CPS803 Group 12: Project Proposal

Hajesha Kandasamy
Ryerson University
Toronto, Canada
hajesha.kandasamy@ryerson.ca

Nicholas Dhanraj
Ryerson University
Toronto, Canada
nicholas.dhanraj@ryerson.ca

I. Introduction

In the past few years, social media has grown into a platform to voice individual's thoughts. Oftentimes these are criticisms or praises. Companies, realizing these trends, have found social media to be a useful source of feedback. However, for large companies, or products with a large following, there are often too many opinions for humans to evaluate. Our group has therefore decided to create a machine learning program to evaluate these tweets as either positive, neutral, or negative.

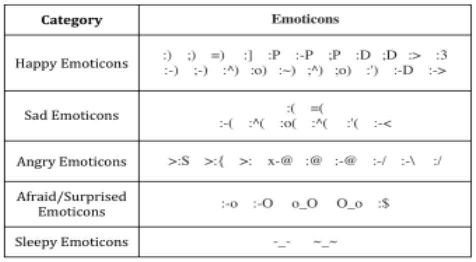
II. Methods

A. Data Processing

To begin with, we are using two datasets for training our project. This is so that we can train on a diverse range of products and brands and encounter distinct terminology associated with various demographics. We will be using the "Coachella 2015 Twitter" (3,846 entries) dataset and approximately 75% of "Brands and Product Emotions" (8,721 entries) dataset provided by Crowdflower. We will use the common features contained in both datasets. These features are the text included in the tweet and either a positive or negative flag indicating the user's emotional inclination. The text of the tweet will be further broken into additional features that were not included in the dataset. The first of these features will be emojis. We will map emojis with the emotions that they are commonly associated with (Table 1). The second is extracting repeated punctuation and single exclamation marks to study the strength of the expressed emotion (ex "McHotdogs Suck!!!"). Similarly, we will be using capitalization as the third feature in determining strength of emotion ("McHotdogs RUN THE WORLD"). Finally, we will be extracting any hashtags from the tweet and consider them as the final feature. We will be using an external library such as WordNinja or a modified Pyenchant to split and process the compounded words.

Both the training and validation data will be cleaned and processed with the following rules and libraries to maintain consistency. We will be removing URLS, numbers (such as phone numbers and dates), user mentions, punctuation (that do not fit the punctuation criteria), and stop words [1]. This is because they provide no additional information that may help our algorithm and by removing them, we may increase the performance instead. We will be using the python NLTK library to provide stop words and aid with the processing. Furthermore, we will further process the data by lemmatizing the words using NLTK's built-in WordNetLemmatizer. The intent is to decouple the meaning of the word from the grammatical structure of the English language [2]. Our final step in processing will be tokenization using the NLTK tokenization library that specializes in casual twitter-aware tokenization. When tokenizing, we will also keep negation into consideration (ex "I am happy" and "I am not happy" have different meanings even though they contain the same trigger) [2].

TABLE I. Example of Emoji Mapping



Category	Emoticons
Happy Emoticons	:) ;) =) :] :P :-P ;P :D ;D :> :3 :-) ;-) :^) :o) :~) ;^) ;o) :') :-D :->
Sad Emoticons	:(=(:-(:^(:o(:^(:'(:-<
Angry Emoticons	>:S >:{ >: x-@ :@ :-@ :-/ :-\ :/
Afraid/Surprised Emoticons	:-o :-O o_O O_o :$
Sleepy Emoticons	-_- ~_~

B. Strategy

To represent and categorize sentiment, we will be using the Russian Circumplex Model. This model has 4 poles, with the West and East poles representing the negative or positive emotion (respectively), while the North and South pole indicate the degree of magnitude [3] (Fig. 1).

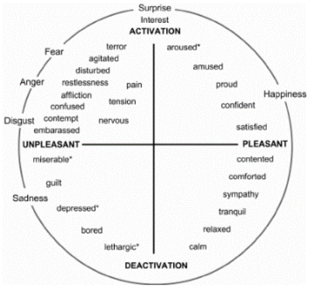


Fig. 1. Russian Circumplex model mapping various emotions

We will be using two baseline models in our project and comparing and contrasting them to determine which would yield the better result. The two models that we will be exploring in our project is the Naive Bayes model and the Support-Vector Model.

The first baseline model we will be looking at is the Naive Bayesian model. This model will use the preprocessed tokenized data and base the output on the individual word frequencies and assumes that they do not have a correlation with each other. This model will use tokens in the form of unigrams (with the exception of negations, as mentioned before). We selected the Naive version of the Bayesian model in particular because the strong decoupled structure of the model, as opposed to other versions which rely on the particular distribution of each feature (as the case in Binarized or Boolean feature Multinomial Naïve Bayes). Furthermore, the Naive Bayesian model allows up to gracefully deal with a reduced number of parameters and is an efficient model with its linear time complexity[4].

$$c_{NB} = argmaxP(c_j) \prod_{i \in postions} P(w_i|c_j) \qquad (1)$$

$$\hat{P}(w|c) = \frac{count(w,c)+1}{count(c)+|V|} \qquad (2)$$

The second model we will use is an SVM (Support-Vector Machine) in which tokenization will be done both as unigrams as well as bigrams. With the objective of the SVM algorithm to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points [4]. It tries to maximize the minimum distance from one of the points to the other given two distinct types of points. In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss, as in (4) & (5).

$$\begin{aligned} &min_{w,b} \ \frac{1}{2}\|w\|^2 && (3) \\ s.t. \quad & y^{(i)}\left(w^T x^{(i)} + b\right) \geq 1, && i = 1, \dots, m \end{aligned}$$

$$c\big(x, y, f(x)\big) = \begin{cases} 0 & {\scriptstyle if\ y*f(x) \geq 1} \\ 1 - y * f(x) & {\scriptstyle else} \end{cases} \qquad (4)$$

$$c\big(x, y, f(x)\big) = (1 - y * f(x)) \qquad (5)$$

III. Experiments

For validation, we will be using the remaining 25% of “Brands and Product Emotions” and the “Apple Twitter Sentiment” (3,886 entries) provided by Crowdflower to validate our dataset. Our intentions with the “Brands and Products” dataset are to test the adaptability of our project and the intention with the “Apple Twitter*”* dataset is to validate an actual scenario that a specific client may encounter. Using this system, we will be able to test the accuracy against the selected crowdsourced dataset (Classification Accuracy). This is simply number of correct predictions divided by the total number of predictions**.** It is not always the case that class distribution is balanced (one class is more frequent than others), in which classification accuracy is not a good indicator of the model performance. One of these scenarios is when your class distribution is imbalanced (one class is more frequent than others) To test for inaccuracies due to this anomaly, we will look at Precision metric.

Precision= True_Positive/ (True_Positive+ False_Positive) (6)

Recall= True_Positive/ (True_Positive+ False_Negative) (7)

For our applications, we wish the percentage of total relevant results correctly classified by the algorithm per relevant results found to be roughly about the same. Naturally we combine these two metrics into one, which is a harmonic mean of precision and recall, also known as F1-score[5].

F1 Score= 2*Precision*Recall/(Precision+Recall) (7)

On a side note, since public opinions are not being published at a rapid rate, we will not be evaluating our project on the basis of time as this is not relevant beyond something that is nice to have.

IV. Milestones

Since our project has a strong NLP base, we recognize that along with the machine learning aspects, there is a lot of preprocessing associated with the project. Along with using external libraries, we recognize that we may need to allocate more time compared to other projects to preprocessing strategies. With that in mind, our milestones will be broken down as follows.

A. Preprocessing

Milestone 1&2 (2 weeks): Filter tweets, lemmatization of Tweets (Person 1) & decompounding hashtags, extracting additional features (Person 2)
Milestone 3 (1 week): Feature Extraction (Person 1&2)

B. Training

Milestone 4 (2 weeks): Training SVM pipeline & model (Person 1) & Training Naive Bayesian pipeline & model (Person 2)

C. Evaluation

Milestone 5 (3 weeks): Evaluate SVM model & pipeline (Person 1) & Evaluate Naive Bayesian model & pipeline (Person 2)

D. Wrap-up

Milestone 6 (1 week): Failure analysis and refinement (Person 2)
Milestone 7 (1 week): Report (Person 1) and Video (Person 1&2)

V. References

[1] Roberts, Kirk & Roach, Michael & Johnson, Joseph & Guthrie, Josh & Harabagiu, Sanda. (2012). EmpaTweet: Annotating and Detecting Emotions on Twitter. Proc. Language Resources and Evaluation Conf.

[2] Badugu, Srinivasu & Suhasini, Matla. (2017). Emotion Detection on Twitter Data using Knowledge Base Approach. International Journal of Computer Applications. 162. 28-33. 10.5120/ijca2017913366.

[3] Roberts, Kirk, and Sanda M Harabagiu. “Statistical and similarity methods for classifying emotion in suicide notes.” Biomedical informatics insights vol. 5,Suppl. 1 (2012): 195-204. doi:10.4137/BII.S8958

[4] “Everything There Is to Know about Sentiment Analysis,” *MonkeyLearn*. [Online]. Available: https://monkeylearn.com/sentiment-analysis/. [Accessed: 02-Oct-2020].

[5] O. Kolchyna, T. T. P. Souza, P. C. Treleaven, and T. Aste, “Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination,” *arxiv.org*, 02-Sep-2015. [Online]. Available: https://arxiv.org/pdf/1507.00955.pdf. [Accessed: 02-Oct-2020].