EmoNet: Fine-Grained Emotion Detection with Gated Recurrent Neural Networks

Muhammad Abdul-Mageed
School of Library, Archival & Information Studies
University of British Columbia
muhammad.mageed@ubc.ca

Lyle Ungar
Computer and Information Science
University of Pennsylvania
ungar@cis.upenn.edu

Abstract

Accurate detection of emotion from natural language has applications ranging from building emotional chatbots to better understanding individuals and their lives. However, progress on emotion detection has been hampered by the absence of large labeled datasets. In this work, we build a very large dataset for fine-grained emotions and develop deep learning models on it. We achieve a new state-of-the-art on 24 fine-grained types of emotions (with an average accuracy of 87.58%). We also extend the task beyond emotion types to model Robert Plutchik's 8 primary emotion dimensions, acquiring a superior accuracy of 95.68%.

1 Introduction

According to the *Oxford English Dictionary*, emotion is defined as "[a] strong feeling deriving from one's circumstances, mood, or relationships with others." [1] This "standard" definition identifies emotions as constructs involving something innate that is often invoked in social interactions and that aids in communicating with others(Hwang and Matsumoto, 2016). It is no exaggeration that humans are emotional beings: Emotions are an integral part of human life, and affect our decision making as well as our mental and physical health. As such, developing emotion detection models is important; they have a wide array of applications, ranging from building nuanced virtual assistants that cater for the emotions of their users to detecting the emotions of social media users in order to understand their mental and/or physical health. However, emotion detection has remained a challenging task, partly due to the limited availability of labeled data and partly due the controversial nature of what emotions themselves are (Aaron C. Weidman and Tracy, 2017).

Recent advances in machine learning for natural language processing (NLP) suggest that, given enough labeled data, there should be an opportunity to build better emotion detection models. Manual labeling of data, however, is costly and so it is desirable to develop labeled emotion data without annotators. While the proliferation of social media has made it possible for us to acquire large datasets with implicit labels in the form of hashtags (Mohammad and Kiritchenko, 2015), such labels are noisy and reliable.

In this work, we seek to enable deep learning by creating a large dataset of fine-grained emotions using Twitter data. More specifically, we harness cues in Twitter data in the form of emotion hashtags as a way to build a labeled emotion dataset that we then exploit using *distant supervision* (Mintz et al., 2009) (the use of hashtags as a surrogate for annotator-generated emotion labels) to build emotion models grounded in psychology. We construct such a dataset and exploit it using powerful deep learning methods to build accurate, high coverage models for emotion prediction. Overall, we make the following contributions: 1) Grounded in psychological theory of emotions, we build a large-scale, high quality dataset of tweets labeled with emotions. Key to this are methods to ensure data quality, 2) we validate the data collection method using human annotations, 3) we develop powerful deep learning models using a gated recurrent network to exploit the data, yielding new state-of-the-art on 24 fine-grained types of emotions, and 4) we extend the task beyond these emotion types to model Plutick's 8 primary emotion dimensions.

[1] https://en.oxforddictionaries.com/definition/emotion.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 718–728
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1067

Our emotion modeling relies on *distant supervision* (Read, 2005; Mintz et al., 2009), the approach of using cues in data (e.g., hashtags or emoticons) as a proxy for "ground truth" labels as we explained above. Distant supervision has been investigated by a number of researchers for emotion detection (Tanaka et al., 2005; Mohammad, 2012; Purver and Battersby, 2012; Wang et al., 2012; Pak and Paroubek, 2010; Yang et al., 2007) and for other semantic tasks such as sentiment analysis (Read, 2005; Go et al., 2009) and sarcasm detection (González-Ibánez et al., 2011). In these works, authors successfully use emoticons and/or hashtags as marks to label data after performing varying degrees of data quality assurance. We take a similar approach, using a larger collection of tweets, richer emotion definitions, and stronger filtering for tweet quality.

The remainder of the paper is organized as follows: We first overview related literature in Section 2, describe our data collection in Section 3.1, and the annotation study we performed to validate our distant supervision method in Section 4. We then describe our methods in Section 5, provide results in Section 6, and conclude in Section 8.

2 Related Work

2.1 Computational Treatment of Emotion

The SemEval-2007 Affective Text task (Strapparava and Mihalcea, 2007) [SEM07] focused on classification of emotion and valence (i.e., positive and negative texts) in news headlines. A total of 1,250 headlines were manually labeled with the 6 basic emotions of Ekman (Ekman, 1972) and made available to participants. Similarly, (Aman and Szpakowicz, 2007) describe an emotion annotation task of identifying emotion category, emotion intensity and the words/phrases that indicate emotion in blog post data of 4,090 sentences and a system exploiting the data. Our work differs from both that of SEM07 (Strapparava and Mihalcea, 2007) and (Aman and Szpakowicz, 2007) in that we focus on a different genre (i.e., Twitter) and investigate distant supervision as a way to acquire a significantly larger labeled dataset.

Our work is similar to (Mohammad, 2012; Mohammad and Kiritchenko, 2015), (Wang et al., 2012), and (Volkova and Bachrach, 2016) who use distant supervision to acquire Twitter data with emotion hashtags and report analyses and experiments to validate the utility of this approach. For example, (Mohammad, 2012) shows that by using a simple domain adaptation method to train a classifier on their data they are able to improve both precision and recall on the SemEval-2007 (Strapparava and Mihalcea, 2007) dataset. As the author points out, this is another premise that the self-labeled hashtags acquired from Twitter are consistent, to some degree, with the emotion labels given by the trained human judges who labeled the SemEval-2007 data. As pointed out earlier, (Wang et al., 2012) randomly sample a set of 400 tweets from their data and human-label as relevant/irrelevant, as a way to verify the distant supervision approach with the quality assurance heuristics they employ. The authors found that the precision on a test set is 93.16%, thus confirming the utility of the heuristics. (Wang et al., 2012) provide a number of important observations, as conclusions based on their work. These include that since they are provided by the tweets' writers, the emotion hashtags are more natural and reliable than the emotion labels traditionally assigned by annotators to data by a few annotators. This is the case since in the lab-condition method annotators need to infer the writers emotions from text, which may not be accurate. Additionally, (Volkova and Bachrach, 2016) follow the same distant supervision approach and find correlations of users' emotional tone and the perceived demographics of these users' social networks exploiting the emotion hashtag-labeled data. Our dataset is more than an order of magnitude larger than (Mohammad, 2012) and (Volkova and Bachrach, 2016) and the range of emotions we target is much more fine grained than (Mohammad, 2012; Wang et al., 2012; Volkova and Bachrach, 2016) since we model 24 emotion types, rather than focus on ≤ 7 basic emotions.

(Yan et al., 2016; Yan and Turtle, 2016a,b) develop a dataset of 15,553 tweets labeled with 28 emotion types and so target a fine-grained range as we do. The authors instruct human annotators under lab conditions to assign any emotion they feel is expressed in the data, allowing them to assign more than one emotion to a given tweet. A set of 28 chosen emotions was then decided upon and further annotations were performed using Amazon Mechanical Turk (AMT). The authors cite an agreement of 0.50 Krippendorff's alpha (α) between the lab/expert annotators, and an (α) of 0.28 between experts and AMT workers. EmoTweet-

28 is a useful resource. However, the agreement between annotators is not high and the set of assigned labels do not adhere to a specific theory of emotion. We use a much larger dataset and report an accuracy of the hashtag approach at 90% based on human judgement as reported in Section 4.

2.2 Mood

A number of studies have also been performed to analyze and/or model mood in social media data. (De Choudhury et al., 2012) identify more than 200 moods frequent on Twitter as extracted from psychological literature and filtered by AMT workers. They then collect tweets which have one of the moods in their mood lexicon in the form of a hashtag. To verify the quality of the mood data, the authors run AMT studies where they ask workers whether a tweet displayed the respective mood hashtag or not and find that in 83% of the cases hashtagged moods at the end of posts did capture users' moods, whereas for posts with mood hashtags anywhere in the tweet, only 58% of the cases capture the mood of users. Although they did not build models for mood detection, the annotation studies (De Choudhury et al., 2012) perform further support our specific use of hashtags to label emotions. (Mishne and De Rijke, 2006) collect user-labeled mood from blog post text on LiveJournal and exploit them for predicting the intensity of moods over a time span rather than at the post level. Similarly, (Nguyen, 2010) builds models to infer patterns of moods in a large collection of LiveJournal posts. Some of the moods in these LiveJournal studies (e.g., *hungry*, *cold*), as (De Choudhury et al., 2012) explain, would not fit any psychological theory. Our work is different in that it is situated in psychological theory of emotion.

2.3 Deep Learning for NLP

In spite of the effectiveness of feature engineering for NLP, it is a labor intensive task that also needs domain expertise. More importantly, feature engineering falls short of extracting and organizing all the discriminative information from data (LeCun et al., 2015; Goodfellow et al., 2016). Neural networks (Goodfellow et al., 2016) have emerged as a successful class of methods that has the power of automatically discovering the representations needed for detection or classification and has been successfully applied to multiple NLP tasks. A line of studies in the literature (e.g., (Labutov and Lipson, 2013; Maas et al., 2011; Tang et al., 2014b,a) aim to learn sentiment-specific word embeddings (Bengio et al., 2003; Mikolov et al., 2013) from neighboring text. Another thread of research focuses on learning semantic composition (Mitchell and Lapata, 2010), including extensions to phrases and sentences with recursive neural networks (a class of syntax-tree models) (Socher et al., 2013; Irsoy and Cardie, 2014; Li et al., 2015) and to documents with distributed representations of sentences and paragraphs (Le and Mikolov, 2014; Tang et al., 2015) for modeling sentiment.

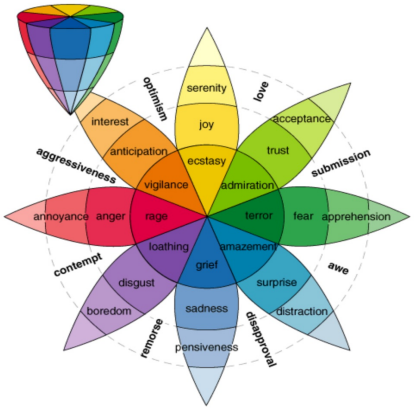
Long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Neural Nets (GRNNs) (Cho et al., 2014; Chung et al., 2015), variations of recurrent neural networks (RNNs), a type of networks suitable for handling time-series data like speech (Graves et al., 2013) or handwriting recognition (Graves, 2012; Graves and Schmidhuber, 2009), have also been used successfully for sentiment analysis (Ren et al., 2016; Liu et al., 2015; Tai et al., 2015; Tang et al., 2015; Zhang et al., 2016). Convolutional neural networks (CNNs) have also been quite successful in NLP, and have been applied to a range of sentence classification tasks, including sentiment analysis (Blunsom et al., 2014; Kim, 2014; Zhang et al., 2015). Other architectures have also been recently proposed (e.g., (Bradbury et al., 2016)). A review of neural network methods for NLP can be found in (Goldberg, 2016).

3 Data

3.1 Collection of a Large-Scale Dataset

To be able to use deep learning for modeling emotion, we needed a large dataset of labeled tweets. Since there is no such human-labeled dataset publicly available, we follow (Mohammad, 2012; Mintz et al., 2009; Purver and Battersby, 2012; González-Ibánez et al., 2011; Wang et al., 2012) in adopting *distant supervision*: We collect tweets with emotion-carrying hashtags as a surrogate for emotion labels. To be able to collect enough tweets to serve our need, we developed a list of hashtags representing each of the 24 emotions proposed by Robert Plutchick (Plutchik, 1980, 1985, 1994). Plutchik (Plutchik, 2001) organizes emotions in a three-dimensional circumplex model analogous to the colors on a color wheel. The cone's vertical dimension represents intensity, and the 3 circle represent degrees of similarity

Figure 1: Plutchik's wheel of emotion.



among the various emotion types. The eight sectors are meant to capture that there are eight primary emotion dimensions arranged as four pairs of opposites. Emotions in the blank spaces are the primary emotion dyads (i.e., emotions that are mixtures of two of the primary emotions). For this work, we exclude the dyads in the exploded model from our treatment. For simplicity, we refer to the circles as `plutchik-1`: with the emotions {*admiration, amazement, ecstasy, grief, loathing, rage, terror, vigilance*}, `plutchik-2`: with the emotions {*joy, trust, fear, surprise, sadness, disgust, anger, anticipation*}, and `plutchik-3`: with the emotions {*acceptance, annoyance, apprehension, boredom, distraction, interest, pensiveness, serenity*}. The wheel is shown in Figure 1.

For each emotion type, we prepared a seed set of hashtags representing the emotion. We used Google synonyms and other online dictionaries and thesauri (e.g., `www.thesaurus.com`) to expand the initial seed set of each emotion. We acquire a total of 665 emotion hashtags across the 24 emotion types. For example, for the *joy* emotion, a subset of the seeds in our expanded set is {*"happy", "happiness", "joy", "joyful", "joyfully", "delighted", "feelingsunny", "blithe", "beatific", "exhilarated", "blissful", "walkingonair", "jubilant"*}. We then used the expanded set to extract tweets with hashtags from the set from a number of massive-scale in-house Twitter datasets. We also used Twitter API to crawl Twitter with hashtags from the expanded set. Using this method, we were able to acquire a dataset of about 1/4 billion tweets covering an extended time span from July 2009 till January 2017.

3.2 Preprocessing and Quality Assurance

Twitter data are very noisy, not only because of use of non-standard typography (which is less of a problem here) but due to the many duplicate tweets and the fact that tweets often have multiple emotion hashtags. Since these reduce our ability to build accurate models, we need to clean the data and remove duplicates. Starting with $>$ 1/4 billion tweets, we employ a rigorous and strict pipeline. This results in a vastly smaller set of about 1.6 million dependable labeled tweets.

Since our goal is to create non-overlapping categories at the level of a tweet, we first removed all tweets with hashtags belonging to more than one emotion of the 24 emotion categories. Since it was observed (e.g., (Mohammad, 2012; Wang et al., 2012)) and also confirmed by our annotation study as described in Section 4, that hashtags in tweets with URLs are less likely to correlate with a true emotion label, we remove all tweets with URLs from our data. We filter out duplicates using a two-step procedure: 1) we remove all retweets (based on existence of the token "RT" regardless of case) and 2) we use the Python library *pandas* `http://pandas.pydata.org/` "drop_duplicates" method to compare the tweet texts of all the tweets after normalizing character repetitions [all consecutive characters of $>$ 2 to 2] and user mentions (as detected by a string starting with an "@" sign). We then performed a manual inspection of a random sample of 1,000 tweets from the data and found no evidence of any remaining tweet duplicates.

Next, even though the emotion hashtags themselves are exclusively in English, we observe the data do have tweets in languages other than English. This is due to code-switching, but also to the fact that our data dates back to 2009 and Twitter did not allow use of hashtags for several non-English languages until 2012. To filter out non-English, we use the langid (Lui and Baldwin, 2012) (`https://github.com/saffsd/langid.py`) library to assign language tags to the tweets. Since the common wisdom in the literature (e.g., (Mohammad, 2012; Wang et al., 2012)) is to restrict data to hashtags occurring in final position of a tweet, we investigate correlations between a tweet's relevance and emotion hashtag location in Section 4 and test models exclusively on data with hashtags occurring in final position. We also only use tweets con-

taining at least 5 words.

Table 2 shows statistics of the data after applying our cleaning, filtering, language identification, and deduplication pipeline. Since our focus is on English, we only show statistics for tweets tagged with an "en" (for "English") label by langid. Table 2 provides three types of relevant statistics: 1) counts of all tweets, 2) counts of tweets with at least 5 words and the emotion hashtags occurring in the *last quarter* of the tweet text (based on character count), and 3) counts of tweets with at least 5 words and the emotion hashtags occurring as the *final* word in the tweet text. As the last column in Table 2 shows, employing our most strict criterion where an emotion hashtag must occur finally in a tweet of a minimal length 5 words, we acquire a total of 1,608,233 tweets: 205,125 tweets for `plutchik-1`, 790,059 for `plutchik-2`, and 613,049 for `plutchik-3`. [2]

Emotion	ct	ct@lq	ct@end
admiration	292,153	150,509	112,694
amazement	568,255	358,472	34,826
ecstasy	54,174	34,307	23,856
grief	102,980	33,141	12,568
loathing	90,465	41,787	456
rage	30,994	11,777	4,749
terror	84,827	25,908	15,268
vigilance	6,171	1,028	708
plutchik-1	**1,230,019**	**656,929**	**205,125**
anger	131,082	82,447	56,472
anticipation	67,175	36,846	26,655
disgust	212,770	145,052	52,067
fear	302,989	153,513	98,657
joy	974,226	522,689	330,738
sadness	1,252,192	762,901	142,300
surprise	143,755	78,570	53,915
trust	198,619	103,332	29,255
plutchik-2	**3,282,808**	**1,885,350**	**790,059**
acceptance	138,899	54,706	16,522
annoyance	954,027	791,869	364,135
apprehension	29,174	11,650	7,828
boredom	872,246	583,994	152,105
distraction	122,009	52,633	617
interest	113,555	67,216	56,659
pensiveness	11,751	5,012	3,513
serenity	97,467	36,817	11,670
plutchik-3	**2,339,128**	**1,603,897**	**613,049**
ALL	**6,851,955**	**4,146,176**	**1,608,233**

Table 2: Data statistics.

4 Annotation Study

In their work, (Wang et al., 2012) manually label a random sample of 400 tweets extracted with hashtags in a similar way as we acquire our data and find that human annotators agree 93% of the time with the hashtag emotion type if the hashtag occurs as the last word in the tweet. We wanted to validate our use of hashtags in a similar fashion and on a bigger random sample. We had human annotators label a random sample of 5,600 tweets that satisfy our preprocessing pipeline. Manual inspection during annotation resulted in further removing a negligible 16 tweets that were found to have problems. For each of the remaining 5,584 tweets, the annotators assign a binary tag from the set {*relevant, irrelevant*} to indicate whether a tweet carries an emotion category as assigned using our distant supervision method or not. Annotators assigned 61.37% ($n = 3,427$) "relevant" tags and 38.63% ($n = 2,157$) "irrelevant" tags. Our analysis of this manually labeled dataset also supports the findings of (Wang et al., 2012): When we limit position of the emotion hashtag to the end of a tweet, we acquire 90.57% relevant data. We also find that if we relax the constraint on the hashtag position such that we allow the hashtag to occur in the last quarter of a tweet (based on a total tweet character count), we acquire 85.43% relevant tweets. We also find that only 23.20% ($n = 795$ out of $3,427$) of the emotion carrying tweets have the emotion hashtags occurring in final position, whereas 31.75% ($n = 1,088$ out of $3,427$) of the tweets have the emotion hashtags in the last quarter of the tweet string. This shows how enforcing a final hashtag location results in loss of a considerable number of emotion tweets. As shown in Table 2, only $1,608,233$ tweets out of a total of $6,851,955$ tweets ($\% = 23,47$) in our bigger dataset have emotion hashtags occurring in final position. Overall, we agree with (Mohammad, 2012; Wang et al., 2012) that the accuracy acquired by enforcing a strict pipeline and limiting to emotion hashtags to final position is a reasonable measure for warranting good-quality data for training supervised systems, an assumption we have also validated with our empirical findings here.

One advantage of using distant supervision under these conditions for labeling emotion data, as (Wang et al., 2012) also notes, is that the label is assigned by the writer of the tweet himself/herself rather than an annotator who could wrongly decide what category a tweet is. After all, emotion is a fuzzy concept and $> 90\%$ agreement as we

[2] The data can be acquired by emailing the first author. The distribution is in the form of tweet ids and labels, to adhere to Twitter conditions.

report here is higher than the human agreement usually acquired on many NLP tasks. Another advantage of this method is obviously that it enables us to acquire a sufficiently large training set to use deep learning. We now turn to describing our deep learning methods.

5 Methods

For our core modeling, we use *Gated Recurrent Neural Networks (GRNNs)*, a modern variation of *recurrent neural networks (RNNs)*, which we now turn to introduce. For notation, we denote scalars with italic lowercase (e.g., x), vectors with bold lowercase (e.g.,$\mathbf{x}$), and matrices with bold uppercase (e.g.,$\mathbf{W}$).

Recurrent Neural Network A recurrent neural network (RNN) is one type of neural network architecture that is particularly suited for modeling sequential information. At each time step t, an RNN takes an input vector $\mathbf{x}_t \in \mathbb{R}^n$ and a hidden state vector $\mathbf{h}_{t-1} \in \mathbb{R}^m$ and produces the next hidden state $\mathbf{h}_t$ by applying the recursive operation:

$$\mathbf{h}_t = f\left(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}\right) \quad (1)$$

Where the input to hidden matrix $\mathbf{W} \in \mathbb{R}^{mxn}$, the hidden to hidden matrix $\mathbf{U} \in \mathbb{R}^{mxm}$, and the bias vector $\mathbf{b} \in \mathbb{R}^m$ are parameters of an affine transformation and f is an element-wise nonlinearity. While an RNN can in theory summarize all historical information up to time step $\mathbf{h}_t$, in practice it runs into the problem of vanishing/exploding gradients (Bengio et al., 1994; Pascanu et al., 2013) while attempting to learn long-range dependencies.

LSTM Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) addresses this exact problem of learning long-term dependencies by augmenting an RNN with a memory cell $\mathbf{c}_t \in \mathbb{R}^n$ at each time step. As such, in addition to the input vector $\mathbf{x}_t$, the hiddent vector $\mathbf{h}_{t-1}$, an LSTM takes a cell state vector $\mathbf{c}_{t-1}$ and produces $\mathbf{h}_t$ and $\mathbf{c}_t$ via the following calculations:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}^i x_t + \mathbf{U}^i \mathbf{h}_{t-1} + \mathbf{b}^i\right) \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}^f x_t + \mathbf{U}^f \mathbf{h}_{t-1} + \mathbf{b}^f\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}^o x_t + \mathbf{U}^o \mathbf{h}_{t-1} + \mathbf{b}^o\right) \\
\mathbf{g}_t &= \tanh\left(\mathbf{W}^g x_t + \mathbf{U}^g \mathbf{h}_{t-1} + \mathbf{b}^g\right) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \quad (2)$$

Where $\sigma(\cdot)$ and $\tanh(\cdot)$ are the element-wise sigmoid and hyperbolic tangent functions, $\odot$ the element-wise multiplication operator, and $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are the *input*, forget, and *output* gates. The $\mathbf{g}_t$ is a new memory cell vector with candidates that could be added to the state. The LSTM parameters $\mathbf{W}_j$, $\mathbf{U}_j$, and $\mathbf{b}_j$ are for $j \in \{i, f, o, g\}$.

GRNNs (Cho et al., 2014; Chung et al., 2015) propose a variation of LSTM with a *reset gate* $\mathbf{r}_t$, an update state $\mathbf{z}_t$, and a new simpler hidden unit $\mathbf{h}_t$, as follows:

$$\begin{aligned}
\mathbf{r}_t &= \sigma\left(\mathbf{W}^r x_t + \mathbf{U}^r \mathbf{h}_{t-1} + \mathbf{b}^r\right) \\
\mathbf{z}_t &= \sigma\left(\mathbf{W}^z x_t + \mathbf{U}^z \mathbf{h}_{t-1} + \mathbf{b}^z\right) \\
\tilde{\mathbf{h}}_t &= \tanh\left(\mathbf{W} x_t + \mathbf{r}_t * \mathbf{U}^{\tilde{h}} \mathbf{h}_{t-1} + \mathbf{b}^{\tilde{h}}\right) \\
\mathbf{h}_t &= \mathbf{z}_t * \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) * \tilde{\mathbf{h}}_t
\end{aligned} \quad (3)$$

The GRNN parameters $\mathbf{W}_j$, $\mathbf{U}_j$, and $\mathbf{b}_j$ are for $j \in \{r, z, \tilde{h}\}$. In this set up, the hidden state is forced to ignore a previous hidden state when the reset gate is close to 0, thus enabling the network to forget or drop irrelevant information. Additionally, the update gate controls how much information carries over from a previous hidden state to the current hidden state (similar to an LSTM memory cell). We use GRNNs as they are simpler and faster than LSTM. For GRNNs, we use Theano (Theano Development Team, 2016).

Online Classifiers We compare the performance of the GRNNs to four online classifiers that are capable of handling the data size: Stochastic Gradient Descent (SGD), Multinomial Naive Bayes (MNB), Perceptron, and the Passive Agressive Classifier (PAC). These classifiers learn online from mini-batches of data. We use mini-batches of 10,000 instances with all the four classifiers. We use the *scikit-learn* implementation of these classifiers (`http://scikit-learn.org`).

why SGD is a classifier?

Settings We aim to model Plutchik's 24 fine-grained emotions as well as his 8 primary emotion dimensions where each 3 related types of emotion (perceived as varying in intensity) are combined in one dimension. We now turn to describing our experiments experiments.

6 Experiments

6.1 Predicting Fine-Grained Emotions

As explained earlier, Plutchik organizes the 24 emotion types in the 3 main circles that we will refer to as *plutchik-1*, *plutchik-2*, and *plutchik-3*.

Emotion	Qadir (2013)		Roberts (2012)		MD (2015)		Wang (2012)		Volkova (2016)		This work	
anger	400	0.44	583	0.64	1,555	0.28	457,972	0.72	4,963	0.80	56,472	0.75
anticip	-	-	-	-	-	-	-	-	-	-	26,655	0.70
disgust	-	-	922	0.67	761	0.19	-	-	12,948	0.92	52,067	0.82
fear	592	0.54	222	0.74	2,816	0.51	11,156	0.44	9,097	0.77	98,657	0.74
joy	1,005	0.59	716	0.68	8,240	0.62	567,487	0.72	15,559	0.79	330,738	0.91
sadness	560	0.46	493	0.69	3,830	0.39	489,831	0.65	4,232	0.62	142,300	0.73
surprise	-	-	324	0.61	3849	0.45	1,991	0.14	8,244	0.64	53,915	0.86
trust	-	-	-	-	-	-	-	-	-	-	29,255	0.82
ALL	**4,500**	**0.53**	**3,777**	**0.67**	**21,051**	**0.49**	**1,991,184**	**-**	**52,925**	**0.78**	**790,059**	**0.83**

how to use F?

Table 6: Comparison (in *F*-score) of our results with GRNNs to published literature. MD = Mohammad (2015). *Note:* For space restrictions, we take the liberty of using the last name of only the first author of each work.

Emotion	SGD	MNB	PRCPTN	PAC
baseline	60.00	60.00	60.00	60.00
admiration	78.30	78.01	74.24	79.86
amazement	37.57	35.71	42.51	46.69
ecstasy	51.53	51.89	47.37	53.53
grief	38.64	36.94	37.33	48.10
loathing	0.00	0.00	2.09	2.99
rage	3.47	4.49	14.02	17.04
terror	33.23	44.12	40.48	47.00
vigilance	2.53	2.56	5.52	8.42
plutchik-1	**60.26**	**60.54**	**59.11**	**64.86**
anger	19.41	13.84	24.54	29.26
anticipation	7.46	12.63	17.29	26.70
disgust	29.51	29.87	31.83	36.60
fear	21.45	25.49	30.41	33.59
joy	72.83	72.96	72.32	75.50
sadness	50.04	51.72	39.58	49.21
surprise	8.46	4.75	17.34	19.54
trust	42.09	38.52	44.48	47.51
plutchik-2	**48.05**	**48.33**	**48.60**	**53.30**
acceptance	0.12	2.74	13.98	13.04
annoyance	80.28	80.71	78.80	81.47
apprehension	0.80	0.00	9.72	10.66
boredom	49.53	51.27	52.02	57.84
distraction	0.00	2.99	3.42	0.00
interest	21.69	30.45	34.85	44.14
pensiveness	2.61	8.08	11.22	12.27
serenity	8.87	19.57	27.23	38.59
plutchik-3	**62.20**	**64.00**	**64.04**	**68.14**
ALL	**56.84**	**57.62**	**57.25**	**62.10**

Table 3: Results in *F*-score with traditional online classifiers.

We model the set of emotions belonging to each of the 3 circles independently, thus casting each as an 8-way classification task. Inspired by observations from the literature and our own annotation study, we limit our data to tweets of at least 5 words with an emotional hashtag occurring at the end. We then split the data representing each of the 3 circles into 80% training (TRAIN), 10% development (DEV), and 10% testing (TEST). As mentioned above, we run experiments with a range of online, out-of-core classifiers as well as the GRNNs. To train the GRNNs, we optimize the hyper-parameters of the network on a development set as we describe below, choosing a vocabulary size of 80K words (a vocabulary size we also use for the out-of-core classifiers), a word embedding vector of size 300 dimensions learnt directly from the training data, an input maximum length of 30 words, 7 epochs, and the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 0.001. We use 3 dense layers each with 1,000 units. We use dropout (Hinton et al., 2012) for regularization, with a dropout rate of 0.5. For our loss function, we use categorical cross-entropy. We use a mini-batch (Cotter et al., 2011) size of 128. We found this architecture to work best with almost all the settings and so we fix it across the board for all experiments with GRNNs.

Results with Traditional Classifiers Results with the online classifiers are presented in terms of *F*-score in Table 3. As the table shows, among this group of classifiers, the Passive Agressive classifier (PAC) acquires the best performance. PAC achieves an overall *F*-score of 64.86% on `plutchik-1`, 53.30% on `plutchik-2`, and 68.14% on `plutchik-3`, two of which are higher than an arbitrary baseline[3] of 60%.

Results with GRNNs Table 4 presents results with GRNNs, compared with the best results using the traditional classifiers as acquired with PAC. As the table shows, the GRNN models are very successful across all the 3 classification tasks. With GRNNs, we acquire an overall *F*-scores of: 91.21% on `plutchik-1`, 82.32% on `plutchik-2`, and 87.47% on `plutchik-3`. These results are 26.35%, 29.02%, and 25.37% higher than PAC, respectively.

what for 3?

Negative Results We experiment with aug-

[3]The arbitrary baseline is higher than the majority class in the training data in any of the 3 cases.

	PAC	GRNNs		
Emotion	**f-score**	**prec**	**rec**	**f-score**
admiration	79.86	94.53	95.28	94.91
amazement	46.69	90.44	89.02	89.73
ecstasy	53.53	83.49	90.01	86.62
grief	48.10	85.07	81.13	83.05
loathing	2.99	83.87	54.17	65.82
rage	17.04	80.00	75.11	77.48
terror	47.00	91.15	84.01	87.44
vigilance	8.42	71.93	70.69	71.30
plutchik-1	**64.86**	**91.26**	**91.24**	**91.21**
anger	29.26	74.95	69.20	71.96
anticipation	26.70	70.05	69.00	69.52
disgust	36.60	82.18	68.84	74.92
fear	33.59	73.74	72.51	73.12
joy	75.50	90.96	93.88	92.40
sadness	49.21	73.20	82.04	77.37
surprise	19.54	85.60	67.40	75.42
trust	47.51	82.43	76.83	79.53
plutchik-2	**53.30**	**82.53**	**82.46**	**82.32**
acceptance	13.04	77.10	71.76	74.33
annoyance	81.47	91.46	95.01	93.20
apprehension	10.66	80.40	61.07	69.41
boredom	57.84	85.95	84.40	85.16
distraction	0.00	87.50	25.00	38.89
interest	44.14	86.79	78.38	82.37
pensiveness	12.27	91.87	43.24	58.80
serenity	38.59	82.15	78.16	80.11
plutchik-3	**68.14**	**88.94**	**89.08**	**88.89**
ALL	**62.10**	**87.58**	**87.59**	**87.47**

what is that

Table 4: Results with GRNNs across Plutchik's 24 emotion categories. We compare to best-performing traditional classifier (i.e. Passive Aggressive).

menting training data reported here in two ways: 1) For each emotion type, we concatenate the training data with training data of tweets that are more (or less) intense from the same sector/dimension in the wheel, and 2) for each emotion type, we add tweets where emotion hashtags occur in the last quarter of a tweet (which were originally filtered out from TRAIN). However, we gain no improvements based on either of these methods, thus reflecting the importance of using high-quality training data and the utility of our strict pipeline.

6.2 Predicting 8 Primary Dimensions

We now investigate the task of predicting each of the 8 primary emotion dimensions represented by the sectors of the wheel (where the three degrees of intensity of a given emotion are reduced to a single emotion dimension [e.g., {*ecstasy, joy, serenity*} are reduced to the *joy* dimension]). We concatenate the 80% training data (TRAIN) from each of the 3 circles' data into a single training set

Dimension	**prec**	**rec**	**f-score**
anger	97.40	97.72	97.56
anticipation	91.18	89.95	90.56
disgust	96.20	93.94	95.06
fear	94.97	94.38	94.68
joy	94.61	96.40	95.50
sadness	95.52	95.25	95.39
surprise	94.99	91.62	93.27
trust	96.36	97.58	96.96
All	**95.68**	**95.68**	**95.68**

accuracy?

Table 5: GRNNs results across 8 emotion dimensions. Each dimension represents three different emotions. For example, the *joy* dimension represents *serenity, joy* and *ecstasy*.

Emotion	**Volkova (2016) model**	**This work**
anger	12.38	74.95
disgust	5.71	82.18
fear	11.18	73.74
joy	44.57	90.96
sadness	18.04	73.20
surprise	5.33	85.60
ALL	**26.95**	**80.12**

what is this..

Table 7: Comparison (in acc) to (Volkova and Bachrach, 2016)'s model.

(TRAIN-ALL), the 10% DEV to form DEV-ALL, and the 10% TEST to form TEST-ALL. We test a number of hyper-parameters on DEV and find the ones we have identified on the fine-grained prediction to work best and so we adopt them as is with the exception of limiting to only 2 epochs. We believe that with a wider exploration of hyper-parameters, improvements could be possible. As Table 5 shows, we are able to model the 8 dimensions with an overall superior accuracy of 95.68%. As far as we know, this is the first work on modeling these dimensions.

7 Comparisons to Other Systems

We compare our results on the 8 basic emotions to the published literature. As Table 6 shows, on this subset of emotions, our system is 4.53% (acc) higher than the best published results (Volkova and Bachrach, 2016), facilitated by the fact that we have an order of magnitude more training data. As shown in Table 7, we also apply (Volkova and Bachrach, 2016)'s pre-trained model on our test set of the 6 emotions they predict (which belong to `plutchik-2`), and acquire an overall accuracy of 26.95%, which is significantly lower than our accuracy.