACM*, 57(6):74–81.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625.

Lisa A Feldman. 1993. Distinguishing depression and anxiety in self-report: Evidence from confirmatory factor analysis on nonclinical and clinical samples. *Journal of consulting and clinical psychology*, 61(4):631.

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. 2015. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 470–478.

Zafar Gilani, Reza Farahbakhsh, and Jon Crowcroft. 2017. Do bots impact twitter activity? In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 781–782. International World Wide Web Conferences Steering Committee.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab@ acl w-nut ner shared task: named entity recognition for twitter microposts using distributed word representations. *ACL-IJCNLP*, 2015:146–153.

Pranav Goel, Devang Kulshreshtha, Prayas Jain, and Kaushal Kumar Shukla. 2017. Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 58–65.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE.

Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7304–7308. IEEE.

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

Maximilian Köper, Evgeny Kim, and Roman Klinger. 2017. Ims at emoint-2017: Emotion intensity prediction with affective norms, automatically extended resources and deep learning. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–57.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 515–520.

Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. 2016. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. *arXiv preprint arXiv:1611.05377*.

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

Saif M Mohammad and Felipe Bravo-Marquez. 2017a. Emotion intensities in tweets.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017b. WASSA-2017 shared task on emotion intensity. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, Copenhagen, Denmark.

Preslav Nakov, Sara Rosenthal, Svetlana Kiritchenko, Saif M Mohammad, Zornitsa Kozareva, Alan Ritter, Veselin Stoyanov, and Xiaodan Zhu. 2016. Developing a successful semeval task in sentiment analysis of twitter and other social media texts. *Language Resources and Evaluation*, 50(1):35–65.

Eunbyung Park, Xufeng Han, Tamara L Berg, and Alexander C Berg. 2016. Combining multiple sources of knowledge in deep cnns for action recognition. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Christian Plahl, Michael Kozielski, Ralf Schlüter, and Hermann Ney. 2013. Feature combination and stacking of recurrent and non-recurrent neural networks for lvcsr. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6714–6718. IEEE.

MD Rutherford, Harnimrat Monica Chattha, and Kristen M Krysko. 2008. The use of aftereffects in the study of relationships among emotion categories. *Journal of Experimental Psychology: Human Perception and Performance*, 34(1):27.

Gary E Schwartz and Daniel A Weinberger. 1980. Patterns of emotional responses to affective situations: Relations among happiness, sadness, anger, fear, depression, and anxiety. *Motivation and emotion*, 4(2):175–191.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

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

Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. 2017. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*.

Bo Wang, Maria Liakata, Arkaitz Zubiaga, Rob Procter, and Eric Jensen. 2016. Smile: Twitter emotion classification using domain adaptation. In *CEUR Workshop Proceedings*, volume 1619, pages 15–21. Sun SITE Central Europe.

Christine D Wilson-Mendenhall, Lisa Feldman Barrett, and Lawrence W Barsalou. 2013. Neural evidence that human emotions share core affective properties. *Psychological science*, 24(6):947–956.