Using millions of emoji occurrences to pretrain any-domain models for detecting emotion, sentiment and sarcasm

Anonymous ACL submission

Abstract

NLP tasks such as emotion detection, sarcasm detection, and sentiment analysis aim to infer information about an author's emotional state and intentions, often conveyed indirectly. There are many ways to signal emotion or sentiment, but learning the complex relation between implicit cues and an author's intentions is difficult due to the scarcity of manually annotated data. This paper shows that by harvesting millions of emoji occurrences and learning to predict them in context, we can learn text representations that are less dependent on large volumes of annotated data. Using emoji prediction to pretrain our model we obtain improvements over the state of the art on 5 out of 6 benchmark datasets.

1 Introduction

Emojis are simple, pictorial glyphs, first introduced in 1997, depicting faces, natural objects, artifacts, etc. They are widely used in social media such as Twitter, where they serve multiple purposes: Glyphs of natural objects and artifacts can serve as substitutions for words to shorten texts, but frequently emojis such as smileys or hearts are used to explicitly signal the author's emotional state or intent, e.g., humor, sarcasm, or objection.

Emojis can therefore serve as a proxy for a classification of the emotional contents of texts. In this paper, we will use emojis as the basis for a form of distant supervision for pretraining models to detect emotion, sarcasm, and sentiment.

Emojis are not always a direct labeling of emotional content. For example, a smiley may serve to disambiguate an ambiguous sentence, or to complement an otherwise relatively negative text, e.g.: *:) but I still don't agree.*. Kunneman et al. (2014) also discusses a similar duality in the use of emotional hashtags such as *#nice* and *#lame*. Nevertheless, our work shows that emojis are able to classify the emotional contents of texts accurately in many cases; see Table 1 for concrete examples. We invite the reviewers to test our DeepMoji model with their own sentences at deepmoji.pythonanywhere.com using the login *deepmoji* and password *acl_2017*.

Table 1: Examples of scorings provided by our DeepMoji model. For each text the top 5 most likely emojis are presented along with the model's probability estimates.

My dad just died. my life is great.	😔 8.0%	😢 6.9%	💔 6.7%	🔫 6.1%	😞 5.7%
Do you know what's even better? I just lost my job	😔 8.9%	😞 7.0%	😓 6.2%	😢 5.4%	😪 5.4%
This is shit.	😡 6.6%	😤 6.5%	😠 6.3%	😒 6.0%	😓 5.0%
This is the shit.	🎧 8.1%	👌 7.3%	💯 4.9%	😈 4.6%	🎶 4.4%
I love how you did that	👏 6.5%	👌 6.0%	😌 5.5%	😍 4.9%	😊 4.2%
I love cruising down the street in my new BMW	😍 29.3%	😋 6.3%	😘 4.5%	😎 4.4%	😈 4.2%
I love your presence and now you won't be here	💔 23.3%	😔 13.9%	😢 9.1%	😞 8.9%	😪 7.3%

Here we present an approach to detecting emotion, sarcasm, and sentiment detection in text that does not rely on any dictionaries or any linguistic resources other than the labeled data available for these tasks. Instead we exploit the transfer potential of a model trained to predict 64 types of emojis based on millions of occurrences.

The architecture we propose, explained in §3, is a dense, deep LSTM that is pretrained to predict emojis and fine-tuned in a layer-wise fashion on the target task. The pretraining leads to improvements across multiple domains, as shown in §4. As our analysis in §5 indicates, the pretraining helps our target task models attend to evidence with little support in the manually annotated data for the target tasks. In addition, the emoji pretrainig acts like a regularizer and as a result, our architecture is not prone to overfitting.

Contributions We show how to use millions of readily available emoji occurrences on Twitter to pretrain any-domain models for emotion, sarcasm, and sentiment detection. We transfer the pretrained models to the target tasks using layer-wise fine-tuning on small amounts of labeled data. We present experiments on 6 datasets across three tasks with improvements over the state of the art on all but one dataset. Finally, we present an analysis of the effect of pretraining, showing that a) pretraining helps the target task models attend to low-support evidence, b) this evidence is not only out-of-vocabulary or rare words, but also phrases, and c) pretraining also helps to regularize the target task models.

2 Related work

The use of emotional expressions in text as noisy labels in order to counter scarcity of labels in sentiment modeling is not a new idea. This idea was originally introduced in (Read, 2005) using emoticons and has since been expanded to also include hashtags and emojis (Suttles and Ide, 2013; Mohammad, 2012). These techniques have also been applied to social media data, e.g. Twitter (Go et al., 2009).

Initial sentiment research focused on classifying content into positive/neutral/negative classes (Pang et al., 2002). The scope has since then been expanded to also classify emotional content using theories of emotion such as Ekman's six basic emotions (Mohammad, 2012) or Plutchik's eight basic emotions (Suttles and Ide, 2013). Emotional expressions are manually chosen to belong to one of the sentiment types or emotional categories. For instance, hashtags such as #anger, #joy, #happytweet, #ugh, #yuck and #fml have in previous research been mapped into emotional categories (Suttles and Ide, 2013; Mohammad, 2012). Other work has also separated emojis into either a 'happy' or 'sad' category and used these for dictionary building or as part of the training procedure (Vo and Zhang, 2016).

Such manual categorization requires an understanding of the emotional content of each expression, which is difficult and time-consuming for sophisticated combinations of emotional content. Moreover, any manual selection and categorization is prone to misinterpretations and may omit important details regarding usage.

In contrast, our approach requires no prior knowledge of the corpus and can capture diverse usage of 64 types of emojis (see Table 1 for examples and Figure 3 for an visualization of how the model implicitly groups emojis).

Another way of automatically interpreting the emotional content of a given emoji was recently introduced by (Eisner et al., 2016), who learn emoji embeddings from the words describing the emoji-semantics in official emoji tables. This approach, in our context, suffers from two severe limitations: a) It requires emojis at test time while there are many domains with limited or no usage of emojis. b) The tables do not capture the dynamics of emoji usage, i.e., drift in an emoji's intended meaning over time.

Knowledge can be transferred from the emoji dataset to the target task in many different ways. In particular, multitask learning has shown promising results when using multiple datasets to improve performance with respect to a target task (Collobert and Weston, 2008). However, multitask learning requires access to the emoji dataset whenever the classifier needs to be tuned for a new target task. Requiring access to the dataset is problematic in terms of violating any data access regulation. There are also problems from a data storage perspective as the dataset used for this research contains hundreds of millions of tweets (see Table 2). Instead we utilize transfer learning (Bengio et al., 2012) as described in §3.3, which does not require access to the original dataset, but only the pretrained classifier.

3 Method

3.1 Pretraining

In many cases, emojis serve as a proxy for the emotional contents of a text. Therefore, pretraining on the classification task of predicting which emoji were initially part of a text can help performance on the target task. See §5.1 for a detailed analysis of why pretraining helps the models. Social media contains large amounts of short texts with emojis that can be utilized as noisy labels for pretraining. Here, we use data from Twitter, but any dataset with emoji occurrences could be used as well. Only English tweets without mentions or URL's are used to minimize the amount of outside contextual information affecting the emotional meaning of the text.

Many tweets contain multiples of the same emoji or multiple different emojis. In the training data, we address this in the following way. For each unique emoji type, we save a separate tweet for the pretraining with that emoji type as the label. We only save a single tweet for the pretraining per unique emoji type regardless of the number of emojis associated with the tweet. This data preprocessing allows the pretraining task to capture that multiple types of emotional content are associated with the tweet while making our pretraining task a single-label classification instead of a more complicated multi-label classification.

To ensure that the pretraining encourages the models to learn a rich set of emotional content rather than only the most used emojis, we ensure that the pretraining dataset is balanced. The pretraining data is split into a training, validation and test set, where the validation and test set is randomly sampled in such a way that each emoji is equally represented. The remaining data is upsampled to create a balanced training dataset. All tweets are tokenized on a word-by-word basis. Words with 2 or more repeated characters are shortened to the same token (e.g. ”loool” and ”loooooool” are tokenized such that they are treated the same).

3.2 Model

With the millions of emoji occurrences available we can train very expressive classifiers with limited risk of overfitting. We utilize the Long Short-Term Memory (LSTM) model that has been successful at many NLP tasks (Hochreiter and Schmidhuber, 1997; Sutskever et al., 2014). We us a variant, DenseLSTM, which uses connections from all previous hidden layers to layers later in the network inspired by (Huang et al., 2016).

The high connectivity of DenseLSTM increases performance, likely due to better gradient-flow from the output layer to the early layers in the network thus making training easier (Huang et al., 2016). Another benefit of the DenseLSTM model is that features can be combined across previous layers. We use 4 bidirectional LSTM layers with 512 hidden units in each (256 in each direction). An illustration of our model is shown in Figure 1. The only regularization used for the pretraining task is a L2 regularization of 1E−4 on the LSTM weights. We do not apply dropout during training due to the size of the pretraining data. After convergence the trained network is retrained lightly

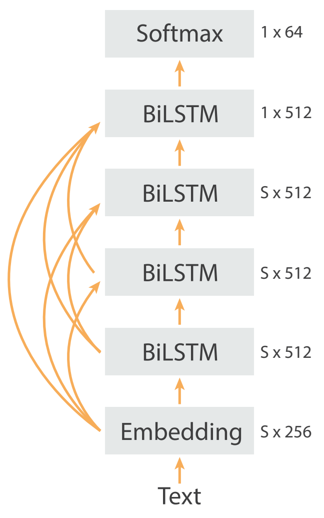

Figure 1: Illustration of the DenseLSTM model used in this work. The model contains 4 bidirectional LSTM layers with 512 hidden units in each (256 in each direction).

on the pretraining dataset with dropout probabilities described in §4.2. Because the same dropout probabilities are used across all target tasks, this slight retraining only has to be done once regardless of the number of target tasks.

3.3 Transfer learning

Our pretrained model can be fine-tuned to the target task in multiple ways. Particularly, two common ways of transferring knowledge is to either use the pretrained model as an initialization (Erhan et al., 2010) or as a feature extractor (Donahue et al., 2014). In the latter case, all layers in the model except the Softmax layer are frozen when fine-tuning for the target task. These two approaches are in the remainder of the paper referred to as respectively ‘full’ and ‘last’ in relation to the trainable layers with each approach.

We propose a third transfer learning approach, ‘seqthraw’, that sequentially unfreezes and fine-tunes a single layer at a time. This approach increases accuracy on the target task at the expense of extra computational power needed for the fine-tuning. By training each layer separately the model is able to adjust the individual patterns across the network with less risk of overfitting than when fine-tuning all layers at a time. The sequential fine-tuning seems to have a regularizing effect similar to what has been examined in previous work in the context of unsupervised learning (Erhan et al., 2010).

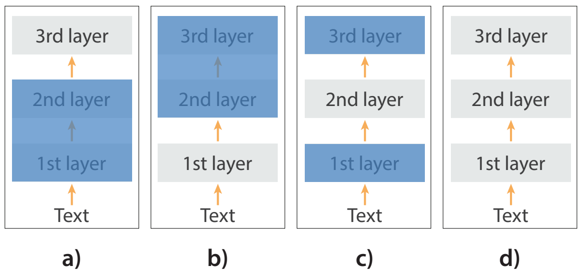

Figure 2: Illustration of the seqthraw transfer learning approach, where each layer is fine-tuned separately. Layers covered with a blue rectangle are frozen. Step a) tunes any new layers, b) then tunes the 1st layer and c) the next layer until all layers have been fine-tuned individually. Lastly, in step d) all layers are fine-tuned together.

More specifically, the seqthraw approach first fine-tunes any new layers (often only a Softmax layer) to the target task until convergence on a validation set. Then the approach fine-tunes each layer individually starting from the first layer in the network (often the embedding layer). Lastly, the entire model is trained with all layers. Each time the model converges as measured on the validation set, the weights are reloaded to the best setting, thereby preventing overfitting in a similar manner to early stopping (Sjöberg and Ljung, 1995). This process is illustrated in Figure 2. Note how only performing step a) in the figure is identical to the feature extraction approach mentioned earlier and only doing step d) is identical to the initialization approach. Although the seqthraw process may seem extensive it is easily implemented with only a few lines of code. Similarly, the additional time spent on fine-tuning is limited when using GPUs on small benchmark datasets as is often the case.

An important benefit of the seqthraw approach is that it makes it easy to expand the vocabulary to other domains with little risk of overfitting. For a given dataset we add up to 10000 new tokens from the training set to the vocabulary. §5.1 contains some analysis on the added word coverage gained from this approach.

None of these transfer learning approaches require the storage of the original dataset, thereby allowing the model to be deployed and fine-tuned separately for a specific task without the burden of storing millions of tweets (see §2).

One difficulty with transferring knowledge from an LSTM model trained on tweets to other domains is that patterns learned on short texts of a few sentences may not be directly generalizable to much longer texts. More importantly, the recurrent structure of the LSTM model makes the neuron activations and associated gradients in the model 'explode' (i.e. attain such high values that they become numerically unstable) when used on texts much longer than the ones seen in the training set (Krueger and Memisevic, 2016). Texts longer than 30 tokens are therefore split into a sequence of smaller text regions of length 20 that each are encoded with the LSTM model similar to (Johnson and Zhang, 2015). We then use average-pooling on the features across all text regions to find the overall features for the text. This region-embedding approach makes it easier for the pretrained model to generalize to texts of any length.

4 Experiments

4.1 Emoji prediction

After preprocessing the tweets as described in §3.1 the pretraining dataset consists of 634M tweets (see Table 2). To evaluate performance on the pretraining task a validation set of 0.64M tweets (10K of each emoji type) and a test set of 6.4M tweets (100K of each emoji type) are used. The remaining tweets are used for the training set, which is balanced using upsampling.

Table 2: The number of tweets in the pretraining dataset associated with each emoji in millions. Note that a single tweet with multiple unique emoji types will be treated as multiple tweets in the pretraining dataset as described in §3.1.

😂	😭	😩	😍	😒	❤️	👌	😊	😔	💕	😁	😏	😘	🙌	☺️	💯
77.8	32.7	31.2	29.5	26.8	22.5	21.5	18.4	15.6	15.5	13.5	13.0	12.8	12.1	12.0	12.0
🎶	🙏	😳	😴	😌	🙈	♥️	✌️	👏	😑	😅	💁	👍	😎	😫	😡
10.7	10.7	10.7	9.1	8.7	8.2	8.1	8.0	7.7	7.3	7.3	6.9	6.7	6.7	6.4	6.1
😕	😐	✨	😋	💙	😢	😞	💜	💖	👀	🔫	😤	🙊	✋	💀	😪
5.9	5.5	5.5	5.4	5.4	5.4	5.4	5.2	5.1	5.1	4.9	4.8	4.8	4.8	4.5	4.5
😈	😷	💔	♡	😉	😄	💪	😜	👊	😣	🎧	😓	🙅	😖	😠	😬
4.4	4.0	4.0	4.0	3.8	3.8	3.8	3.8	3.7	3.3	3.2	3.1	3.0	3.0	2.8	2.4

Table 3: Accuracy of classifiers on the emoji prediction task. Parameters in millions.

	Params	Top 1	Top 5
Random	–	1.6%	7.8%
fasttext	16.8	14.2%	39.0%
LSTM	22.6	18.0%	45.5%
DenseLSTM	27.3	**18.6%**	**46.3%**

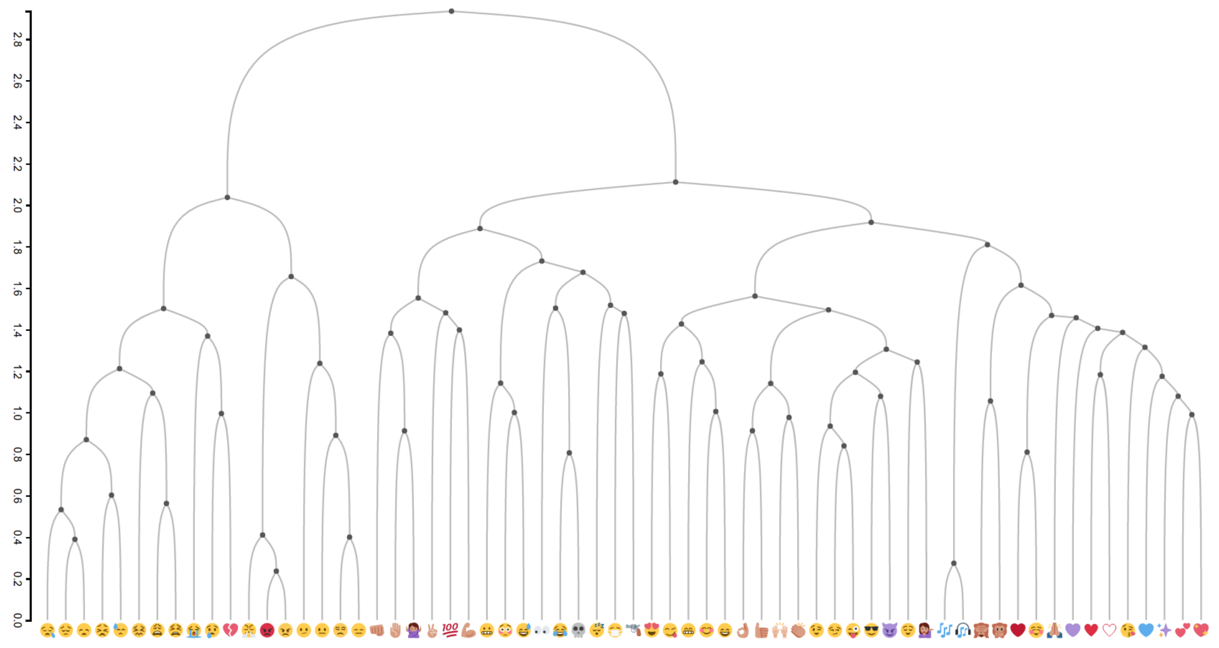

Figure 3: Hierarchical clustering of the DeepMoji model's predictions across categories on the test set. The dendrogram visualized here shows how the model learns to group emojis into overall categories and subcategories based on emotional content. The y-axis is the distance on the correlation matrix of the model's predictions measured using average linkage.

The performance of different classifiers is evaluated on the pretraining task with the results shown in Table 3. Both top 1 and top 5 accuracy is used for the evaluation as the emoji labels are noisy with multiple emojis being potentially correct for any given sentence. Informal experiments do find, however, that increased performance on the pretraining task leads to increased performance on the target task.

For comparison we also train a bag-of-words classifier, fasttext (Joulin et al., 2016), that has recently shown competitive results. We use 256 dimensions for this classifier utilizing the same vocabulary as the LSTM models, thereby making it very similar to the embedding layer used by the LSTM models. The difference in top 5 accuracy between the fasttext classifier (39.0%) and the best LSTM model (46.3%) underlines the difficulty of the emoji prediction task. As the two classifiers use the same embedding dimensions and vocabulary, this difference in accuracy demonstrates the importance of capturing the context in which each word is used.

Many of the emojis used carry similar emotional content. Through hierarchical clustering on the correlation matrix of the model's predictions on the test set we can see that the model captures many similarities that one would intuitively expect (see Figure 3). For instance, the model groups emojis into overall categories associated with e.g. negativity, positivity or love. Similarly, the model learns to differentiate within these categories, mapping sad emojis in one subcategory of negativity, annoyed in another subcategory and angry in a third one.

4.2 Benchmarking

We benchmark our method on 3 different NLP tasks using 6 datasets across 5 domains. To make for a fair comparison, we compare Deepmoji to other methods that also utilize other data sources in addition to the benchmark dataset. To easily compare the performance across datasets we evaluate the datasets using classification, discretizing any datasets with continuous labels. Through discretization we also reduce the influence of noise in crowd-annotated data, since the classification task becomes crude rather than fine-grained. An averaged F1-measure across classes is used for evaluation in emotion analysis and sarcasm detection as these consist of highly unbalanced datasets while sentiment datasets are evaluated using accuracy.

Some of the datasets originate from domains that differ substantially from the tweets on which

Table 4: Description of benchmark datasets.

Identifier	Study	Task	Domain	Classes	N_{train}	N_{test}
SE0714	(Strapparava and Mihalcea, 2007)	Emotions	Headlines	3	250	1000
Olympic	(Sintsova et al., 2013)	Valence + Arousal	Tweets	4	250	709
PsychExp	(Wallbott and Scherer, 1986)	Emotions	Experiences	7	6000	1480
SS-Twitter	(Thelwall et al., 2012)	Sentiment	Tweets	2	1700	402
SS-Youtube	(Thelwall et al., 2012)	Sentiment	Video Comments	2	1700	442
SCv2-GEN	(Oraby et al., 2016)	Sarcasm	Debate Forums	2	2600	660

our method is pretrained. A key difference between the datasets is their length, with the headlines from the SemEval 2007 Task 14 (Strapparava and Mihalcea, 2007) containing emotional content in just a few words and the board posts from the Internet Argument Corpus (Oraby et al., 2016) containing long texts of information needed for detecting sarcasm.

An issue with many of the benchmarks is the data scarcity, which is particularly problematic within emotion analysis. Many recent papers proposing new methods for emotion analysis such as (Staiano and Guerini, 2014) only evaluate performance on a single benchmark dataset, SemEval 2007 Task 14, that contains 1250 observations. Recently, criticism has been raised concerning the use of correlation with continuous ratings as a measure (Buechel and Hahn, 2016), making only the somewhat limited binary evaluation possible. We only evaluate the emotions {Fear, Joy, Sadness} as the remaining emotions occur in less than 5% of the observations.

To fully evaluate our method on emotion analysis against the current methods we thus make use of two other datasets. Sintsova et al. has created a dataset of emotions in tweets related to the Olympic Games. Each tweet can be assigned multiple emotions out of 20 possible emotions, making evaluation difficult. To counter this difficulty, we have chosen to convert the labels to 4 classes of low/high valence and low/high arousal based on the Geneva Emotion Wheel that the study used. A tweet is deemed as having emotions within the valence/arousal class if the average evaluation by raters for that class is 2.0 or higher, where 'Low' = 1, 'Medium' = 2 and 'High' = 3.

Another useful dataset is the ISEAR databank (Wallbott and Scherer, 1986), which was created over many years by a large group of psychologists that interviewed respondents in 37 countries. Each observation in the dataset is a self-reported experience mapped to 1 of 7 possible emotions, making for an interesting benchmark dataset of a decent size (see Table 4). As these two datasets do not have prior evaluations, we evaluate against the current state-of-the-art approach, which is based on a valence-arousal-dominance framework (Buechel and Hahn, 2016). The scores extracted using this lexicon-based approach are mapped to the classes in the datasets using a logistic regression with parameter optimization using cross-validation. We will release our preprocessing code and hope that these 2 two datasets will be used for future benchmarking within emotion analysis.

We evaluate sentiment analysis performance on 2 datasets from SentiStrength (Thelwall et al., 2010), SS-Twitter and SS-Youtube, and follow the relabeling described in (Saif et al., 2013) to make the labels binary. For proper evaluation on sentiment prediction using outside datasets we evaluate our method versus the newest version (updated November 2016) of the SentiStrength method. As with the emotion analysis, a logistic regression is trained using cross-validation to map the extracted scores to the appropriate class.

For sarcasm detection we use the sarcasm dataset v2 from the Internet Argument Corpus (Oraby et al., 2016). This dataset contains both a quoted text and a sarcastic response, but for simplicity we only model the response and do not consider the context.

The datasets that do not have a pre-existing training/test division are split as shown in Table 4. Our training/test splits will be made public.

For training we utilize the Adam optimizer (Kingma and Ba, 2014) with gradient clipping of the norm to 1. To prevent overfitting on the small datasets, 10% dropout is applied to the embedding layer, whereas the LSTM layers have 25% dropout applied to the input connections and recurrent connections. Details are described in

Table 5: Results across benchmark datasets. The variations of the DeepMoji model refers to the transfer learning approaches mentioned in §3.3 with 'new' being a model trained without transfer learning.

Dataset	Measure	State of the art	DeepMoji (new)	DeepMoji (full)	DeepMoji (last)	DeepMoji (seqthraw)
SE0714	F1	**0.43** [Buechel]	0.24	0.30	0.35	0.35
Olympic	F1	0.49 [Buechel]	0.45	0.59	0.61	**0.62**
PsychExp	F1	0.45 [Buechel]	0.51	0.63	0.56	**0.64**
SS-Twitter	Acc	73.4% [Saif]	70.0%	87.6%	81.6%	**90.0%**
SS-Youtube	Acc	85.0% [Thelwall]	80.3%	90.6%	89.8%	**91.0%**
SCv2-GEN	F1	0.73 [Oraby]	0.70	0.75	**0.76**	**0.76**

(Gal and Ghahramani, 2016).

Table 5 shows that the model out performs the state of the art on 6 out of 5 benchmarks. The new transfer learning approach, seqthraw, yields the best performance of the different transfer learning techniques. The highest gain in performance is on the SS-Twitter sentiment dataset, which is intuitive as it shares domain with the pretraining data.

5 Model Analysis

5.1 Analyzing the effect of pretraining

Performance on the target task benefits greatly from the pretraining as seen in Table 5 by comparing DeepMoji (new) to DeepMoji (seqthraw). In this section we experimentally decompose the benefit of pretraining into 3 effects: word coverage, phrase coverage and regularization.

There are numerous ways to express a specific emotion, sentiment or sarcastic comment. Consequently, the test set may contain specific language usage not present in the training set. The pretraining helps the target task models attend to low-support evidence by having previously observed similar usage in the pretraining dataset.

We first examine this effect by measuring the improvement in word coverage on the test set when using the pretraining. Table 6 shows how the word coverage increase is more substantial on small datasets such as the ones used for emotion analysis. Note that word coverage can be a slightly misleading metric in this context as for many of these small datasets a word will often occur only once in the training set. In contrast, all of the words in the pretraining vocabulary are present in thousands (if not millions) of observations in the emoji pretraining dataset thus making it easier for the model to learn a good representation of the emotional and semantic meaning. The added benefit of pretraining therefore likely extends beyond the differences seen in Table 6.

Table 6: Word coverage on benchmark test sets using only the vocabulary generated by finding words in the training data, the pretraining vocabulary (last approach) or a combination of both vocabularies (full or seqthraw approach).

Dataset	Own	DeepMoji (last)	DeepMoji (full/seqthraw)
SE0714	44.6%	92.3%	92.6%
Olympic	75.4%	92.0%	95.6%
PsychExp	96.6%	98.0%	98.7%
SS-Twitter	85.9%	89.9%	95.8%
SS-Youtube	85.8%	95.5%	65.1%
SCv2-GEN	94.0%	96.4%	97.7%

To examine the importance of capturing phrases and the context of each word, we evaluate the accuracy on the SS-Twitter dataset using only the embedding layer in our DeepMoji model vs. using all layers. Table 7 shows how the LSTM layers dramatically improves the accuracy from 78.4% to 90.0%, thereby emphasizing the importance of phrase coverage. One concept that the LSTM layers likely help the model learn is negation, which is known to be important for sentiment analysis (Wiegand et al., 2010).

Table 7: Accuracy on the SS-Twitter benchmark dataset using embedding dimension only vs. all layers model.

	Accuracy
Random	57.6%
State of the art	73.4%
DeepMoji (embed only)	78.4%
DeepMoji (all layers)	90.0%

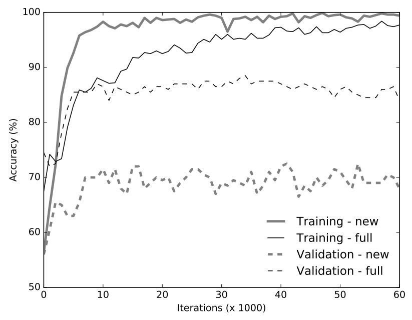

Figure 4: Plot of the training and validation error when training on the SS-Twitter dataset. Both models have all layers unfrozen, the same learning rate and same amount of regularization.

Lastly, Figure 4 shows an example of how the pretraining helps to regularize the target task model, which otherwise quickly overfits. The seqthraw transfer learning approach further increases this regularization by fine-tuning the model layer wise, thereby adding additional regularization as described in Section 3.3.

5.2 Comparing with human-level agreement

To understand how well our DeepMoji classifier performs compared to humans we collect our own dataset of random tweets annotated for positive/negative sentiment. Each tweet is annotated by a minimum of 10 English-speaking Amazon Mechanical Turkers (MTurk's) living in the US. Tweets are rated on a scale from 1 to 9 with a 'Do not know' option, and guidelines regarding how to rate the tweets are provided to the human raters. The tweets are selected to contain only English text, no mentions and no URL's to make them easier to rate without any additional contextual information. Tweets where more than half of the evaluators chose 'Do not know' are removed (98 tweets).

For each tweet, a MTurk rating is selected at random to be the 'human evaluation', whereas the remaining MTurk ratings are averaged to form the ground truth. The sentiment label for a given tweet is thus the overall consensus among raters (not incl. the randomly-selected 'human evaluation' rater). To ensure that the label categories are clearly separated the neutral tweets in the interval $[4.5, 5.5]$ are removed (roughly 29% of the tweets). The remaining dataset consists of 7347 tweets. Of these tweets, 5000 are used for training/validation and the remaining are used as the test set. Our DeepMoji model is trained using the seqthraw transfer learning approach.

Table 8 shows that the agreement of the random MTurk rater is 76.1%, meaning that a random rater will agree with the consensus on the polarity with the population of raters 76.1% of the time. Our DeepMoji model achieves 79.4% agreement and it is therefore better at capturing the overall sentiment rating of the raters than a single rater.

Table 8: Comparison of agreement between classifiers and Amazon Mechanical Turkers on sentiment prediction of tweets.

	Agreement
Random	50.1%
fasttext	71.0%
MTurk	76.1%
DeepMoji	$\mathbf{79.4}\%$

6 Discussion

We have developed a method that exploits the large quantities of unlabeled data on social media to yield performance gains across 5 domains and 3 NLP tasks. Our method beats the state of the art across 5 out of 6 benchmarks. SemEval 2007 Task 14 is the only benchmark dataset, where the model cannot obtain a state-of-the-art performance. Containing only 250 training observations with an average of 7 tokens each, the SemEval 2007 Task 14 dataset has very limited information for the DeepMoji model to use for adapting to the new domain. Most datasets will have substantially more information for fine-tuning.

Our experiments find that our DeepMoji model is capable of capturing complex emotions across multiple domains. To find limitations of the model we plan to invite the public to help explore cases where our model fails through an online interface. In the same spirit we invite the reviewers to test our DeepMoji model at deepmoji.pythonanywhere.com using the login *deepmoji* and password *acl_2017*.

The emotions, sentiment and sarcasm thought to be conveyed by a specific text differ between individuals. It's interesting to see that our DeepMoji model has a higher agreement with the MTurk raters as a whole than an individual MTurk when rating sentiment of tweets.

References

Yoshua Bengio et al. 2012. Deep learning of representations for unsupervised and transfer learning. *ICML Unsupervised and Transfer Learning* 27:17–36.

Sven Buechel and Udo Hahn. 2016. Emotion analysis as a regression problem - dimensional models and their implications on emotion representation and metrical evaluation. In *ECAI 2016. 22nd European Conference on Artificial Intelligence*.

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*. ACM, pages 160–167.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *Icml*. volume 32, pages 647–655.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359* .

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11(Feb):625–660.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*. pages 1019–1027.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision .

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993* .

Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in neural information processing systems*. pages 919–927.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* .

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

David Krueger and Roland Memisevic. 2016. Regularizing rnns by stabilizing activations. In *Proceeding of the International Conference on Learning Representations*.

FA Kunneman, CC Liebrecht, and APJ van den Bosch. 2014. The (un) predictability of emotional hashtags in twitter .

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725* .

Saif M Mohammad. 2012. # emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, pages 246–255.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. page 31.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pages 79–86.

Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop*. Association for Computational Linguistics, pages 43–48.

Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. 2013. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold .

Valentina Sintsova, Claudiu-Cristian Musat, and Pearl Pu. 2013. Fine-grained emotion recognition in olympic tweets based on human computation. In *4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. EPFL-CONF-197185.

Jonas Sjöberg and Lennart Ljung. 1995. Overtraining, regularization and searching for a minimum, with application to neural networks. *International Journal of Control* 62(6):1391–1407.

Jacopo Staiano and Marco Guerini. 2014. Depechemood: A lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605* .

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

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*. Springer, pages 121–136.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology* 63(1):163–173.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 61(12):2544–2558.

Duy Tin Vo and Yue Zhang. 2016. Dont count, predict! an automatic approach to learning sentiment lexicons for short text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. volume 2, pages 219–224.

Harald G Wallbott and Klaus R Scherer. 1986. How universal and specific is emotional experience? evidence from 27 countries on five continents. *Information (International Social Science Council)* 25(4):763–795.

Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. 2010. A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*. Association for Computational Linguistics, pages 60–68.