PlusEmo2Vec at SemEval-2018 Task 1: Exploiting emotion knowledge from emoji and #hashtags

Ji Ho Park, Peng Xu, Pascale Fung
Centre for Artificial Intelligence Research (CAiRE)
Hong Kong University of Science and Technology
{jhpark, pxuab}@connect.ust.hk, pascale@ece.ust.hk

Abstract

This paper describes our system that has been submitted to SemEval-2018 Task 1: Affect in Tweets (AIT) to solve five subtasks. We focus on modeling both sentence and word level representations of emotion inside texts through large distantly labeled corpora with emojis and hashtags. We transfer the emotional knowledge by exploiting neural network models as feature extractors and use these representations for traditional machine learning models such as support vector regression (SVR) and logistic regression to solve the competition tasks. Our system is placed among the Top3 for all subtasks we participated.

1 Introduction

Finding a good representation of texts is very challenging since texts are sequences of words which are represented in a discrete space of the vocabulary. For this reason, many past works have investigated in finding the mapping of words (Mikolov et al., 2013; Pennington et al., 2014) or sentences (Kiros et al., 2015) to continuous spaces, so that each text can be represented by a fixed-size, real-valued N-dimensional vector. This vector representation then can be applied to machine learning models to solve problems like classification and regression. A good representation should contain essential information inside each text and be a useful input for statistical models.

Emotions in texts further deepen the complexity of modeling natural language since they not only depend on the semantics of a language but also are inherently subjective and ambiguous. Despite the difficulty, accounting for emotion is important in achieving true natural language understanding, especially in areas involving human-computer interactions such as dialogue systems (Fung, 2015).

Humans can naturally capture and express different emotions in texts, so machines should also be able to infer them. Many works (Tang et al., 2014; Felbo et al., 2017; Thelwall, 2017) explored modeling sentiment or emotion in texts in various directions. This work is highly related to these efforts.

Semeval-2018 Task 1: Affect in Tweets (AIT-2018) encourages more efforts in this area with the task of sentiment analysis, which is one of the most practical applications of modeling emotional text representations. We have participated in five subtasks regarding English tweets: emotion intensity regression, emotion intensity ordinal classification, valence (sentiment) regression, valence ordinal classification, and emotion classification (More details on the tasks in Mohammad et al. (2018)).

Although these five tasks take different formats, the most important objective is finding a good representation of the tweets regarding emotions. However, the given competition training datasets are too small to achieve our goal (Table 3). Therefore, we explore utilizing larger datasets that are distantly supervised by emojis and hashtags to learn a robust representation and transfer the knowledge of each dataset to the competition datasets to solve the tasks. We aim to minimize the use of lexicons and linguistic features by replacing them with continuous vector representations.

2 Emoji sentence representations

Thanks to the endless stream of social media such as Twitter and Facebook, researchers nowadays are lucky enough to have access to almost an unlimited number of texts generated every day. Nevertheless, annotating these texts with explicit emotion or sentiment human labels is very expensive and difficult. For this reason, many works naturally focused on finding direct or indirect evidence of emotion inside each text, such as hash-

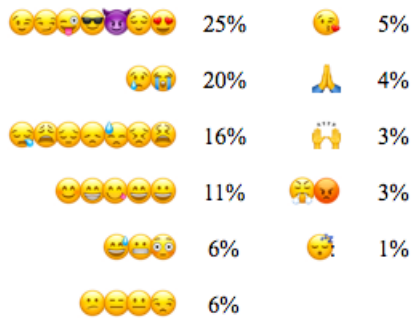| | 25% | | 5% |
	20%		4%
	16%		3%
	11%		3%
	6%		1%
	6%		

Figure 1: 11 clusters of emojis used as categorical labels and their distributions in the training set. Because some emojis appear much less frequently than others, we group the 34 emojis into 11 clusters according to the distance on the correlation matrix of the hierarchical clustering from Felbo et al. (2017) and use them as categorical labels

tags and emoticons (Suttles and Ide, 2013; Wang et al., 2012), and found them useful to distantly label an emotion of each text. Furthermore, the recent popular culture of using emojis (Wood and Ruder, 2016) inside social media posts and messages provides us even richer evidence of different emotions, and they have been proven to be very effective in learning rich representations for various affect-related tasks (Felbo et al., 2017).

2.1 Methodology & Emoji Dataset

In this paper, we compare two models using two different emoji dataset to transform the competition data into robust sentence representations.

First model is the pre-trained DeepMoji model (Felbo et al., 2017), which is trained through emoji predictions on a dataset of 1.2 billion tweets with 64 common emoji labels. We use the pretrained deep learning network, which consists of Bidirectional Long Short Term Memory (Bi-LSTM) with attention, except the last softmax layer, as a feature extractor of the original competition datasets. As a result, each sample is transformed into a 2304-dimensional vector from the model.

The second model is our proposed emoji cluster model. We crawled 8.1 million tweets with each of which has 34 different facial and hand emojis, assuming these kinds of emojis are more relevant to emotions. Since some emojis appear much less frequently than others, we cluster the 34 emojis into 11 clusters (Figure 1) according to the distance on the correlation matrix of the hierarchical clustering from Felbo et al. (2017). Sam-

ples with emojis in the same cluster are assigned the same categorical label for prediction. Samples with multiple emojis are duplicated in the training set, whereas in the dev and test set we only use samples with one emoji to avoid confusion. We then train a one-layer Bi-LSTM classifier with 512 hidden units to predict the emoji cluster of each sample. We take part of the dataset to construct a balanced dev set with 15,000 samples per class (total 165,000) for hyperparameter tuning and early stopping. We use 200 dimension Glove vectors pre-trained on a much larger Twitter corpus to initialize the embedding layer.

The motivation for exploring two different models is that, firstly, we want to replicate the effectiveness of using emoji for representing emotions from the previous work (Felbo et al., 2017) with a smaller dataset and a simpler model. Note that the dataset size of the emoji cluster model is less than 1% of that of the first model, whereas DeepMoji uses more than 1 billion training samples. Moreover, the first model implements a two-layer Bi-LSTM with self-attention, which has much more parameters than the second model's simple one-layer Bi-LSTM does. Secondly, we want to verify that ensembling both emoji representations trained from different datasets to boost our performance. We will present the result of the comparisons and the ensembles in Section 5.2.

One thing i dislike is laggers man
I hate inconsistency
The paper is irritating me
As of right now i hate dre
im sick of crying im tired of trying
why body pain why
uuugh i really have nothing to do right now
i dont wanna go back to mex
looking forward to holiday
well today am on lake garda enjoying the life
perfect time to read book
im feeling great enjoying my holiday

Table 1: Test samples from the Emoji Cluster model and their top-3 nearest sentences according to the learned representations. It shows that emotionally similar sentences are clustered together

2.2 Evaluation

As a result, the model can achieve 29.8% top-1 accuracy and 61.0% top-3 accuracy on the emoji cluster prediction task. Since the objective of this model is not to predict the cluster label but to find a good sentence representation, we visual-

ize the test set samples to discover that samples with similar semantics and emotions are grouped together (Table 1). Finally, similar to the first model, we use this model as a feature extractor on the competition datasets. Each text sample in the competition datasets is transformed into a 512-dimensional vector through the model except the last class predicting softmax layer.

In conclusion, we trained two deep learning models with two different emoji datasets to extract emoji representations of the competition datasets. They are transformed into high dimensional, real-valued, and continuous vectors, which can be used as features for the classification and regression tasks.

From now on, we will call the vectors from the first model, *DeepMoji representations*, and those from the second, *Emoji Cluster representations*.

3 Emotional word vectors (EVEC)

We also explore word-level representations, along with emoji sentence representations. Although sentence-level representations already build up from word representations (in particular we use pretrained Glove vectors (Pennington et al., 2014)), they may not be enough to attend to the valence that each word contains. Previous works (Tang et al., 2014) examine the significance of using sentiment-specific word embedding for related tasks. For this reason, we train emotional word vectors that not only cluster together direct emotion words such as anger and joy, but also capture emotions inside indirect emotion words, such as anger inside headache and joy inside beach. We learn these vectors by training a Convolutional Neural Network (CNN) from another separate Twitter corpus distantly labeled with hashtags.

3.1 Methodology

Our intuition to learn effective emotion word vectors is that given a document labeled with emotion there exists one or more emotionally significant words inside. Nevertheless, we do not know which ones are more important. We assume that a deep learning model, which learns the representations of the data with different level of abstractions (LeCun et al., 2015) will be able to capture those words and encode the information in its word embedding layer while classifying the documents label.

For the model structure, we use CNN since it

is proven to be effective in text classification tasks by looking at the documents n-gram features and its gradient can be directly back-propagated to the word embedding, whereas Recurrent Neural Network (RNN) models are updated sequentially. We use a similar structure used by (Kim, 2014), which includes a max-pooling layer to force the network to find the most relevant feature for predicting the emotion class correctly.

After the CNN network learns how to classify the documents into different emotion categories, we extract emotional word vectors from the network's embedding layer and use them as same as how other word embeddings, such as word2vec (Mikolov et al., 2013) or Glove, are used, treating them as features for other classification or regression models.

3.2 Hashtag Dataset

To accumulate a large corpus of emotion-labeled texts, we use a distant supervision method by using hashtags of tweets to automatically annotate emotions. Such method has proven to provide relevant emotion labels by previous works (Wang et al., 2012). Their source of the emotion words came from emotion words list made from Shaver et al. (1987), where the authors organize emotions into a hierarchy in which the first layer contains six basic emotions and each emotion has a list of emotion words. Wang et al. (2012) again expanded the list by including their lexical variants and also introduced some filtering heuristics, such as only using tweets with emotional hashtags at the end of tweets to make the distant supervision more relevant to human annotation. We combine their dataset, another public dataset [1], which used the same method, and our own extracted tweets between January and October 2017 using the Twitter Firehose API.

For the emotion labels, we focus on four emotion categories: joy, sadness, anger, and fear, since the competition tasks are only limited to those categories. In total, our hashtag dataset consists of 1.9 million tweets (Table 2).

3.3 Evaluation

For every sample in the SemEval competition dataset, we extract all emotional word vectors of the words in the sentence and simply average them

---

[1]http://hci.epfl.ch/sharing-emotion-lexicons-and-data#emo-hash-data

Emotion Label	%	Samples
Joy	36.5%	It's been such a great week **#happy**
Sadness	33.8%	I think I miss my boyfriend **#lonely**
Anger	23.5%	Ignoring me isn't going to make our problems go away. **#annoyed**
Fear	6%	What to wear for this job orientation.. **#nervous**

Table 2: Description of the Twitter hashtag corpus. Hashtags at the end were removed from the document and used as labels. It is hard to construct a well-balanced dataset for all four classes since Twitter users tend to use more hashtags related to happy and sad emotions.

into one vector. For words out of vocabulary of the hashtag corpus, we add zero vectors with the same dimension. As a result, every sentence is transformed into a 300-dimension vector to be used as features for the competition tasks. We expect these emotional word vectors can replace sentiment or emotion lexicons, since they are continuous representations learned from a large corpus, which can be more robust and rich in information about emotions inside words.

4 System Description

4.1 Features

These are the three features that are used as input for our system to solve SemEval-2018 Task 1.

Emoji Sentence Representations: Two models will be compared - *DeepMoji representations* (2304 dimensions) and *Emoji cluster representations* (512 dimensions). See Section 2.

Emotional Word Vectors (EVEC): Average of emotional word vectors learned from hashtag corpus (300 dimensions). See Section 3.

Tweet-specific features: We employ Tweet-specific features to capture information that two previous representations cannot. Inspired from the previous SemEval papers (Zhou et al., 2016; Balikas and Amini, 2016), we choose five features, (1) number of words in uppercase, (2) number of positive and negative emoticons, (3) Sum of emoji valence score [2], (4) number of elongated words, and (5) number of exclamation & question marks. Note that we do not use any linguistic features or sentiment/emotion lexicons for our system.

4.2 Preprocessing

Tweets in the competition datasets are tokenized after all non-alphanumeric characters are removed, except for extracting tweet-specific features. Some words, especially for hashtags, are merged together (e.g. #iloveyou), so unknown

words in the vocabulary is put into a wordsegment library [3] to preserve the right segment (e.g. i, love, you). Then, the tokens are transformed into emoji sentence representations (2304 or 512 dimensions) and emotional word vectors (300 dimensions), according to the vocabulary of the emoji and hashtag dataset. These datasets respectively have 262,975 and 48,929 words in their vocabularies.

4.3 Regression and Ordinal Classification

Due to the fact that the datasets of regression tasks (EI-reg & V-reg) and ordinal classification tasks (EI-oc & V-oc) have the same sample sentences, we assume that regression labels are more informative than the ordinals, since they tell us the rank among the samples within the same ordinal class. Therefore, we first train a regression model and then use it to predict ordinals, rather than training a separate classifier. We later prove that this trick yields a better result in ordinal classification.

For regression, since our features are extracted from deep learning models, we find Support Vector Regression (SVR) and Kernel-Ridge Regression methods, which are effective for nonlinear features, perform better than linear methods. We tune the hyper-parameters with the given development (dev) set and later merge both train and dev set to train the final model with the best hyperparameter found. Also, we try ensembles by averaging the final regression predictions of different methods or feature combinations to boost performance. The best groups of models are selected by the development set results of many combinations.

Another important finding is that the mapping between the regression labels and ordinal labels are very different among emotion categories. For example in Figure 2, Class 0 for fear is distributed in [0,0.6], whereas class 0 for joy is distributed in [0, 0.4]. Therefore, we try to find the mapping from the regression values (continuous) to ordinal

---

[2]https://github.com/words/emoji-emotion

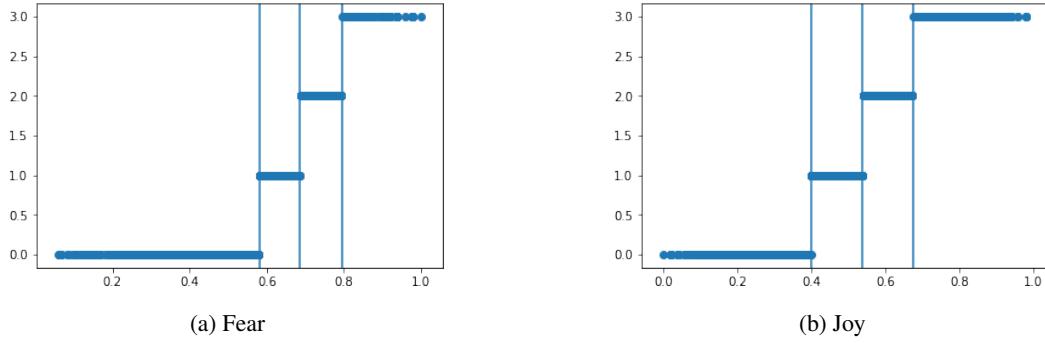[3]http://www.grantjenks.com/docs/wordsegment/

(a) Fear

(b) Joy

Figure 2: Distribution of regression labels (x-axis) and ordinal labels (y-axis) on the training dataset of Task 1a & 2a. Class 0 for fear is distributed in [0,0.6], whereas class 0 for joy is distributed in [0, 0.4]. Vertical lines are boundaries between ordinal classes, which are used for scope mapping method

values (discrete) from the training dataset. We experiment with three different mapping:

1. *naive mapping*: divides [0,1] into same size segments according to the number of ordinals

2. *scope mapping*: finds the boundary of each segment in the training dataset (vertical lines on Figure 2)

3. *polynomial mapping*: fits a polynomial regression function from the training data and finds the closest ordinal label.

4.4 Multi-label Classification

This task of multi-label classification is different from previous tasks in that the model needs to predict the binary label for each of the 11 classes given a tweet. The task is difficult in terms of three aspects. Firstly, some of the classes have opposite emotions (such as optimism and pessimism) but may have been labeled both as true. Secondly, it is not trivial to distinguish similar emotions such as joy, love, and optimism, which will include a lot of noise in the labels and make it hard to perform classification during training. Lastly, most of the tweets are labeled with no more than 3 categories out of 11 classes, which make the labels very sparse and imbalanced (Table 4).

We propose to train two models to tackle this problem: regularized linear regression and logistic regression classifier chain (Read et al., 2009). Both models aim to exploit labels' correlation to perform multi-label classification.

4.4.1 Regularized linear regression model

We formulate the multi-label classification problem as a linear regression with label distance as the regularization term. We denote the features for i-th tweet as $x_i \in R^N$ where N is the number of features and the number of categories as C. Our prediction is $y_i' = W * x_i$ where $W \in R^{M*C}$ is the weight of the linear regression model. We take the following formula as loss function to minimize. The loss consists of two parts. First part aims to minimize the mean square loss between our prediction y_i' and ground truth label y_i. The second part is the regularization term to capture relationship among different emotion labels. To model the correlations among emotions, we implicitly treat each emotion category as a vertice in an undirected graph g and use Laplacian matrix of g for regularization (Grone et al., 1990; Shahid et al., 2016) .

$$loss = \frac{1}{M} \sum_i^M (y' - y)^2 + \lambda y_i'^T L y_i'$$

$$L = D - A$$

where M is the number of samples, $L \in R^{C*C}$ is the Laplacian matrix, $A \in R^{C*C}$ is the Euclidean matrix, $D \in R^{C*C}$ is the Degree matrix. To derive L, we first compute the co-occurrence matrix $O \in R^{C*C}$ among the emotion labels and take each row/column $O_i \in R^C$ as the representation of each emotion. Then we compute the distance matrix A by taking the Euclidean distance of different labels. That is $A_{ij} = (O_i - O_j)^2$. Here, A can be regarded as the adjacency matrix of the graph g. Afterwards, we calculate the degree matrix D by summing up each row/column and making it a diagonal matrix.

Subtasks	1a&2a	3a&4a	5a
Train	7,102	1,181	6,838
Dev	1,464	449	886
Test	4,068	937	3,259

Table 3: Statistics of the competition dataset for all 5 subtasks

# of labels	0	1	2	3	4	5	6
%	2.9	14.3	40.6	30.9	9.6	1.4	0.2

Table 4: Number of multi-labels. Most samples have from 1-3 labels, but can have no labels or up to 6 labels. (subtask 5a)

4.4.2 Logistic regression classifier chain

Classifier chain is another method to capture the correlation of emotion labels. It treats the multi-label problem as a sequence of binary classification problem while taking the prediction of the previous classifier as extra input. For example, when training the i-th emotion category, we take both the features of input tweet and also the 1st, 2nd, $\cdots$, (i-1)-th prediction as the input of our logistic regression classifier to predict the i-th emotion label of input tweet. We further ensemble 10 logistic regression chains by shuffling the sequence of 11 emotion labels to achieve better generalization ability.

5 Experiments & Results

Most of our system experiments were implemented by using PyTorch (Paszke et al., 2017) and Scikit-learn (Pedregosa et al., 2011).

5.1 Competition dataset

SemEval-2018 Affect in Tweets (AIT) is created by human annotators through crowd-sourcing methods (Mohammad and Kiritchenko, 2018). Total three datasets are given: emotion intensity (with four emotion categories; Subtask 1a & 2a), sentiment intensity (subtask 3a & 4a), and multi-label emotion classification (subtask 5a).

For emotion and sentiment intensity datasets, each tweet sample has both an *ordinal label* (coarse; {0,1,2,3} for emotion, {-3,-2,-1,0,1,2,3} for sentiment) and real-value *regression label* (fine-grained; [0,1]). For multi-label emotion classification dataset, each can have none or up to six number of multi-labels (Table 4).

We used the given development set to tune the hyper-parameters and select models. For the fi-

nal submission, we merged the train & development set together to retrain the model with the best hyper-parameter found (Table 3).

5.2 Regression: Subtask 1a & 3a

Table 5 shows the test set results on regression tasks, Subtask 1a&3a. We experimented with different features that we introduced before to analyze the effectiveness of each representation. For emoji sentence representations, *emoji cluster* worked better on sadness and sentiment, whereas *DeepMoji* outperformed in anger, fear, and joy. We presumed such difference was due to the different emoji types of the two datasets used to train each model. *Emoji cluster* only used 11 classes of emojis that were clustered together, but *DeepMoji* used 64 emoji classes. It may be possible clustering of emoji classes made it easy for regression models to predict the intensities in certain emotion categories, whereas some emotion categories needed more detailed representations.

The emotional word vectors overall did help enhance the performance of the regression model for all emotion categories. This shows that emotional word vectors can serve as additional word-level information which are helpful for solving this task.

Tweet-specific features boosted the performance, notably for sentiment, since features like capital letters, emojis, elongated words, and the number of exclamation marks, could help to figure out the subtle difference of the emotion intensities.

One thing to note is that our system's rank in the fear category (7th) is relatively lower than other emotion categories. We found out from the previous literature (Wood and Ruder, 2016) that fear emojis were the most ambiguous, having the least correlation with human-annotated emotion labels among the six emotion categories. On the other hand, joy emojis were the most highly correlated. This may explain our best performance in the joy category and worst performance in the fear category. Future systems using emojis as a dataset may need to take this shortcoming into account.

5.3 Ordinal Classification: Subtask 2a& 4a

As mentioned in Sec 4.3, we used our best regression model to also predict ordinal labels. Since each emotion category has a different distribution of regression labels and ordinal labels, we experimented three different mappings, *naive mapping*, *scope mapping*, and *polynomial mapping*. Using

Features	Regression Method	Pearson correlation (all instances)				
		1a (EI-reg)				3a(V-reg)
		Anger	Fear	Joy	Sadness	Valence
Emoji Cluster	SVR	.733	.632	.679	.693	.811
Emoji Cluster	KernelRidge	.735	.638	.675	.692	.809
DeepMoji	SVR	.772	.675	.736	.664	.798
DeepMoji	KernelRidge	.778	.672	.737	.698	.798
Emoji Cluster + EVEC	SVR	.739	.678	.701	.706	.815
Emoji Cluster + EVEC	KernelRidge	.741	.694	.709	.709	.822
DeepMoji + EVEC	SVR	.781	.694	.749	.708	.810
DeepMoji + EVEC	KernelRidge	.779	.702	.754	.710	.813
DeepMoji + feat.	SVR	.785	.680	.739	.714	.824
DeepMoji + feat.	KernelRidge	.781	.670	.691	.711	.829
Emoji Cluster + EVEC + feat.	SVR	.757	.684	.720	.725	**.844**
Emoji Cluster + EVEC + feat.	KernelRidge	.757	.698	.693	.721	.840
DeepMoji + EVEC + feat.	SVR	**.792**	.709	**.763**	.732	.837
DeepMoji + EVEC + feat.	KernelRidge	.790	**.716**	.734	**.739**	.826
Best Ensemble		**.811(2)**	**.728(7)**	**.773(2)**	**.753(5)**	**.860(3)**

Table 5: Test set results on Subtask 1a & 3a. For 1a, separate regression models were trained for each emotion category. The number next to the best result(bold & underlined) indicates our ranking of the competition. Underlined ones show the models that were selected for ensemble according to the dev set.

Task		Pearson (all instances)		
		Naive	Scope	Poly
	Anger	.654	.664	**.704(2)**
	Fear	.498	.562	**.570(*)**
2a (EI-oc)	Joy	.632	**.720(1)**	.712
	Sadness	.645	**.697(*)**	.692
4a (V-oc)	Valence	.813	.816	**.833(2)**

Table 6: Test set results on Subtask 2a & 4a. The predictions of the best regression models are mapped into ordinal predictions. The number next to the best result(bold & underlined) indicates our ranking of the competition. (*) indicates better results that we acquired after our final submission

the training set, we found the ideal mapping function to match the regression predictions and the ordinal predictions.

Test set results (Table 6) on ordinal classification show that our mapping methods are indeed much more effective. For anger, fear, and sentiment categories, polynomial mapping performed the best, whereas scope mapping outperformed for joy and sadness categories. With our method, we achieved higher ranks in ordinal classification tasks (2a & 4a), placed both in 2nd. Figure 3 shows how a cubic function is fitted to find the mapping between regression labels and ordinal labels.

Additionally, we report some results better than the final submission. The change is due to a new model selection strategy. For the final submission, we searched for the optimal pair of regression model & mapping method by looking at the
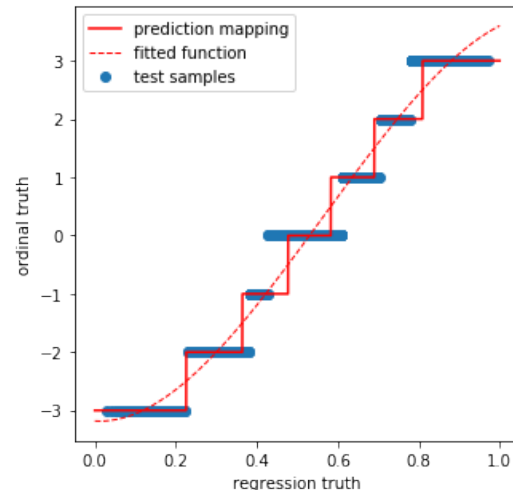

Figure 3: Plot of test labels and the mapping function derived from the training set. A polynomial function is fitted to map the regression predictions into ordinal predictions

ordinal classification results on the development set. However, it turned out that always using the best ensemble prediction and then searching for the optimal mapping method with respect to the development set was better.

5.4 Multi-label Classification: Subtask 5a

We found the best hyper-parameters by evaluating on our development set. We initialized the weight matrix W with a normal distribution of standard deviation of 0.1. We used gradient descent to optimize this function and set the learning rate to 1.0.

270

Features	CC	RLR
Emoji Cluster	.528	.545
DeepMoji	.532	.552
Emoji Cluster + EVEC	.545	.558
DeepMoji + EVEC	.544	.558
Emoji Cluster + EVEC + f	.546	.558
DeepMoji + EVEC + f	.550	.562
Best ensemble		.576(3)

Table 7: Test set results on Subtask 5a. The competition metric is Jaccard index.

	Gender		Race	
	Ours	Avg	**Ours**	Avg
Anger	**0.5%**	0.1%	**1%**	0.4%
Fear	**-0.9%**	-0.3%	**3.3%**	0.5%
Joy	**-1.2%**	0.4%	**-0.9%**	-0.7%
Sadness	**0%**	0.2%	**1.3%**	0.8%
Valence	**-0.6%**	0.5%	**-1%**	-0.6%

Table 8: Average differences of the system's bias. Gender difference is from female to male, and race differences is from African American names to European American names (sign of the percentage indicates the direction). "Ours" indicate the bias of our system, and "Avg" is the average of the biases of all systems from the competition.

The optimal λ we found was -0.0001.

We found that regularized linear regression model was always better than classifier chain model. The ensemble of classifier chain and regularized linear regression of both features combination(underlined elements in Table 7) achieved best performance than any single model (Table 7).

5.5 Analysis of system's gender/racial biases

In this year's competition, the organizers gave out a mystery test set that was included in the regression tasks (subtask 1a & 3a). At the end of the evaluation period, they announced that these were set of pair sentences that differ only in the subject's or object's gender or racial names (See the task paper Mohammad et al. (2018) for details). It turned out that our system also included some biases like most other systems did, but fairly small, less than 1.5% for gender bias and 3.5% for racial bias (Table 8). We believe that this is an interesting experiment and look forward to discussing more about the issue during the workshop.

6 Conclusion

In this paper, we explored a couple of different methods to find good representations of emo-

Subtask	System	Score(rank)
1a EI-reg	SeerNet	.799(1)
	NTUA-SLP	.776(2)
	PlusEmo2Vec	.766(3)
	psyML	.765(4)
2a EI-oc	SeerNet	.695(1)
	PlusEmo2Vec	.659(2)
	psyML	.653(3)
3a V-reg	SeerNet	.873(1)
	TCS Research	.861(2)
	PlusEmo2Vec	.860(3)
	NTUA-SLP	.851(4)
4a V-oc	SeerNet	.836(1)
	PlusEmo2Vec	.833(2)
	Amobee	.813(3)
5a E-c	NTUA-SLP	.588(1)
	TCS Research	.582(2)
	PlusEmo2Vec	.576(3)
	psyML	.574(4)

Table 9: Official final scoreboard on all 5 subtasks that we participated. Scores for Subtask 1-4 are macro-average of the Pearson scores of 4 emotion categories and 5 is Jaccard index. About 35 participants are in each task.

tions inside tweets for solving 5 subtasks of predicting emotion/sentiment intensity and emotion labels. We used external datasets, which were much larger than the competition dataset but distantly labeled with emojis and #hashtags, to exploit the transferred knowledge to build a more robust machine learning system to solve the task. We avoided using traditional NLP features like linguistic features and emotion/sentiment lexicons by substituting them with continuous vector representations learned from huge corpora.

We performed experiments to show that emoji sentence representations and emotional word vectors trained from neural networks can be used with tweet-specific features as input for other traditional regression models, such as SVR and Kernel Regression, to solve the task of regression and ordinal classification. We proved the effectiveness of finding the mapping of the relationship between regression and ordinal labels from the training set to perform ordinal classification. Moreover, we tried using classifier chain and regularized logistic regression to deal with multi-label classification.

As a final official result (Table 9), our system ranked among the top three in every subtask of the competition we participated. For future work, we want to work further on employing these emotion representations on other tasks, such as text generation, while we gather more data and improve the model to train the representations.

References

Georgios Balikas and Massih-Reza Amini. 2016. Twise at semeval-2016 task 4: Twitter sentiment classification. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016).*

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. *EMNLP 2017.*

Pascale Fung. 2015. Robots with heart. *Scientific American*, 313(5):60–63.

Robert Grone, Russell Merris, and V S‿ Sunder. 1990. The laplacian spectrum of a graph. *SIAM Journal on Matrix Analysis and Applications*, 11(2):218–238.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP, publisher=Citeseer,.*

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*, pages 3294–3302.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Saif M. Mohammad and Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference*, Miyazaki, Japan.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP2014*, pages 1532–1543.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2009. Classifier chains for multilabel classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer.

Nauman Shahid, Nathanael Perraudin, Vassilis Kalofolias, Benjamin Ricaud, and Pierre Vandergheynst. 2016. Pca using graph total variation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4668–4672. Ieee.

Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'connor. 1987. Emotion knowledge: Further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061.

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

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565.

Mike Thelwall. 2017. The heart and soul of the web? sentiment strength detection in the social web with sentistrength. In *Cyberemotions*, pages 119–134. Springer.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P Sheth. 2012. Harnessing twitter" big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 587–592. IEEE.

Ian Wood and Sebastian Ruder. 2016. Emoji as emotion tags for tweets. In *Emotion and Sentiment Analysis Workshop, at LREC2016*. LREC2016.

Yunxiao Zhou, Zhihua Zhang, and Man Lan. 2016. Ecnu at semeval-2016 task 4: An empirical investigation of traditional nlp features and word embedding features for sentence-level and topic-level sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 256–261.