Multi-Channel Convolutional Neural Network for Twitter Emotion and Sentiment Recognition

Jumayel Islam[1], **Robert E. Mercer**[1], **Lu Xiao**[2]
[1]Department of Computer Science, The University of Western Ontario
[2]School of Information Studies, Syracuse University
jislam3@uwo.ca, mercer@csd.uwo.ca, lxiao04@syr.edu

Abstract

The advent of micro-blogging sites has paved the way for researchers to collect and analyze huge volumes of data in recent years. Twitter, being one of the leading social networking sites worldwide, provides a great opportunity to its users for expressing their states of mind via short messages which are called tweets. The urgency of identifying emotions and sentiments conveyed through tweets has led to several research works. It provides a great way to understand human psychology and impose a challenge to researchers to analyze their content easily. In this paper, we propose a novel use of a multi-channel convolutional neural architecture which can effectively use different emotion and sentiment indicators such as hashtags, emoticons and emojis that are present in the tweets and improve the performance of emotion and sentiment identification. We also investigate the incorporation of different lexical features in the neural network model and its effect on the emotion and sentiment identification task. We analyze our model on some standard datasets and compare its effectiveness with existing techniques.

1 Introduction

Social networking sites (e.g., Twitter) have become immensely popular in the last decade. User generated content (e.g., blog posts, statuses, tweets etc.) in social media provides a wide range of opinionated, emotional and sentimental content which gives researchers a massive data source to explore. For example, Twitter, being one of the leading social networking giants, provides an online environment that allows people of various backgrounds and locations to share their opinions and views on different matters. As of July 2018, over 500 million tweets are sent per day having over 300 million monthly active users.[1]

There is often a misconception about sentiments and emotions as these subjectivity terms have been used interchangeably (Munezero et al., 2014). Munezero et al. (2014) differentiate these two terms along with other subjectivity terms and provide the computational linguistics community with clear concepts for effective analysis of text. While sentiment classification tasks deal with the polarity of a given text (whether a piece of text expresses positive, negative or neutral sentiment) and the intensity of the sentiment expressed, emotion mining tasks naturally deal with human emotions which in some end purposes are more desirable (Ren and Quan, 2012; Desmet and Hoste, 2013; Mohammad et al., 2015). Detecting emotion and sentiment from noisy twitter data is really challenging due to its nature. Tweets tend to be short in length and have a diverse vocabulary making them harder to analyze due to the limited contextual information they contain. In this study, we are interested in tackling these two tasks with a novel use of a single neural network architecture.

A number of emotion theories are available which suggest different sets of basic emotions. Interestingly, *joy, sadness, anger, fear* and *surprise* are common to all. To the best of our knowledge, the model suggested by Ekman (1999) is the most broadly used emotion model. In this study, we use Ekman's basic emotions together with other sets of emotions (Plutchik, 1984; Shaver et al., 1987).

In early textual emotion mining and sentiment analysis research, the usefulness of using external lexicons along with predefined rules has been demonstrated (Aman and Szpakowicz, 2008; Neviarouskaya et al., 2007; Bandhakavi et al., 2017; Thelwall et al., 2010; Gilbert, 2014). Aman and Szpakowicz (2008) used *Roget's Thesaurus* along with *WordNet-Affect* for fine-grained emotion prediction from blog data. Bandhakavi et al. (2017) propose a unigram mixture model (UMM)

[1] http://www.internetlivestats.com/twitter-statistics/

Proceedings of NAACL-HLT 2019, pages 1355–1365
Minneapolis, Minnesota, June 2 - June 7, 2019.

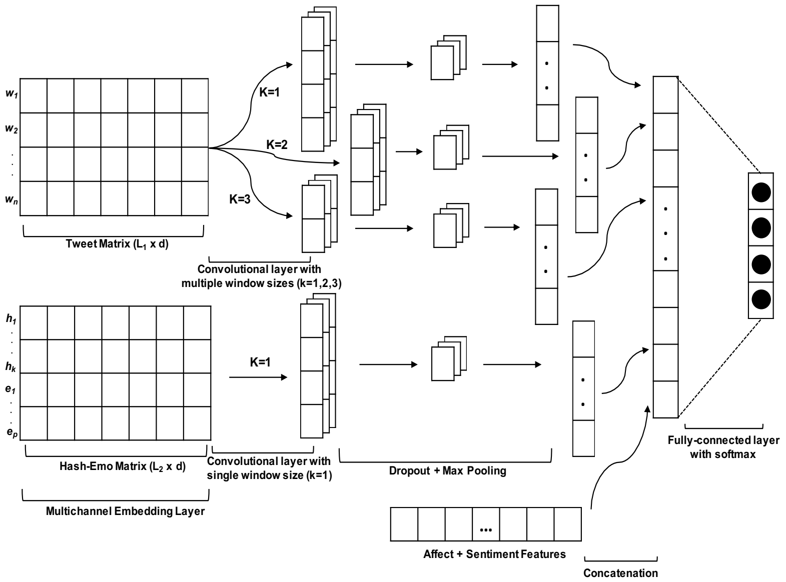
w_1
w_2
w_n
K=1
K=2
K=3
Tweet Matrix (L_1 x d)
Convolutional layer with multiple window sizes (k=1,2,3)
h_1
h_k
e_1
e_p
K=1
Hash-Emo Matrix (L_2 x d)
Convolutional layer with single window size (k=1)
Multichannel Embedding Layer
Dropout + Max Pooling
...
Affect + Sentiment Features
Concatenation
Fully-connected layer with softmax

Figure 1: Overview of the MC-CNN model

to create a domain-specific lexicon which performs better in extracting features than Point-wise Mutual Information and supervised Latent Dirichlet Allocation methods. Neviarouskaya et al. (2007) propose a rule-based system which can handle informal texts in particular. They built a database of abbreviations, emoticons, affect words, etc., in which each entry is labeled with an emotion and its intensity. Thelwall et al. (2010) propose an algorithm, *SentiStrength*, which utilizes a dictionary of sentiment words associated with strength measures to deal with short informal texts from social media. Gilbert (2014) propose *VADER*, a rule-based model for sentiment analysis. They built a lexicon which is specially attuned to microblog-like contexts and their model outperforms individual human raters. More recently, deep learning models have proven to be very successful when applied on various text-related tasks (Kim, 2014; Kalchbrenner et al., 2014; dos Santos and Gatti, 2014; Tai et al., 2015; Wang et al., 2016; Felbo et al., 2017; Abdul-Mageed and Ungar, 2017). Kim (2014) showed the effectiveness of a simple CNN model that leverages pre-trained word vectors for a sentence classification task. Kalchbrenner et al. (2014) propose a dynamic CNN model using a dynamic k-max pooling mechanism which is able to generate a feature graph which captures a variety of word relations. They showed the efficacy of their model by achieving high performances on binary and multi-class sentiment classification tasks without any feature engineering. dos Santos and Gatti (2014) propose a deep CNN model that uses both character and word-level information allowing them to achieve state-of-the-art performance on both binary and fine-grained multi-class sentiment classification for one of the twitter datasets. Tai et al. (2015) propose a Tree-LSTM model which captures syntactic properties in text. Their model performs particularly well on sentiment classification. Wang et al. (2016) propose a regional CNN-LSTM model for dimensional sentiment analysis. Their proposed model computes *valence-arousal* ratings from texts and outperforms several regression-based methods. Felbo et al. (2017) propose a bi-directional LSTM model with attention showing that their model learns better representations when distant supervision is expanded to a set of noisy labels. Abdul-Mageed and Ungar (2017) also used distant supervision to build a large twitter dataset and proposed a Gated Recurrent Neural Network model for fine-grained emotion detection.

The recent success of neural based models motivated us to take a different look at the sentiment and emotion prediction from the noisy twitter data task. Compared with sequential models, CNN models train relatively faster and seem to work very well on noisy data such as tweets which are grammatically error-prone. We decided to work with CNN models after our initial experiments suggested that they perform comparatively better than a simple BiLSTM model on twitter dataset. We address the following questions in this paper:

- Can CNN models be used in a way that can improve the performance of detecting emotion and sentiment from noisy Twitter data?

- How important are hashtag words, emoticons and emojis as predictors of emotion and sentiment in micro-blogging sites? How can we encode them in a multi-channel convolutional neural network?

- How can we add external features to a CNN model effectively?

The remainder of the paper is organized as follows: We describe our model architecture in detail in Section 2. In Section 3, we describe the datasets and lexicons used in our experiments. As well, we describe the experimental setup required for working with Twitter data. In Section 4, we discuss the results from our experiments. In Section 5, we discuss our findings with particular attention paid to answering the above questions. Finally, in Section 6, we give a summary of our work followed by our remarks on future studies.

2 Multi-channel CNN Model

We represent the architecture of our model in Fig. 1. The model consists of an embedding layer with two channels, a convolution layer with different kernel sizes and multiple filters, a dropout layer for regularization, a max pooling layer, multiple hidden layers and a softmax layer. We now describe each of these layers in detail.

2.1 Embedding Layer

In this layer, two embedding matrices, the Tweet Matrix and the Hash-Emo Matrix, are passed through two different channels of our convolutional neural network. The first matrix represents a particular tweet. Each tweet t_i consists of a sequence of tokens $w_1, w_2, \ldots, w_{n_i}$. L_1 is the maximum tweet length. The height of the Tweet Matrix is L_1. Short tweets are padded using zero padding.

In the Tweet Matrix, every word is represented as a d-dimensional word vector. Since tweets are usually noisy, short in length, and have different kinds of features other than text, it's useful to have a word embedding specially trained on a large amount of Tweet data (Tang et al., 2014). Previous research (Collobert et al., 2011; Socher et al., 2011) has shown the usefulness of using pre-trained word vectors to improve the performance of various models. As a result, in our experiments, we have used the publicly available pre-trained `GloVe` word vectors for Twitter by (Pennington et al., 2014a). The word vectors are trained on 27B word tokens in an unsupervised manner.

In this layer, we also pass another matrix called the Hash-Emo Matrix through a different channel in our network. This matrix is composed of three different sets of features: hashtags, emoticons and emojis. These are considered as distinguishable traits to showcase one's mood (Zhao et al., 2012). People like to use hashtags to express their emotional state through various micro-blogging sites (e.g., Twitter) (Qadir and Riloff, 2014). Also graphical emoticons or emojis can convey strong emotion or sentiment. So for each tweet t_i, we extract hashtags $h_1, h_2, \ldots, h_{k_i}$ and emoticons/emojis $e_1, e_2, \ldots, e_{p_i}$. We concatenate the hashtags and emoticon/emoji vectors for each tweet t_i to get the Hash-Emo Matrix. We introduce a hyper-parameter L_2 as a threshold on the height of the Hash-Emo Matrix. Tweets with the number of hash-emo features less than L_2 are padded with zero while tweets with more hash-emo features than L_2 are truncated. We use word vectors from `GloVe` with dimension d for hashtags words. In the case that no word vector is found for a particular word we randomly initialize it. We also do random initialization of word vectors for emoticons. For emojis, we first map it to something descriptive (to be discussed in more detail in Section 3.2) and then generate random word vectors. These word vectors are tuned during the training phase.

2.2 Convolutional Layer

In this layer, we apply m filters of varying window sizes over the Tweet Matrix from the embedding layer. Here, window size (k) refers to the number of adjacent word vectors in the Tweet Matrix that are filtered together (when $k > 1$). Sliding our filter down we repeat this for the rest of the word vectors. Let $w_i \in \mathbb{R}^d$ be the d-dimensional word vector corresponding to the i-th word in a tweet. Also let $w_{i:i+j}$ denote the concatenation of word vectors $w_i, w_{i+1}, \ldots, w_{i+j}$ and $F \in \mathbb{R}^{k \times d}$ denote the filter matrix. Thus a feature f_i is generated by:

$$f_i = F \otimes w_{i:i+k-1} + b \qquad (1)$$

where b is a bias term and $\otimes$ represents the convolution action (a sum over element-wise multiplications). At this stage, we apply a nonlinear activation function such as *ReLU* (Nair and Hinton, 2010) before passing it through the dropout layer. We use multiple filters with the same window size in order to learn complementary features from the same window. Different window sizes (k) allow us to extract active local k-gram features.

For the Hash-Emo Matrix, we apply m filters to each vector to generate local unigram features in different scales before passing it to the next layer.

2.3 Pooling Layer

In this layer, employing a max-over pooling operation (Collobert et al., 2011) on the output from the previous layer for each channel extracts the most salient features. In this way, for each filter, we get the maximum value. So we get features equal to the number of filters in this stage. We chose max pooling instead of other pooling schemes because Zhang and Wallace (2017) showed that max pooling consistently performs better than other pooling strategies for various sentence classification tasks.

2.4 Hidden Layers

We concatenate all the feature vectors from the previous layer. In addition, we concatenate additional sentiment and affect feature vectors (which are described in detail in Section 3.2) as well which forms a large feature vector. This is then passed through a number of hidden layers. A non-linear activation function (i.e., ReLU (Nair and Hinton, 2010)) is applied in each layer before the vector is finally passed through the output layer. We tried a different activation function (tanh) as well, but ReLU worked the best for us.

2.5 Output Layer

This is a fully connected layer which maps the inputs to a number of outputs corresponding to the number of classes we have. For multi-class classification task, we use softmax as the activation function and categorical cross-entropy as the loss function. The output of the softmax function is equivalent to a categorical probability distribution which generally indicates the probability that any of the classes are true. For binary classification task, we use sigmoid as the activation function and binary cross-entropy as our loss function.

3 Experiments

In this section, we describe in detail the datasets and experimental procedures used in our study.

Emotion	**Dataset**			
	BTD	TEC	CBET	SE
joy	409,983	8,240	10,691	3,011
sadness	351,963	3,830	8,623	2,905
anger	311,851	1,555	9,023	3,091
love	175,077	–	9,398	–
thankfulness	80,291	–	8,544	–
fear	76,580	2,816	9,021	3,627
surprise	14,141	3,849	8,552	–
guilt	–	–	8,540	–
disgust	–	761	8,545	–
Total	1419,886	21,051	80,937	12,634

(a)

Dataset	**#Positive**	**#Negative**	**#Neutral**
STS-Gold	632	1,402	–
STS-Test	182	177	139
SS-Twitter	1,252	1,037	1,953

(b)

Table 1: (a) Basic statistics of the emotion datasets used in our experiments. (b) Basic statistics of sentiment labeled datasets used in our experiments.

3.1 Datasets

We used a number of emotion and sentiment datasets for our experiments. A description of each dataset is given below:

BTD. Big Twitter Data is an emotion-labeled Twitter dataset provided by Wang et al. (2012). The dataset had been automatically annotated based on the seven emotion category seed words (Shaver et al., 1987) being a hashtag and the quality of the data was verified by two annotators as described in (Wang et al., 2012). We were only able to retrieve a portion of the original dataset as many tweets were either removed or not available at the time we fetched the data using the Twitter API. We applied the heuristics from (Wang et al., 2012) to remove any hashtags from the tweets which belong to the list of emotion seed words.

TEC. Twitter Emotion Corpus has been published by Mohammad (2012) for research purposes. About 21,000 tweets were collected based on hashtags corresponding to Ekman's (1999) six basic emotions. The dataset has been used in related works (Shahraki and Zaiane, 2017; Balahur, 2013; Mohammad and Kiritchenko, 2015).

CBET. The Cleaned Balanced Emotional Tweet dataset is provided by Shahraki and Zaiane (2017). To the best of our knowledge, this is one of the largest publically available balanced datasets for twitter emotion detection research. The dataset contains 80,937 tweets with nine emotion categories including Ekman's six basic emotions.

SE. The SemEval-2018 Task 1 - Affect dataset was provided by Mohammad et al. (2018). The SemEval task was to estimate the intensity of a given tweet and its corresponding emotion. However, in this study, we utilize the labeled dataset only to classify the tweets into four emotion categories and use the training, development and test sets provided in this dataset in our experiments.

STS-Gold. This dataset was constructed by Saif et al. (2013) for Twitter sentiment analysis. The dataset contains a total of 2,034 tweets labeled (positive/negative) by three annotators. This dataset has been extensively used in several works for model evaluation (Saif et al., 2014b; Krouska et al., 2017; Saif et al., 2014a).

STS. The Stanford Twitter Sentiment dataset was introduced by Go et al. (2009). It consists of a training set and a test set. The training set contains around 1.6 million tweets, whereas the test set contains 498 tweets. The training set was built automatically based on several emoticons as potential identifiers of sentiment. However, the test set was manually annotated and heavily used for model evaluation in related research. We perform one experiment with all three labels (positive/negative/neutral) to compare the performance of different variants of our model and another one with two labels (positive/negative) to make comparison with related works (Jianqiang et al., 2018; dos Santos and Gatti, 2014; Go et al., 2009).

SS-Twitter. The Sentiment Strength Twitter dataset was constructed by Thelwall et al. (2012) to evaluate SentiStrength. The tweets were manually labeled by multiple persons. Each tweet is assigned a number between 1 and 5 for both positive and negative sentiments, 1 represents weak sentiment strength and 5 represents strong sentiment strength. We followed the heuristics used by Saif et al. (2013) to obtain a single sentiment label for each tweet, giving us a total of $4,242$ positive, negative and neutral tweets. The transformed dataset has been used in other literature (Go et al., 2009; Zhang et al., 2018).

We provide basic statistics of the datasets used in our experiments in Table 1.

3.2 Experimental Setup

Data Cleaning. Twitter data is unstructured and highly informal (Yoon et al., 2013) and thus it requires a great deal of effort to make it suitable for any model. NLTK (Bird and Loper, 2004) provides a regular-expression based tokenizer for Twitter, TweetTokenizer, which preserves user mentions, hashtags, urls, emoticons and emojis in particular. It also reduces the length of repeated characters to three (i.e. ”Haaaaaapy” will become ”Haaapy”)”. In our experiments, we utilized the TweetTokenizer to tokenize tweets.

To accommodate the pretrained word vectors from (Pennington et al., 2014b), we pre-processed each tweet in a number of ways. We lowercased all the letters in the tweet. User mentions have been replaced with <user> token (i.e. @username1 will become <user>). In addition, we also removed urls from the tweets as urls do not provide any emotional value. We also normalized certain negative words (e.g., “won’t” will become “will not”). Using slang words is a very common practice in social media. We compiled a list of the most common slang words from various online resources[2] and replaced all of the occurrences with their full form (e.g., “nvm” will become “never mind”). Our list of slang words doesn’t contain any word which has multiple meanings. Usage of certain punctuation is often crucial in social media posts as it helps the user to emphasize certain things. We found that two punctuation symbols (! and ?) are common among social media users to express certain emotional states. We kept these symbols in our text and normalized the repetitions (e.g., “!!!” will become “! <repeat>”).

The use of emojis and emoticons has increased significantly with the advent of various social media sites. Emoticons (e.g., :-D) are essentially a combination of punctuation marks, letters and numbers used to create pictorial icons which generally display an emotion or sentiment. On the other hand, emojis are pictographs of faces, objects and symbols. The primary purpose of using emojis and emoticons is to convey certain emotions and sentiments (Dresner and Herring, 2010). One advantage of using the TweetTokenizer is that it gives us emoticons and emojis as tokens. Though we use the emoticons as is in our experiment, we utilize a python library called “emoji” to get descriptive details about the pictorial image. For example, “☺” represents *“smiling_face”*.

In our experiments, we removed stop-words and replaced numbers occurring in the tweets with the token <number>. We also stripped off “#” symbols from all the hashtags within the tweets (e.g.,

[2] Example: https://slangit.com/terms/social_media

Emotion	**Dataset**											
	BTD			TEC			CBET			SemEval		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
joy	68.4	77.4	72.6	67.4	77.1	71.8	58.1	56.1	57.1	78.5	70.1	74.1
sadness	72.7	74.5	73.6	48.8	53.7	50.9	38.0	43.3	40.5	62.6	41.0	49.6
anger	74.7	79.1	76.8	34.5	23.8	27.7	49.3	52.1	50.7	59.7	63.6	61.6
love	57.0	46.4	51.1	–	–	–	65.4	53.3	58.7	–	–	–
thankfulness	63.2	55.3	59.0	–	–	–	66.1	68.0	67.0	–	–	–
fear	57.6	38.3	46.0	61.5	57.2	58.6	70.3	69.6	70.0	51.6	71.9	60.1
surprise	88.1	16.1	27.1	55.9	50.2	52.5	51.0	55.3	53.0	–	–	–
guilt	–	–	–	–	–	–	53.8	49.6	51.6	–	–	–
disgust	–	–	–	67.4	77.1	71.8	59.3	61.0	60.2	–	–	–
Avg.	**68.9**	**55.3**	**58.0**	**55.9**	**56.5**	**55.6**	**56.8**	**56.5**	**56.5**	**63.1**	**61.7**	**61.3**

Table 2: Results (in %) of our model (MC-CNN) for four emotion-labeled datasets.

"#depressed" will become "depressed") and used the stripped version of hashtags on the second channel of our model. We only kept tokens with more than one character.

Input Features. Along with word embeddings, we used additional affect and sentiment features in our network. In our experiments, we used a feature vector V_f where each value in the vector corresponds to a particular lexical feature ranging between $[0, 1]$. We utilized a number of publicly available lexicons which are described briefly below to construct the vector.

Warriner et al. (2013) provides a lexicon consisting of 14,000 English lemmas with valence, arousal and dominance scores. Three components of emotion are scored for each word between 1 and 9 in this lexicon. We calculate the average score for each component across all tokens in a tweet and normalize them in the range [0, 1]. Gilbert (2014) provides a list of lexical features along with their associated sentiment intensity measures. We use this lexicon to calculate the average of positive, negative, and neutral scores over all the tokens in a tweet. In addition, we used the *NRC Emotion Lexicon* provided by Mohammad and Turney (2013) which consists of a list of unigrams and their association with one of the emotion categories (anger, anticipation, disgust, fear, joy, sadness, surprise, trust). We use the percentage of tokens belonging to each emotion category as features. We also used the *NRC Affect Intensity Lexicon* provided by Mohammad and Bravo-Marquez (2017) and *NRC Hashtag Emotion Lexicon* provided by Mohammad and Kiritchenko (2015) which contain real-valued fine-grained word-emotion association scores for words and hashtag words.

We combined two lexicons *MPQA* and *BingLiu* provided by Wilson et al. (2005) and Hu and Liu (2004), respectively, and used them to calculate the percentage of positive and negative tokens belonging to each tweet. We also used *AFINN* (Nielsen, 2011) which contains a list of English words rated for valence with an integer between −5 (negative) to +5 (positive). We first normalized the scores in the range [0,1] and then calculated the average of this score over all the tokens in a tweet. Lastly, we detect the presence of consecutive exclamation (!) and question marks (?) in a tweet and use them as boolean features.

Network Parameters and Training. Zhang and Wallace (2017) performed a sensitivity analysis on various parameters of a one-layer CNN model and showed how tuning the parameters can affect the performance of a model. Inspired by the work done by (Zhang and Wallace, 2017), we also searched for the optimal parameter configurations in our network. Table 3 shows different hyper-parameter configurations that we tried and the final configuration that was used in our model. The final configuration was based on both performance and training time. The embedding dimension has been set to 100 for both channels of our network as it worked best for us among other dimensions. We also experimented with a different number of fil-

Hyper-parameter	**Ranges**	**Selected**
Embedding dimension	50/100/200	100
Number of filters	64/128/256	128
Kernel sizes	1/2/3/4/5	1/2/3
Batch size	16/30/50	16
Epochs	10/20	10
Dropout rate	0.1/0.2/0.5	0.5
Learning rate	0.015/0.001/0.01	0.001

Table 3: Ranges of different hyper-parameters searched during tuning and the final configurations selected for our experiments

Datasets	Methods	Positive			Negative			Average			Accuracy
		P	R	F1	P	R	F1	P	R	F1	
STS-Gold	A	70.5	74.1	72.2	88.0	86.0	87.0	79.3	88.0	79.6	82.3
	B	–	–	–	–	–	–	79.5	77.9	78.6	82.1
	C	–	–	–	–	–	–	–	–	77.5	80.3
	D	75.4	74.9	75.1	90.2	90.3	90.2	82.8	82.6	82.7	86.0
	Ours	87.9	82.0	84.5	92.1	94.6	93.3	90.0	88.3	88.9	**90.7**
STS-Test	D	88.0	89.5	88.7	87.2	85.4	86.3	87.6	87.4	87.5	87.6
	E	–	–	–	–	–	–	–	–	–	86.4
	F	–	–	–	–	–	–	–	–	–	83.0
	Ours	90.2	91.2	90.5	91.3	89.3	89.9	90.8	90.3	90.2	**90.3**
SS-Twitter	G	–	–	–	–	–	–	67.8	52.7	59.3	61.9
	F	–	–	76.6	–	–	69.2	–	–	72.9	73.4
	Ours	81.3	84.7	82.0	72.5	72.7	72.2	76.9	78.7	77.1	**79.3**

Table 4: Results (in %) of our model (MC-CNN) from 10-fold cross-validation compared against other methods for sentiment labeled datasets (2-class). Bold text indicates the best performance in a column. **A:** Thelwall-Lexicon (Updated + Expanded) (Saif et al., 2014b). **B:** SentiStrength (Krouska et al., 2017). **C:** SentiCircle with Pivot (Saif et al., 2014a). **D:** Deep Convolutional Neural Network (Jianqiang et al., 2018). **E:** Character to Sentence Convolutional Neural Network (CharSCNN) (dos Santos and Gatti, 2014). **F:** Maximum Entropy (Saif et al., 2013). **G:** Quantum Language Model + Quantum Relative Entropy (Zhang et al., 2018).

ters and varying kernel sizes. The combination of kernel sizes, $(k = 1, 2, 3)$ in the first channel and $k = 1$ in the second channel worked the best for us. We also experimented with various batch sizes and the performance of the model remained reasonably constant, though the training time varied significantly. In our network, we used three hidden layers. In addition, we used the Adam optimizer (Kingma and Ba, 2014) and the back-propagation (Rumelhart et al., 1986) algorithm for training our model. Keras 2.2.0 was used for implementing the model.

Regularization. In order to reduce overfitting, it is a common practice to employ regularization strategies in CNNs. In our experiments, we used dropout regularization (Srivastava et al., 2014) for both of the channels after the convolutional layer. We experimented with three different dropout rates as seen in Table 3 and also with no dropout at all. The model works better when we apply dropouts after the convolutional layer.

4 Results

In this section, we describe the results obtained in our experiments. We use precision, recall, F1-score and accuracy as our evaluation metrics.

In recent emotion category recognition studies on Twitter data, people tend to construct their own dataset by collecting tweets from Twitter for their experiments. Hence, it is hard to find a large enough benchmark dataset to compare the performance with other people's work. In this study, we experimented with four emotion labeled datasets which have been made publicly available by their authors. Table 2 shows the results for each emotion category for all of the datasets. For the BTD dataset, we trained our model with $1,136,305$ tweets, while we used $140,979$ and $142,602$ tweets as development and test data respectively. We used the same training, development and test sets as (Wang et al., 2012) except that our retrieved dataset contains fewer samples. We achieved relatively high F1-scores of $\mathbf{72.6\%}, \mathbf{73.6\%}$ and $\mathbf{76.8\%}$ for *joy, sadness* and *anger*, respectively, whereas for *surprise* we get a low F1-score of $\mathbf{27.1\%}$. This is probably due to the imbalanced nature of the dataset as can be seen in Table 1. The number of samples for *joy, sadness* and *anger* is much higher than for *surprise*. Our model achieves an accuracy of $\mathbf{69.2\%}$, whereas Wang et al. (2012) reported an accuracy of $\mathbf{65.6\%}$ when trained on a much larger dataset. We can not make direct comparison with (Wang et al., 2012) since we were not able to retrieve the full test set due to the unavailability of some tweets at the time of fetching data from Twitter. For the TEC dataset, we evaluated our model with 10-fold cross validation. Mohammad (2012) reported an F1-score of $\mathbf{49.9\%}$ with SVM, whereas our model achieves an F1-score of $\mathbf{55.6\%}$. For the CBET dataset, we used 80% of the data as the training set and the remaining 20% as the test set. We get an average F1-score of $\mathbf{56.5\%}$. We also used 10-fold cross-validation for the SemEval dataset and achieved an F1-score of $\mathbf{61.3\%}$. Table 4 shows the performance of our model with 10-fold cross-

Datasets	Methods	Positive			Negative			Neutral			Accuracy
		P	R	F1	P	R	F1	P	R	F1	
STS-Test	CNN	73.6	84.1	76.8	73.5	74.2	72.8	74.2	64.3	65.8	75.1
	MC-CNN†	63.1	83.4	70.3	76.5	70.4	72.8	71.6	53.9	60.8	70.6
	MC-CNN†‡	**80.3**	**83.8**	**81.4**	**87.5**	**81.2**	**83.5**	**79.2**	**77.9**	**77.7**	**81.5**
SS-Twitter	CNN	48.9	51.7	49.3	43.9	53.7	47.4	67.8	**66.8**	64.3	59.1
	MC-CNN†	61.6	62.0	61.4	55.3	65.5	59.7	71.6	62.7	66.4	63.2
	MC-CNN†‡	**65.1**	**65.4**	**64.0**	**56.2**	**65.7**	**60.0**	**72.2**	62.7	**66.7**	**64.6**

Table 5: Results (in %) of three variants of our model from 10-fold cross-validation for sentiment labeled datasets (3-class). Bold text indicates the best performance in a column.† represents the inclusion of Hash-Emo embedding into the network. ‡ represents the inclusion of external features into the network.

validation on different sentiment datasets with two classes (positive and negative). For the STS-Gold dataset, our model achieves an accuracy of **90.7%** whereas the previous best accuracy (**86.0%**) was reported by Jianqiang et al. (2018) with a deep CNN model. Our model achieves the best accuracy (**90.3%**) for the STS-Test dataset as well, while the previous best (**87.6%**) was reported in (Jianqiang et al., 2018). dos Santos and Gatti (2014) also experimented with the same dataset with their Character to Sentence CNN model, reporting an accuracy of **86.4%**. Lastly, for the SS-Twitter dataset, our model achieves an accuracy of **79.3%** whereas Zhang et al. (2018) and Saif et al. (2013) reported an accuracy of **61.9%** and **73.4%**, respectively.

Tables 5 and 6 show the performance of three variants of our model on the sentiment labeled datasets and the emotion labeled datasets respectively. The first variant is a basic CNN model without hash-emo embedding or any additional features. The second variant includes the hash-emo embedding, while the last variant combines additional lexical features as well. It can be observed that when we introduce the second channel with hash-emo embedding, we get a significant increase in accuracy for most of the datasets. We can see in Table 5 that, for STS-Test and SS-Twitter datasets, we get better F1-scores for all three sentiment labels when we include the hash-emo embedding along with external lexical features. In Table 6, we can see that, inclusion of hash-emo embedding in the network gives us **2.4**, **3.3**, **2.3** and **3.5** percentage points increase in accuracy and the inclusion of additional features as well gives us **3.1**, **4.6**, **2.6** and **5.7** percentage points increase in accuracy for BTD, TEC, CBET and SE datasets, respectively, over the base models.

Models	Dataset			
	BTD	*TEC*	*CBET*	*SE*
CNN	66.1	54.3	53.8	56.3
MC-CNN†	68.5	57.6	56.1	59.8
MC-CNN†‡	69.2	58.9	56.4	62.0

Table 6: Comparison of results (accuracy in %) of three variants of our model. † represents the inclusion of Hash-Emo embedding into the network. ‡ represents the inclusion of external features into the network.

5 Discussion

In this study, we have showed the effectiveness of encoding hashtags, emoticons and emojis through a separate channel in a CNN network for emotion and sentiment detection tasks. Our MC-CNN model with hash-emo embedding performs well when compared to the basic CNN model. To the best of our knowledge, our model achieves the best accuracies on the three sentiment datasets, and has significant improvement in performance on the four emotion labeled datasets over the basic CNN model. The results show the importance of hashtags, emoticons and emojis in social media as predictors of emotion and sentiment. The model performs even better when additional lexical features are introduced into the network.

6 Conclusions and Future Work

In this paper, we propose a novel use of a multi-channel convolutional neural architecture which effectively encodes different types of emotion indicators which are found in social media posts. Results suggest that encoding the emotion indicators through a separate channel provides significant improvement over the traditional CNN based models. We also demonstrate a simple approach to incorporate different lexical features in the network giving us comparatively better results when used along with our MC-CNN model. Our model performs particularly well on two important tasks in social media: emotion detection and sentiment analysis. This model can be extended to perform other tasks as well. In future, we would like to explore character embedding as this can give us crucial linguistic features from noisy twitter data.

Acknowledgements

We are thankful for the support by The Social Sciences and Humanities Research Council of Canada (SSHRC) through our Partnership Development Grant titled “Living Archives of Rwandan Exiles and Survivors in Canada”.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 718–728.

Saima Aman and Stan Szpakowicz. 2008. Using roget's thesaurus for fine-grained emotion recognition. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Alexandra Balahur. 2013. Sentiment analysis in social media texts. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 120–128.

Anil Bandhakavi, Nirmalie Wiratunga, Deepak Padmanabhan, and Stewart Massie. 2017. Lexicon based feature extraction for emotion text classification. *Pattern recognition letters*, 93:133–142.

Steven Bird and Edward Loper. 2004. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Bart Desmet and VéRonique Hoste. 2013. Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351–6358.

Eli Dresner and Susan C Herring. 2010. Functions of the nonverbal in cmc: Emoticons and illocutionary force. *Communication theory*, 20(3):249–268.

Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion*, pages 45–60.

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

CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12).

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Zhao Jianqiang, Gui Xiaolin, and Zhang Xuejun. 2018. Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, 6:23253–23260.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Lei Ba. 2014. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*.

Akrivi Krouska, Christos Troussas, and Maria Virvou. 2017. Comparative evaluation of algorithms for sentiment analysis over social networking services. *J Univers Comput Sci*, 23(8):755–768.

Saif Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 65–77. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17.

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*, pages 246–255. Association for Computational Linguistics.

Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Saif M Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. 2015. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499.

Myriam D Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. 2014. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing*, 5(2):101–111.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2007. Textual affect sensing for sociable and expressive online communication. In *International Conference on Affective Computing and Intelligent Interaction*, pages 218–229. Springer.

Finn Årup Nielsen. 2011. A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, Heraklion, Crete, Greece, May 30, 2011*, pages 93–98.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014a. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

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

Robert Plutchik. 1984. Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219.

Ashequl Qadir and Ellen Riloff. 2014. Learning emotion indicators from tweets: Hashtags, hashtag patterns, and phrases. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1203–1209. Association for Computational Linguistics.

Fuji Ren and Changqin Quan. 2012. Linguistic-based emotion analysis and recognition for measuring consumer satisfaction: an application of affective computing. *Information Technology and Management*, 13(4):321–332.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533.

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. *first ESSEM workshop*.

Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. 2014a. Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In *European Semantic Web Conference*, pages 83–98. Springer.

Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. 2014b. Adapting sentiment lexicons using contextual semantics for sentiment analysis of twitter. In *European Semantic Web Conference*, pages 54–63. Springer.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

Ameneh Gholipour Shahraki and Osmar R Zaiane. 2017. Lexical and learning-based emotion mining from text. In *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing*.

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.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics.

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

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566. Association for Computational Linguistics.

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), pages 1555–1565. Association for Computational Linguistics.

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.

Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. 2016. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 225–230.

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.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Sunmoo Yoon, Noémie Elhadad, and Suzanne Bakken. 2013. A practical approach for content mining of tweets. *American journal of preventive medicine*, 45(1):122–129.

Yazhou Zhang, Dawei Song, Xiang Li, and Peng Zhang. 2018. Unsupervised sentiment analysis of twitter posts using density matrix representation. In *European Conference on Information Retrieval*, pages 316–329. Springer.

Ye Zhang and Byron Wallace. 2017. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263. Asian Federation of Natural Language Processing.

Jichang Zhao, Li Dong, Junjie Wu, and Ke Xu. 2012. Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1528–1531. ACM.