Multi-Sentiment Modeling with Scalable Systematic Labeled Data Generation via Word2Vec Clustering

Dhruv Mayank
Data Science Labs
Sysomos LP
Toronto, ON M5H 2R2
Email: dhruv.mayank@gmail.com

Kanchana Padmanabhan
Data Science Labs
Sysomos LP
Toronto, ON M5H 2R2
Email: kanchana.padmanabhan@gmail.com

Koushik Pal
Data Science Labs
Sysomos LP
Toronto, ON M5H 2R2
Email: koushik.pal@gmail.com

***Abstract*—Social networks are now a primary source for news and opinions on topics ranging from sports to politics. Analyzing opinions with an associated sentiment is crucial to the success of any campaign (product, marketing, or political). However, there are two significant challenges that need to be overcome. First, social networks produce large volumes of data at high velocities. Using traditional (semi-) manual methods to gather training data is, therefore, impractical and expensive. Second, humans express more than two emotions, therefore, the typical binary good/bad or positive/negative classifiers are no longer sufficient to address the complex needs of the social marketing domain. This paper introduces a hugely scalable approach to gathering training data by using emojis as proxy for user sentiments. This paper also introduces a systematic Word2Vec based clustering method to generate emoji clusters that arguably represent different human emotions (multi-sentiment). Finally, this paper also introduces a threshold-based formulation to predicting one or two class labels (multi-label) for a given document. Our scalable multi-sentiment multi-label model produces a cross-validation accuracy of 71.55% (±0.22%). To compare against other models in the literature, we also trained a binary (positive vs. negative) classifier. It produces a cross-validation accuracy of 84.95% (±0.17%), which is arguably better than several results reported in literature thus far.**

***Keywords*—Sentiment Analysis, Multi-sentiment, Multi-label, Emoji, Word2Vec Clustering.**

I. Introduction

Social networks, such as Twitter, Facebook and Tumblr to name a few, have become the primary opinion and information sharing platforms among billions of Internet users. People are keen to post opinions about a variety of topics such as products, movies, music, politics, and current affairs. We are at an interesting phase where social network engagement (people posting or sharing about a specific topic) has become a significant measure of success for a product, a movie, or even political candidacy. Of course, volume of engagement alone is insufficient to judge success. The measure of success is deeply coupled with the sentiment of a particular topic. Measure of sentiment often affects how a marketer, a celebrity, or a political party reacts to a situation. Below is an example of a social media post with a negative tone about a company, and a reply from the concerned company 5 minutes later. The authors of the tweets have been made anonymous.

- **Tweet (3:17 PM):** *Booked a full-size car @XYZ, as Gold member; too bad, no more, pick a small car. Then they don't reduce the price. #RipOff*
- **Reply (3:22 PM):** *@XXXX Hi XX, We're sorry to hear that. Please DM us your rental info. We'd like to look into this.*

Social networks have an implicit need for good sentiment models. However, social networks produce large volumes of data at high velocities — *big data revolution*. For example, Twitter produces approx. 350K tweets per minute [1]. This means that any machine learning model built for social network data needs to take advantage of the scale.

A model built for sentiment classification is typically trained on a relatively small (~ 1000) dataset that is (semi-) manually tagged. This requires humans to read a text, understand the sentiment, and tag it into the relevant class. Social media text in general are shorter, casual, and typically not well constructed (in comparison to reviews on other websites such as Amazon, Yelp or IMDB). The short and ill-constructed social media posts make it difficult for humans (who are not the content authors) to arrive at the right label. Moreover, the scale of social media makes this process impractical and expensive. Also, as more sentiment class labels are required, we need more ground truth data for training.

We propose to use emojis as sentiment class labels to obtain massive amounts (e.g., 38.1M gathered for our experiments) of training data with little-to-no manual intervention. Social networks, and other messaging platforms, allow a user to express emotions through special characters called emojis. Emojis help unify and understand emotions across a variety of writing styles; e.g, anger expressed in American English versus anger expressed in British English. This is similar to the approach by *Pang et al. [2], [3]* that uses star ratings as polarity signals for movie reviews, and by *Read [4]* and *Go et al. [5]* that uses tweets with emoticons as labels. The automatic massive training data generation pipeline can help us move towards more complex models. For example, deep learning models can extract much more generalized features, but they need to feed on a large dataset to be able to do that.

If volume and velocity are two important aspects of social media, variety is another. Emojis allow people to express a variety of basic emotions (happy, sad, anger) along with different degrees of those emotions (mad with rage vs. disappointed) [6]. This mimics how humans express emotions. For example, [7] lists six emotion classes with 42 different degrees of emotion. Healey and Ramaswamy [8] have developed a twitter sentiment vizualization based on Russell model of eight emotional effects [9] that uses ANEW (Affective Norms for English Words) [10]. Hence, a binary good/bad classification [11], [12], [13], [14], [5] is no longer sufficient to address the complex needs of the social marketing domain. We need a multi-sentiment model to predict different human emotions. Additionally, the same text may express more than one emotion and that requires a multi-label prediction model as well. A multi-sentiment multi-label model will enable a marketer to make more nuanced and targeted messages for a successful campaign.

Emojis help us naturally move away from the simple binary sentiment classification to a multi-sentiment model. Our initial experiment with a model constructed using ~ 49 emojis as class labels yielded an accuracy of $< 10\%$. This is because emojis are messy (veracity) and often incorrectly used, thereby requiring significant preprocessing to make them usable (value).

In this paper, we describe a scalable methodology for generating training data for multi-sentiment models, using emojis as proxy for user sentiments. This paper introduces a systematic approach to obtaining sentiment class labels using Word2Vec [15] based emoji clustering. Arguably, the emoji clusters generated are representatives of different human emotions such as love, angry and sad. This paper also introduces a new threshold-based formulation to predicting the top one or the top two sentiment labels for a given document. As stated earlier, emojis are messy. So, in Section II-B, we explicitly detail several issues that occur while using emojis (e.g., emojis that look similar but convey entirely different meanings) and possible solutions to the issues. We use data gathered from Twitter fire hose to demonstrate the validity of the methodology.

Our multi-sentiment multi-label model with 6 different sentiment classes produces a 10-fold cross-validation accuracy of 71.55% ($\pm$0.22%) (cf. Table XII). Our binary (positive vs. negative) classifier produces an accuracy of 84.95% ($\pm$0.17%) (cf. Table VIII) which is better than 82.2% and 82.9% obtained using the methodologies of *Go et al. [5]* and *Pang et al. [2], [3]*, respectively.

Finally, we believe that this approach to generating labeled data is generalizable beyond just sentiment analysis. Any set of keywords that reliably has a meaning can be used as labels, and in particular, they can be clustered with a Word2Vec model. This allows scalable label generation for any problem in which these keywords can be identified.

II. Methodology

A. Problem Definition

Given a set S of tweets and a set E of emojis that convey some sentiment, a set $T = \{(s,\ e) \mid s \in S,\ e \in E\}$ is generated using tweets that have (single) emojis. The emojis act as the sentiment labels for the tweets and hence a many-to-one relationship exists between S and E.

The goal is to train a classifier model using data T so that tweets with no emojis (or non-sentiment emojis) can be assigned a sentiment. The emojis convey several different sentiments such as happy, sad, angry and love. Thus, the problem moves beyond the typical positive/negative binary classification to the multi-sentiment domain. Moreover, using emojis as class labels relieves one from the need for any manual tagging of training data.

B. Emoji Selection

The first and the most important step in the process is the selection of emojis that can act as class labels and good representatives of several human sentiments.

The original data we collected consisted of 49 emojis using 38.1 M tweets (cf. Section II-D for more details). An initial inspection of the data brought to light several problems, the biggest of which is that several emojis are being used in unexpected ways.

The worst offenders are 😁 and 😬. Looking at the Twitter representation of these emojis, the similarity is evident. The issue, however, is that the first is meant to convey a grin, while the second is meant to convey a grimace. Unfortunately, because of their similarity, they are often used in place of each other. We confirmed the same using Word2Vec where 😁 is most similar to 😬. The following examples illustrate this:

- 😁 used as 😬 : *In the process of working on one project I have created about four more for myself* 😁
- 😬 used as 😁 : *this just made me even more excited to see your face* 😬

Another emoji that causes a problem is 😅. We found several tweets where this emoji is used as if the sweat is a tear and several more where it is used as just a smiley face. Here are examples of the usage we expected, as well as each of the unexpected ones.

- Expected : *Day 5 of being deathly ill in bed: starting to have conversations with people in my head to pass time* 😅
- Negative : *just thinking ab work tomorrow is making me nervous* 😅
- Positive : *i'm ready for football season* 😅

The emoji 😓 also has sweat which can be mistaken for a tear, though this is less of a problem because it already conveys a negative sentiment. That being said, we still removed it from the data because of the two interpretations (sad vs. disappointed) in which it is being used. There are other emojis with multiple meanings such as this. The emoji 😐 is used by some as a completely neutral emoji, while others use it to convey annoyance, similar to 😑. The emojis 😫 and 😩 are very similar, and they both are used in several situations. Some people use them to convey anger,

while others use them to convey sadness, and some even use them to convey extreme happiness.

- 😐 used neutrally : *They're trying to keep a straight face* 😐 (in reference to this)
- 😐 used to mean annoyed : *Don't even get me started with this topic* 😐
- 😩 used positively : *These PROMposals are so freaking cute!* 😩 😩

In addition to removing emojis that had conflicting usage, we also removed some emojis that were not frequent enough in our dataset. This included the cat emojis, for example. Figure 1 demonstrates the frequency counts of these 49 emojis in our dataset. Any emoji with a frequency count of less than 70K was automatically ignored.

The above processing steps resulted in a set of 37 emojis. Instead of using these raw emojis as class labels, we clustered them into a few sentiment classes. This is because several emojis often convey a similar sentiment with varying degree of the sentiment. For instance, sadness is conveyed via the emojis 😟 😕 😞 😖 😔 😪 and 😢.

C. Word2Vec Clustering

We took a systematic approach to clustering the emojis together into a few sentiment classes. We used an in-house Word2Vec [15] model which was trained on $42.3M$ tweets and a vocabulary of size $\sim$ $250K$ that includes all of the emojis. Using the vectors of the pertinent emojis, we clustered them with agglomerative clustering. We started with 10 clusters because we were hoping to get 10 emotions as output. The resulting clusters are described in Table I.

Sentiment	emojis
love	😙 😊 😚 😍 😘 💕
good	👍 👏 👌 🙌
angry	😠 😡 😤 😑 😒 😴
joking	😉 😜 😏
silly	😋 😌 😛
smileys	😆 😃 😀 😄
sad	😟 😕 😞 😖 😔 😪 😢
like	😚 ❤
funny	😂
cool	😎

TABLE I: Clustering of 37 emojis into 10 sentiments

After experimenting with these clusters, we felt that there are still some emojis left that are not well defined enough in terms of sentiments, specifically those in the clusters with low f-scores, namely, cool, joking, silly, love, and smileys, as Table II demonstrates. The details of the classification model can be found in Section III-B.

Sentiment	Precision	Recall	F-score
angry	0.33	0.51	0.40
cool	0.32	0.34	0.33
joking	0.26	0.19	0.22
silly	0.31	0.26	0.28
funny	0.34	0.39	0.37
good	0.70	0.46	0.56
love	0.32	0.28	0.30
like	0.58	0.66	0.62
sad	0.40	0.45	0.42
smileys	0.34	0.32	0.33
Average	0.39	0.38	0.38

TABLE II: Precision, recall and f-scores for 10-class classification

Sentiment	emojis
love	😙 😊 😚 😍 😘 💕
good	👍 👏 👌 🙌
angry	😠 😡 😤 😑 😒 😴
sad	😟 😕 😞 😖 😔 😪 😢
like	😚 ❤
funny	😂

TABLE III: Clustering of 26 emojis into 6 sentiments

We removed all clusters with low f-scores, except love. We kept love because it contains emojis that are extremely widely used, with multiple emojis that have been used in over $1M$ tweets. With the remaining emojis, we re-ran agglomerative clustering to make sure that the clusters remained the same given the new emoji set, and we found that they did. This finally left us with 26 emojis defining 6 sentiment classes. Table III gives the breakdown, and Table IV shows the precision, recall and f-scores for these 6 classes. The details of the classification model can be found in Section III-C.

To compare our results with other work being done in the literature, we also ran a 2-class classification by clustering the emojis into a positive and a negative class. We, however, excluded the funny class as it is used in both a positive and a negative connotation. We then re-ran agglomerative clustering for the remaining emojis, looking for two clusters. Table V shows the breakdown of these 2 classes. The clustering naturally separates emojis with positive sentiments from emojis with negative sentiments. The angry and the sad class merge into one cluster, while the three positive classes merge into another. Table VI details the precision, recall and f-scores for the two clusters. The details of the classification model can be found in Section III-A.

D. Data Preprocessing

The raw data collected from Twitter consisted of all English text tweets (excluding retweets) between April 1 and April 7 of 2016, for a total of $38.1M$ tweets. All tweets

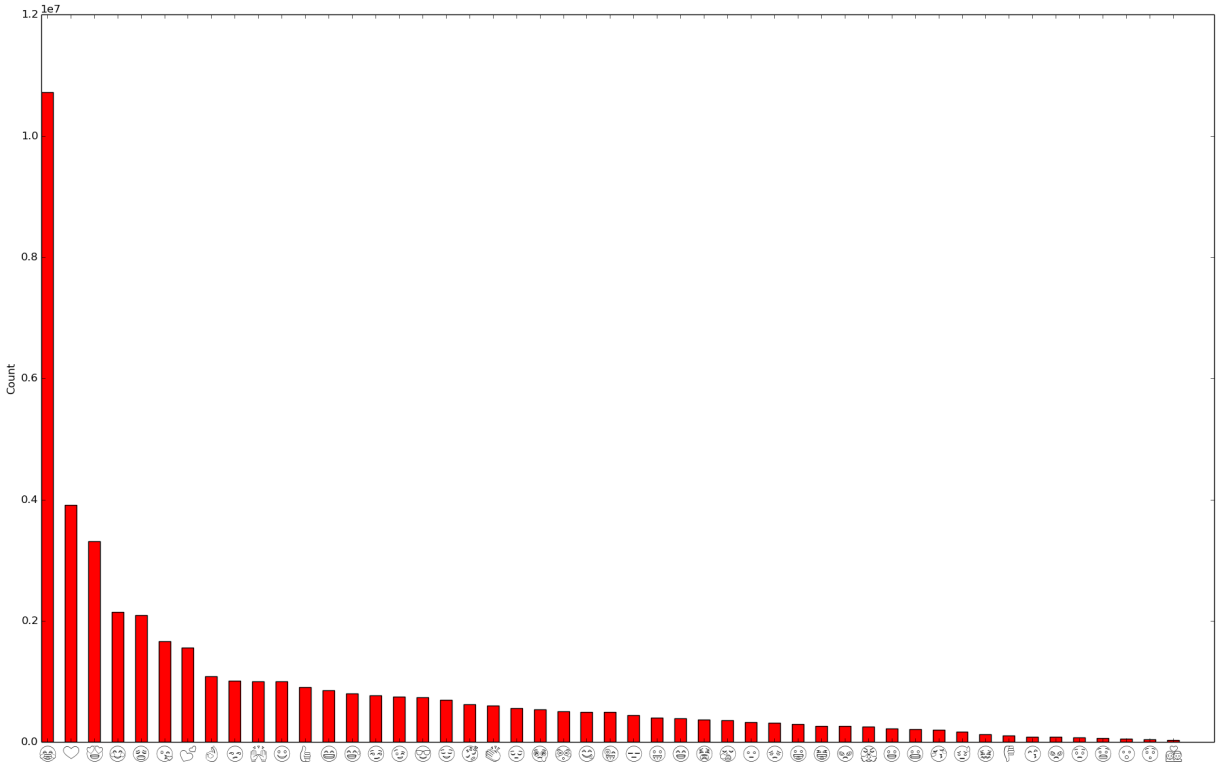

Fig. 1: Emoji Distribution

Sentiment	Precision	Recall	F-score
angry	0.43	0.52	0.47
funny	0.47	0.52	0.50
good	0.63	0.58	0.60
love	0.53	0.46	0.49
like	0.74	0.64	0.69
sad	0.49	0.50	0.50
Average	0.55	0.54	0.54

TABLE IV: Precision, recall and f-scores for 6-class classification

Sentiment	emojis
positive	
negative	

TABLE V: Clustering of 25 emojis into 2 sentiments

had to contain an emoji from the list of relevant emojis (cf. Section II-B). Tweets that contained more than one emoji from the list of relevant emojis were removed. To regularize the data, we removed the characters !?., " from the tweets. The entire text was converted to lowercase, and split by whitespace. We removed all urls and media

Sentiment	Precision	Recall	F-score
negative	0.86	0.84	0.85
positive	0.84	0.86	0.85
Average	0.85	0.85	0.85

TABLE VI: Precision, recall and f-scores for 2-class classification

urls, and replaced them with the keyword *URL*. We also removed all usernames and hashtags, stripped the @ and # from them, and added them back in. The reason for doing this was that there are times where these can be attached to other text or characters without a space separating them.

We then assigned sentiments based on the emojis (cf. Section II-C), and took a random sample of the data such that each sentiment had 100K tweets. Any emoji that fitted multiple sentiments were removed from the dataset to avoid confusion. The next step was to create a collection of all of the words in the dataset along with their frequency counts. We did this to be able to exclude infrequent words (words that occurred less than 15 times in the entire dataset). We used all of the remaining words as features. Table VII shows the number of features before and after removing the infrequent ones for each of the 10-

No. of classes	No. of features before	No. of features after
10	725556	21020
6	622307	16269
2	214682	16017

TABLE VII: Number of features before and after pruning in the 10-class, 6-class, and 2-class classification problems

class, 6-class and 2-class classification problems.

E. TFIDF

Term frequency inverse document frequency (TFIDF) [16] is an effective way to narrow down on the relevant features. Let D be the corpus of tweets and d be a tweet in the corpus. For a given word t in d,

$$\text{TFIDF}(t, D) = f_{t,d} * \log\Big(\frac{N}{|\{d \in D : t \in d\}|}\Big),$$

where $f_{t,d}$ is the frequency of the word t in the tweet d, and N is the number of tweets in the corpus. In this formula, $f_{t,d}$ is the term frequency, while the rest is the inverse document frequency. The inverse document frequency decreases logarithmically as the number of tweets that a word appears in approaches N (the total number of tweets). This means that very common words, such as 'I', 'to', 'you' and 'the', are devalued because they occur in the largest percentage of the documents and, therefore, do not convey any significant information about the documents they occur in, while the rarer words are given greater importance and rightly so. In each of our classifiers, we computed the TFIDF scores of each of the features for each of the training documents, and passed those scores as inputs to our models.

F. Model Selection

The classification models we chose are reflective of the two classification tasks at hand: (1) multi-sentiment multi-label classification, and (2) binary positive-negative classification. SVM was chosen as one of the models because it is a robust binary classifier. Multinomial Naive Bayes (MNB) was chosen as (1) it is a multi-class classifier, (2) it produces probabilities that can be used in the Top 2 selection (cf. Section II-G), and (3) it has been previously shown to be good for text classification tasks [17]. Section III details the results obtained using both classifiers. SVM was used in the one-vs-all mode when training for the multi-label sentiment task.

G. Top 2 Selection

An issue we ran into while making predictions for a given tweet is that several tweets arguably had multiple sentiments. The following tweet is an example in which the author is upset but finds the situation funny as well:

> *Messaged my older sister that I was pregnant (April Fools) and the stupid girl told my mum. Now mum's incredibly upset w/ me* 😢😩😂

Hence, it is reasonable to make multiple predictions for several tweets. Since our best model has only 6 classes, it is excessive to predict 3 or more classes for a given input. So, we decided to return the labels with the top two probabilities provided they are "close". The precise condition we formulated for returning labels with the top two probabilities is

$$\psi := \frac{p_2}{p_1 + p_2} > \delta,$$

where p_i is the probability that the ith result (ordered from highest probability to lowest) is correct and δ is the threshold above which top two labels are returned instead of the top one label. The quantity ψ ranges from 0 (meaning that the classifier is certain about the label with highest probability) to 0.5 (meaning that the classifier finds the labels with the first and the second highest probabilities equally valid).

In order to choose a good threshold δ, we varied the value of δ between 0.5 and 0 at 0.1 intervals and plotted the corresponding accuracy of the model (cf. Figure 2). We see that the accuracy increases as we move closer to 0 and decreases as we move closer to 0.5, as expected. The graph shows an elbow at the point 0.3, where the gains from decreasing it further were marginal compared to the gains from decreasing it up to this point. Hence, we chose 0.3 as value for δ in all our experiments. We didn't choose $\delta = 0$ because top 2 sentiments does not apply to every tweet. If the assigned sentiment was in either of the two predicted results, the tweet was marked as successfully predicted.

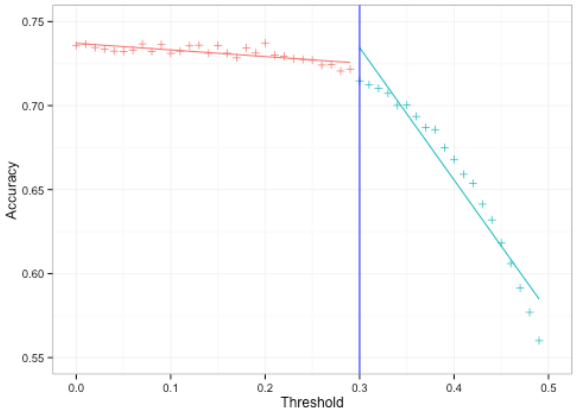

Fig. 2: Model accuracy vs. threshold for selection of δ

III. Experiments & Discussion

All results shown in this section are for 10-fold cross-validation unless specified otherwise. We say "top 1 selection" to refer to always choosing the best class label, and "top 2 selection" to refer to choosing either the best or the top two class labels using the process described in Section II-G. The data collection, prepossessing and feature generation procedures are described in Section II-D.

A. Two Sentiments Classification Results

In this section, we present the results for our positive-negative binary sentiment classifier. Recall that the two

classes are generated using $n = 2$ in the agglomerative clustering (cf. Table V for the emojis in the two clusters). These classifiers naturally use top 1 selection because there are only two possible labels. Table VIII show the results for the binary classifier using a Naive Bayes (NB) and Support Vector Machine (SVM), respectively. The best model is SVM with an accuracy of **84.95**% (±**0.17**%) and an F-score of **0.85** (cf. Table VI). Accuracy from the Naive Bayes model is only marginally lower at 82.75% (±0.26%). Details of the results can be found in Tables IX and X. Our SVM classifier has a significantly better accuracy (+2.75%) than the SVM classifier from *Go et. al, [5]* (cf. Table VIII). Our SVM model also has a better accuracy (+2.05%) in comparison to *Pang et. al [2], [3]* who classified movie reviews into two sentiments (as reported in [5]). Table VIII details these comparisons along with the best results from *Barbosa et al [14]*, *Agarwal et al [11]*, and *Liu et al [18]* (as reported in [12]) that use twitter data.

B. Ten Sentiments Classification Results

In this section, we present the results for our ten sentiments classifier. Recall that the ten classes are generated using $n = 10$ in the agglomerative clustering (cf. Table I for the emojis in the ten clusters). Table XI shows the results of using the Multinomial Naive Bayes classifier with top 2 selection. The average overall accuracy is **54.42**% (±**0.15**%). Only two of the ten classes have $\sim 70\%$ average accuracy. One of them is **like**, which also performs well in the six sentiments classification (cf. Section III-C). The **cool**, **joking**, **silly**, **smileys** and **love** clusters have less than 50% average accuracy. Additionally, these four classes have the lowest recall and precision (cf. Table II). In Section II-B, we have discussed these four classes and provided reasons for removing four of these five classes from our final model.

C. Six Sentiments Classification Results

In this section, we present the results for the six sentiments classifier. Recall that the six classes are generated using $n = 6$ in the agglomerative clustering (cf. Table III for the emojis in the six clusters). Table XII displays the results of using our best model — Multinomial Naive Bayes classifier with top 2 selection. The average overall accuracy is **71.55**% (±**0.22**%), which is 17.13% more than the ten class model discussed in the previous section. Five of the six classes have an average accuracy of $\sim 70\%$. The **love** class has the least accuracy of 63%. However, as discussed in Section II-B, love is one of the poorly performing clusters and is only included due to its abundant usage. The **like** class has the best precision and recall (cf. Table IV). The **like** class is an interesting class. It predates Twitter (and most social media platforms), and when agglomerative clustering is run for several n from $n = 4$ to $n = 16$, this class appears as its own class for every choice of n. This shows that over the years people have developed a specific use for the two emojis in this class.

These results can also be compared to Table XII, which shows the results of using Multinomial Naive Bayes classifier with top 1 selection. **Using the top 2 selection results in a gain of nearly 18% in the average overall accuracy.** The **angry** class has a gain of 25.33%, the maximum gain across all classes. The **like** class has the highest accuracy of 63.79%, which attests to the previously made statement about the distinct usage of this class. Details of cross-validation results can be found in Table XIII and Table XIV.

To round off the results, we ran a multi-class classifier using an one-vs-all SVM. We used only the top 1 selection results since SVM cannot return two results. SVM returns marginally better (+1.97%) accuracy than the MNB model with top 1 selection (cf. Table XII), but unfortunately the SVM model does not have the ability to return two results. Refer to Tables XIV and XV) for more details on the results.

D. Six Sentiments Classification: TFIDF vs. Counts

All experiments reported so far were conducted using TFIDF feature values. To understand the impact of TFIDF scores, we ran an experiment using only the counts as feature values (cf. Table XII). We can see that using TFIDF scores produces a model more accurate (+**4.9**%) than using simple counts (cf. Table XII). The **angry** class has the maximum increase (+9.55%) in accuracy among all classes. Details of the cross-validation can be found in Table XVI.

IV. Conclusions & Future Directions

We have shown that using emoji clusters as sentiment labels is an effective way to build a multi-sentiment multi-label classifier. The effectiveness of the model comes from the systematic preprocessing approach to labelling the data using emojis. We have also shown that the models developed, both the multi-sentiment multi-label classifier and the binary classifier, achieve high accuracy. There are also several future directions we are pursuing at the moment. We describe a few of these below.

During the different phases of data preprocessing, we realized that a user typically uses a only a small subset of emojis in very specific and personal context as emojis can be interpreted differently by different people. We are working on a model that can capture these user-specific preferences based on words/features, emojis, topics, etc.

Another important aspect is that emojis do not appear the same across platforms such as iPhone, Android and Twitter. This causes a lot of confusion in the way these emojis are interpreted and used in those platforms. As an example, it is hard to differentiate between a sweat and a tear on some platforms, and consequently they are used interchangeably, whilst that is not the case on other platforms. Additionally, despite the fact that Twitter has its own font for rendering emojis, on mobile it is often overridden by the phone's system emoji font. Therefore, it makes sense to build a model that can understand these platform specific features to improve model performance and perhaps allow for the inclusion of some emojis that we had to currently exclude.

Model	Our Model	Go *et al.* [5]	Pang *et al.* [2], [3]	Barbosa et al [14]	Agarwal et al [11]	Liu et al [18]
NB	82.75	81.3	81.0	–	–	–
SVM	**84.95**	82.2	82.9	81.3	75.39	82.52

TABLE VIII: Two Sentiments Classification: Comparison

Sentiment	CV 1	CV 2	CV 3	CV 4	CV 5	CV 6	CV 7	CV 8	CV 9	CV 10	Average
positive	0.7772	0.7755	0.7775	0.7831	0.7741	0.7714	0.7639	0.7739	0.7641	0.7761	0.7737
negative	0.8797	0.8793	0.8781	0.8809	0.8842	0.8817	0.8801	0.8823	0.8842	0.8826	0.8813
Average	0.8284	0.8274	0.8278	0.8320	0.8291	0.8266	0.8220	0.8281	0.8241	0.8293	**0.8275**

TABLE IX: Two Sentiments Classification: Naive Bayes Classifier

Sentiment	CV 1	CV 2	CV 3	CV 4	CV 5	CV 6	CV 7	CV 8	CV 9	CV 10	Average
positive	0.8622	0.8640	0.8583	0.8594	0.8631	0.8610	0.8646	0.8614	0.8602	0.8645	0.8619
negative	0.8385	0.8353	0.8422	0.8329	0.8345	0.8421	0.8375	0.8322	0.8408	0.8356	0.8372
Average	0.8503	0.8497	0.8503	0.8462	0.8488	0.8515	0.8510	0.8468	0.8505	0.8500	**0.8495**

TABLE X: Two Sentiments Classification: SVM Classifier

Sentiment	CV 1	CV 2	CV 3	CV 4	CV 5	CV 6	CV 7	CV 8	CV 9	CV 10	Average
angry	0.7211	0.7238	0.7183	0.7181	0.7196	0.7163	0.7253	0.7174	0.7303	0.7186	**0.7209**
cool	0.4951	0.4805	0.4882	0.4862	0.4881	0.4907	0.4802	0.4822	0.4882	0.4883	0.4868
joking	0.3662	0.3602	0.3725	0.3619	0.3723	0.3633	0.3593	0.3671	0.3601	0.3680	0.3651
silly	0.4357	0.4412	0.4364	0.4322	0.4345	0.4468	0.4363	0.4419	0.4423	0.4400	0.4387
funny	0.6095	0.6061	0.6026	0.6062	0.6140	0.5988	0.6053	0.6099	0.6132	0.6227	0.6088
good	0.5437	0.5422	0.5295	0.5419	0.5452	0.5376	0.5400	0.5424	0.5371	0.5431	0.5403
love	0.4464	0.4401	0.4502	0.4502	0.4496	0.4544	0.4474	0.4591	0.4497	0.4466	0.4494
like	0.7154	0.7189	0.7088	0.7175	0.7155	0.7103	0.7123	0.7089	0.7152	0.7102	**0.7133**
sad	0.6265	0.6334	0.6276	0.6310	0.6279	0.6403	0.6334	0.6406	0.6271	0.6360	0.6324
smileys	0.4830	0.4916	0.4833	0.4832	0.4871	0.4860	0.4839	0.4932	0.4857	0.4905	0.4868
Average	0.5443	0.5438	0.5417	0.5428	0.5454	0.5445	0.5423	0.5463	0.5449	0.5464	**0.5442**

TABLE XI: Ten Sentiments Classification: Multinomial Naive Bayes Classifier using Top 2 selection

Finally, we are also looking at using a deep neural network model to improve model learning and performance. Tweets provide an ideal input for a deep learning network because they have a fixed length of 140 characters, and our method for label generation allows us to collect enough data to actually be able to train the model. Deep learning networks are inherently multi-class and also allow for multi-labeling, thereby making them a good fit for our problem. There is potential for the deep learning models to extract more generalized features that can be used for other problems such as topic modeling [19].

References

[1] "Twitter usage statistics," http://www.internetlivestats.com/twitter-statistics/, accessed: 2016-06-19.

[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, 2002, pp. 79–86.

[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1-2, pp. 1–135, Jan. 2008.

[4] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL Student Research Workshop*, 2005, pp. 43–48.

[5] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Processing*, pp. 1–6, 2009.

[6] P. Kralj Novak, J. Smailović, B. Sluban, and I. Mozetič, "Sentiment of emojis," *PLoS ONE*, vol. 10, pp. 1–22, 2015.

[7] "List of human emotions," http://www.listofhumanemotions.com/listofhumanemotions, accessed: 2016-06-14.

[8] "Tweet sentiment visualization," https://www.csc.ncsu.edu/faculty/healey/tweet_viz/, accessed: 2016-06-14.

[9] J. Russell, "A circumplex model of affect," *Journal of personality and social psychology*, vol. 39, pp. 1161–1178, 1980.

[10] M. M. Bradley and P. J. Lang, "Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings," Center for Research in Psychophysiology, University of Florida, Tech. Rep., 1999.

[11] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Languages in Social Media*, 2011, pp. 30–38.

[12] V. Kharde and S. Sonawane, "Sentiment analysis of twitter data : A survey of techniques," *CoRR*, vol. abs/1601.06971, 2016.

[13] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, 2010.

[14] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 2010, pp. 36–44.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 3111–3119.

[16] A. Aizawa, "An information-theoretic perspective of tf—idf measures," *Inf. Process. Manage.*, vol. 39, pp. 45–65, 2003.

[17] C. D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.

[18] S. Liu, F. Li, F. Li, X. Cheng, and H. Shen, "Adaptive co-training svm for sentiment classification on tweets," in *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, 2013, pp. 2079–2088.

[19] X. Zhang and Y. LeCun, "Text Understanding from Scratch," Cornell University, Tech. Rep., 2015.

Sentiment	MNB with Top 2	MNB with Top 1	SVM with Top 1	MNB with Top 2 & Counts
angry	0.7697	0.5164	0.4913	0.6742
funny	0.7087	0.5130	0.5186	0.6617
good	0.6868	0.5848	0.6089	0.6716
love	0.6356	0.4609	0.5643	0.6072
like	0.7741	0.6379	0.6138	0.7516
sad	0.7169	0.5010	0.5352	0.6327
Average	**0.7155**	**0.5357**	**0.5554**	**0.6665**

TABLE XII: Six Sentiments Classification: Summary of Results

Sentiment	CV 1	CV 2	CV 3	CV 4	CV 5	CV 6	CV 7	CV 8	CV 9	CV 10	Average
angry	0.7747	0.7690	0.7660	0.7792	0.7661	0.7725	0.7673	0.7667	0.7624	0.7735	**0.7697**
funny	0.7033	0.7106	0.7086	0.7101	0.7058	0.7150	0.7167	0.6995	0.7052	0.7121	0.7087
good	0.6865	0.6907	0.6949	0.6847	0.6790	0.6885	0.6880	0.6900	0.6839	0.6917	0.6878
love	0.6330	0.6340	0.6421	0.6329	0.6410	0.6342	0.6338	0.6268	0.6340	0.6445	0.6356
like	0.7743	0.7696	0.7788	0.7806	0.7714	0.7675	0.7801	0.7666	0.7790	0.7732	**0.7741**
sad	0.7210	0.7104	0.7182	0.7123	0.7257	0.7175	0.7143	0.7177	0.7130	0.7186	0.7169
Average	0.7155	0.7141	0.7181	0.7166	0.7148	0.7159	0.7167	0.7112	0.7129	0.7189	**0.7155**

TABLE XIII: Six Sentiments Classification: Multinomial Naive Bayes Classifier using Top 2 selection

Sentiment	CV 1	CV 2	CV 3	CV 4	CV 5	CV 6	CV 7	CV 8	CV 9	CV 10	Average
angry	0.5110	0.5211	0.5244	0.5133	0.5118	0.5093	0.5210	0.5172	0.5161	0.5244	0.5164
funny	0.5021	0.5111	0.5225	0.5140	0.5098	0.5149	0.5130	0.5102	0.5196	0.5129	0.5130
good	0.5939	0.5835	0.5848	0.5882	0.5885	0.5755	0.5820	0.5860	0.5869	0.5848	0.5848
love	0.4542	0.4502	0.4658	0.4678	0.4575	0.4629	0.4662	0.4569	0.4618	0.4658	0.4609
like	0.6436	0.6338	0.6354	0.6362	0.6396	0.6272	0.6404	0.6419	0.6422	0.6354	0.6379
sad	0.4982	0.5008	0.4947	0.5049	0.5022	0.5015	0.5083	0.5011	0.5030	0.4947	0.5010
Average	0.5338	0.5334	0.5363	0.5374	0.5349	0.5319	0.5385	0.5356	0.5383	0.5363	**0.5357**

TABLE XIV: Six Sentiments Classification: Multinomial Naive Bayes Classifier using Top 1 selection

Sentiment	CV 1	CV 2	CV 3	CV 4	CV 5	CV 6	CV 7	CV 8	CV 9	CV 10	Average
angry	0.4910	0.4885	0.4980	0.4903	0.4883	0.4884	0.4920	0.4926	0.4960	0.4881	0.4913
funny	0.5157	0.5167	0.5097	0.5252	0.5175	0.5197	0.5211	0.5197	0.5146	0.5256	0.5186
good	0.6128	0.6098	0.6109	0.6131	0.6118	0.6052	0.6133	0.6112	0.6026	0.5986	0.6089
love	0.5686	0.5664	0.5609	0.5661	0.5637	0.5582	0.5663	0.5593	0.5695	0.5644	0.5643
like	0.6174	0.6034	0.6117	0.6099	0.6171	0.6153	0.6169	0.6140	0.6197	0.6122	0.6138
sad	0.5315	0.5357	0.5339	0.5346	0.5352	0.5310	0.5362	0.5427	0.5340	0.5375	0.5352
Average	0.5562	0.5534	0.5542	0.5565	0.5556	0.5530	0.5576	0.5566	0.5561	0.5544	**0.5554**

TABLE XV: Six Sentiments Classification: SVM Classifier using Top 1 selection

Sentiment	CV 1	CV 2	CV 3	CV 4	CV 5	CV 6	CV 7	CV 8	CV 9	CV 10	Average
angry	0.6694	0.6802	0.6753	0.6683	0.6796	0.6762	0.6714	0.6736	0.6777	0.6700	0.6742
funny	0.6638	0.6660	0.6652	0.6599	0.6582	0.6666	0.6585	0.6565	0.6584	0.6641	0.6617
good	0.6683	0.6692	0.6700	0.6702	0.6759	0.6747	0.6777	0.6730	0.6686	0.6680	0.6716
love	0.6127	0.6032	0.7584	0.6072	0.6172	0.6028	0.6012	0.6112	0.6028	0.6082	0.6072
like	0.7511	0.7530	0.6053	0.7419	0.7513	0.7510	0.7509	0.7476	0.7538	0.7565	0.7516
sad	0.6307	0.6308	0.6343	0.6265	0.6333	0.6379	0.6394	0.6322	0.6248	0.6373	0.6327
Average	0.6660	0.6671	0.6681	0.6623	0.6693	0.6682	0.6665	0.6657	0.6644	0.6674	**0.6665**

TABLE XVI: Six Sentiments Classification: Multinomial Naive Bayes Classifier using Counts and Top 2 selection