Sylvester: An Approach to Emotion Classification

Jalil Dennis, Colton Wirgau, Shadi Banitaan
Department of Mathematics, Computer Science and Software Engineering
University of Detroit Mercy
Detroit, MI 48221, USA
E-mail: dennisja@udmercy.edu, wirgauch@udmercy.edu, banitash@udmercy.edu

Abstract

With the rapid growth of user generated content on microblogging sites such as Twitter, there is a high demand to develop tools for identifying people's emotions expressed in text. Emotion classification can give a very good insight to how people truly feel about a given subject. In fact, there are six emotions that are universally recognizable across language and culture. These emotions are Fear, Disgust, Excitement, Sadness, Happiness, and Anger. User generated content usually contains the sentiment or the emotion of the writers at the time that they were writing. The time required to annotate a learning set is a major overhead of the emotion classification task. Reducing the time required to annotate a learning set intuitively would benefit any business or company that wishes to leverage their own classification as it reduces the cost of training a data set significantly. Thus, we have created Sylvester, an approach to conducting automated annotation and classification of tweets based on emoji's or hashtags. An experimental evaluation is conducted and results are evaluated in terms of precision, recall, and f-score.

Keywords: Supervised Learning, Machine Learning, Sentiment Analysis, Emotion Classification

1. INTRODUCTION

Emotion classification is a task that humans carry out regularly. In fact, there are six emotions that are universally recognizable across language and culture. Those emotions are Fear, Disgust, Excitement, Sadness, Happiness, and Anger [1]. Because these emotions are recognized across culture and language, this work assumes that written texts may contain the sentiment of the writers at the time that they were writing. The time required to annotate a learning set is a major overhead of the classification task. Reducing the time required to annotate a learning set intuitively would benefit any business or company that wishes to leverage their own classification as it reduces the cost of training a data set significantly. Thus, we have created and evaluated an approach to conducting automated annotation and classification of tweets based on emoji's or hashtags.

An emoji is a term used to describe a popularly used digital image that is often used in communication within social media, texting, etc. from mobile devices and computers. These emojis often denote some type of emotion and can actually hold a lot of weight in the expression of emotions through text. A hashtag is denoted by a #, and it is used in the realm of Twitter (and other social media) to link concepts to that particular hashtag. A lot of valuable information can be gathered through hashtags in a very direct manner from Twitter.

Emotion classification, especially in this context, can give very good insight to how people truly feel about a given subject. The information gathered could be used in various matters that may prove to be more valuable than other methods. For instance, a business may use reviews from their website to gain insight into what their customers think about one of their products. This may prove to show very lopsided results, as many users would not post a review unless they had an extremely good or extremely bad experience with the product. This could be helpful, but it may not be the feeling of the majority of people who did not have an extreme experience, but may still have an unheard opinion. On the other hand, people seem to voice their opinions on social media, such as Twitter, much more often. This is why Sylvester (our emotion classification tool) could be useful to a business, because it could classify an emotion about any subject based upon what people are actually saying. The knowledge gained could give more insight, in many cases, about how people truly feel.

This paper is organized as follows: Related work and research which aided in making assumptions about human behavior and social media and how that information is useful to this work will be discussed in section 2. Following that, the system will be broken down on a larger scale. Each component of the system will be broken down, discussing how the system works. Our experimentation will be discussed pertaining to

dataset evaluation measures and results. Section 6 concludes the paper.

2. RELATED WORK

The growth of social media has attracted researchers to develop approaches that detect people's opinions and emotions. In this section, the most closely related work to our approach is presented. A study by Suttles and Ide found that hashtags provide an accurate indication of the sentiment of a tweet if that hashtag corresponds with an emotion. They classified emotions based on a set of eight bipolar emotions identified by Plutchik (joy/sadness, anger/fear, trust/disgust, and surprise/anticipation) [2].

To prevent manual annotation, [8] and [9] compiled a large hashtag emotion corpus that is annotated with emotion labels using emotion-word hashtags. Support Vector Machines (SVM) with Sequential Minimal Optimization has been used to build the prediction models. The experiments demonstrated that self-labeled hashtag annotations are consistent. It also showed that self-labeled emotion hashtags correspond well with annotations of expert human judges.

A study by Vo and Collier found that, during earthquakes, all people tweeted in ways that could be recognized as fearful or anxious by an artificial intelligence system. This sets the stage for considering the notion that "most people express themselves similarly. [3]"

Two studies, [4] and [5], show that emoji's can be associated with specific emotions accurately and that the emoji being used, in and of itself, without a training set or any other artificial intelligence technique, can be used to predict sentiment to around 83% accuracy. These results are confirmed by MoodLens which implemented a system that is emoji driven in Chinese [6].

Balabantaray et al. [7] built an emotion classifier to identify the emotion class of Twitter writers. Their approach involved getting several hundred thousand tweets which were given to five judges. Each tweet would be seen by two judges and their agreement was analyzed. The features they decided on dealt primarily with WordNet synonyms, bigrams, unigrams, and left-right context. They measured success by seeing how often the system inferred the correct result. They found that in general the average inference was correct approximately 73 percent of the time. The biggest drawback here is that the original annotation was done manually.

The manual annotation process may take months and thus can affect the results as the language used on twitter may change more rapidly than that [7]. Further the costs of paying people to read tweets, judge them, and evaluate how well the judges agreed is a major overhead intuitively due to finances. Thus, the current state could benefit greatly from a system that no longer requires that many tweets are read and classified manually.

This drives the main goal of this work which is to determine the effectiveness of a system that automatically annotates and classifies tweets based on specific observations. These observations are driven by the above studies and they are, for example, whether the tweet contains a specific emoji that is associated with happiness, or if the tweet contains "#happy."

3. METHODOLOGY

The goal of this work is to be able to automatically classify tweets into emotions based off Ekman's six universally recognized facial expressions of emotion and neutral. This project is different than other sentiment analysis systems because it will conduct automated annotation for the training set. It will do this by mapping emoji's or emoticons to one of the six emotions laid out by Ekman. To achieve this the system created for this design should involve five core components:

1. Tweet Scraper
2. Feature Extractor
3. Automated Annotation Tool with GUI
4. Predictive Model Builder
5. Inference Maker with GUI

The five components are described below. Figure 1 illustrates the flow of the proposed system.

3.1 Twitter Scraper

This is the very first component that we need to create because it supplies all the tweets that will be used in the training set of the predictive model. For Sylvester to have a training set many tweets need to be harvested and stored. One option to attain these tweets is the use of a scraper to get the details about these tweets. However, attaining tweets this way would have provided limited results. For example, getting certain meta-data about the tweets would be more challenging.

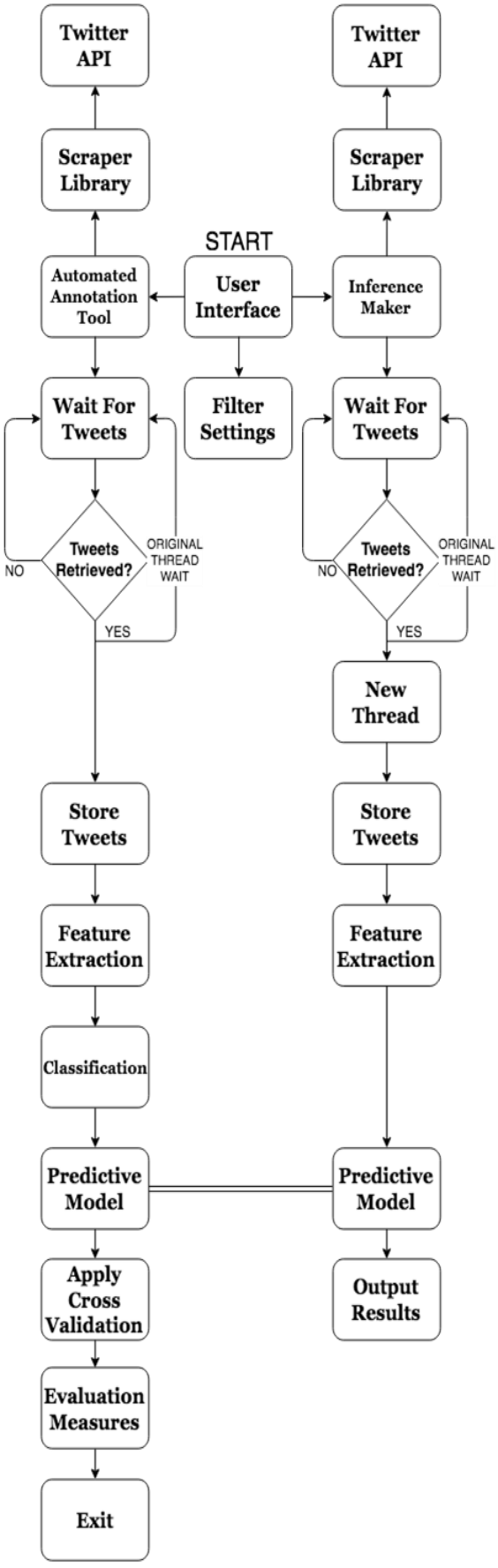

Fig. 1: The Flow of the Proposed System

One prime example of this is the fact that a user who writes a tweet sets their primary language which is then checked for. The language ISO Code is not provided in any of the HTML source on the above page and so a scraper would not be able to retrieve this data. Instead we built a scraper using the new Twitter Stream API. This allows us to see the ISO Code as Twitter will provide it and it allows us to run our own text analysis on the text to assess the language.

3.2 Feature Extractor

The Feature Extractor is another core component to the design of Sylvester. It is used in both the Automated Annotation Tool and the Inference Maker, so it is an integral part of the design. Once the tweets have been scraped and gathered, the Feature Extractor takes the tweets that are gathered and make them more useful to a computer by separating out the tweets by emoji's, removing stop words, and categorizing these tweets based upon those filters. Features are ultimately extracted based on a vocabulary that the extractor creates based on terms document frequency.

The TF-IDF (Term Frequency-Inverse Document Frequency) weight of a term is the product of the term frequency weight and the inverse document frequency weight [10]. The TF-IDF value is high when the term frequency in the given document is high and the document frequency of the term in the whole collection is low.

After a list of all terms has been determined, all tweets are queried for each term and determine a score. This would be done after all tweets are retrieved and before the training set annotation begins so that the vector space model can be created using only terms with, for example, the top thousand scores for document frequency.

One core step of the feature extractor deals with is stopword removal. Many words in the English language add little reason. For example, "the" is a word that appears frequently. Because it appears so frequently it's TF-IDF weight would be extremely high. Given the nature of how this would affect the predictive model that would be created it would be best if the system did not consider words like "the." Sylvester only considers cases where the stopword is located in between two whitespace characters. However, a common typo to occur involves cases where the author might accidentally append or prepend one word to another. For example, "theice" is "the" and "ice" combines. There has been no effort to detect these errors in the project, but the thought is that these typos will not affect the overall output.

3.3 Automated Annotation Tool

The Automated Annotation Tool starts in the GUI which takes inputs from the user to create filters. The Annotation Maker was started by mocking up fake filters that would be passed by hard coding in the results into the Annotation Tool and treating it as user results from the GUI. The first version of the annotation tool had nothing because it literally was a boilerplate. We began by mocking up the data for the output of the user interface, instead of creating the user interface first and then sending that output to the Annotation Maker. The user interface was added after the functionality.

It is important to note that emojis are mapped to sentiments based on Sentiment of Emojis [4]. The paper provides several graphs and tables that help map these emoji's intuitively. Figure 2 provides various emojis and their sentiment score.

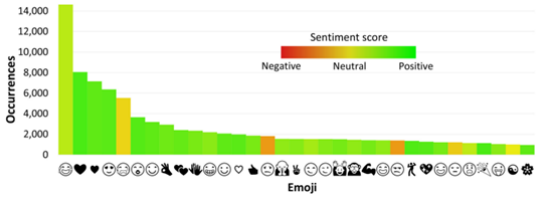

Fig. 2: Sentiment Score of Emoji's [4]

The automated annotation tool makes use of the tweet scraper created. Upon receiving tweets, it first determines whether an emotion based hashtag or emoji is located. If an emoji or hashtag is found it then executes a checking method that determines whether the tweet satisfies requirements for length and language. After confirming that all requirements for the tweet have been satisfied it stores the tweets in memory. The emoji or hashtag is used to annotate the tweet. After finding five tweets the tool begins a MySQL transaction to store those tweets. Once those are stored they are disposed of in memory.

For tweets where there are multiple emoji's or hashtags present we count the occurrences of each in terms of how they relate to each sentiment. The sentiment with the most occurrences is used as a class label. Otherwise, if there is a tie, the tweet is discarded.

Finally, in order to avoid the emoji's becoming a feature, as well as to discard all text that is not written in the phonetic alphabet the tweets are encoded in utf8, thereby casting all non-phonetic alphabet/numeric characters to question marks.

3.4 Predictive Model Builder

One of the major choices we need to make is how a feature set would be determined. Feature selection is an important part of building an accurate predictive model. In order to do this we opted to conduct feature selection based on information gain. The information gain score ranges from 0 to 1 where the attributes that contribute more information will have higher scores.

Given the training data set, the objective is to build a classifier to predict the emotion for new tweets. Two classification techniques are used and compared in this work, Naive Bayes and Random Forest.

3.5 Inference Maker

The inference maker is the final product that an end user will be able to use to determine the sentiment of a filter of their choice in a simple manner. The training set is leveraged and the system will be able to make inferences based on what it has learned. The user would not just want the tweets, but also information on how people feel about the coup, so the inference maker would take that same filter, and the same results that they would find with that filter and display information on the sentiment of tweets about the filter. The inference maker is based on the rest of the core components and is tested properly. The graphical user interface makes the inference maker functional to the end user. The user interface looks great and is extremely easy to use for the average user with little to no computer skills. Figure 3 shows the Inference tool being used to predict the sentiment of tweets that contain the word snow. The design is intuitive and simple to use.

4. DATA SET

Data was collected over the course of a month using the Twitter Stream API. Tweets that were stored had to have emoji's or hashtags that matched with one of the Eckman's emotions. Naturally, the data was not balanced to begin with. The final count of tweets matched with each sentiment can be found in Table 1.

One can see that the distribution of tweets is far from balanced and we needed to combat this issue when building a learning set. Thus, we designed the model builder so that it would ask for how many tweets should be present for each category. If the number of tweets available for a sentiment is greater than what is needed, then the tweets chosen would be a random subsample without repetition of tweets from that sentiment. If the number of tweets

available for a sentiment is less than what is needed, then the tweets are oversampled until the number needed is achieved. We decided that the learning set should have 5,000 tweets from each category to avoid oversampling too much. As a result the learning set contained 30,000 tweets in total.

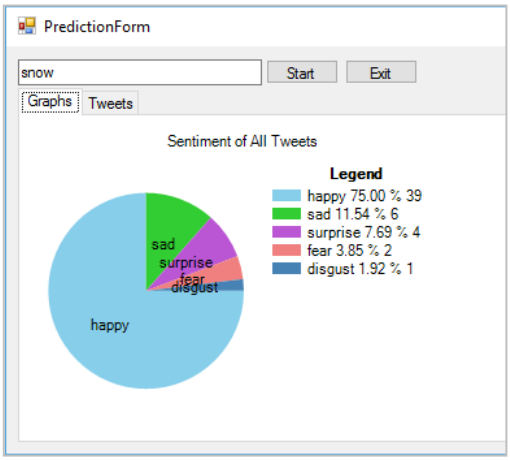

Fig. 3: The Inference Maker Tool

Table 1: Summery of the Data Set

Sentiment	Total	Percent
anger	1991	1.585%
disgust	945	0.753%
fear	4330	3.448%
happy	83756	66.695%
sad	31956	25.447%
surprise	2602	2.072%
Total	125580	

5. RESULTS

After creating vector space models for the learning set, we evaluated each model using 10-Fold cross-validation. The machine learning techniques that were used were Naive Bayes and Random Forest for models with 20, 40, 60, 80, 100, 120, 140, 160, 180, and 200 features where were chosen using the information gain analysis described earlier. The results from the 10-Fold cross-validation was used to calculate the precision, recall, and f-score. Further we reported the time required to build each model. The results can be found in Table 2.

Naive Bayes achieved the highest F-score for models that had 20 and 60 features. Otherwise it was seen that the F-score declined overall. However for Random Forest the F-score continued to increase as the feature set became larger. One possible reason is that Naive Bayes does not combat the effect of over fitting of training sets as Random Forest was designed to. However, it is very clear that the amount of time required to process large models with Random Forest is a major overhead which provides F-scores that are only marginally better than those of smaller feature sets.

Figure 4 better visualizes the F-score by showing Naive Bayes and Random Forest scores side by side. For smaller feature sets the disparity between the two is significantly lower, however the disparity increases as the F-score for Naive Bayes decreases and the F-score for Random Forest increases.

6. CONCLUSION

In this paper, we present Sylvester, an approach to automatic annotation and classification of emotions. Sylvester employs both emojis and hashtags for the creation of emotion labels for tweets. It builds a predictive model for the identification of the six emotions namely fear, disgust, excitement, sadness, happiness, and anger. Both Naive Bayes and Random Forest are used to build the predictive models. The results reveal that Random Forest outperforms Naive Bayes in terms of F-score. The information gain is used to reduce dimensionality of the feature set. The results of applying the information gain show that Naive Bayes achieves the highest F-score for models that have 20 and 60 features. However for Random Forest the F-score continued to increase as the feature set becomes larger. Future directions include expanding Sylvester to be able to annotate and classify tweets in other languages such as Spanish and Arabic.

Table 2: Classification Results using Naive Bayes and Random Forest

	Naïve Bayes				Random Forest			
Feature Size	Percision	Recall	F-Score	Time to Build (s)	Percision	Recall	F-Score	Time to Build (s)
20	0.835	0.692	0.701	0.10	0.784	0.722	0.738	7.87
40	0.792	0.690	0.699	0.10	0.781	0.727	0.741	12.57
60	0.766	0.691	0.704	0.16	0.783	0.736	0.749	22.23
80	0.764	0.680	0.687	0.21	0.781	0.738	0.750	29.58
100	0.783	0.674	0.669	0.27	0.784	0.745	0.757	44.88
120	0.787	0.675	0.671	0.30	0.784	0.747	0.758	62.79
140	0.783	0.672	0.668	0.39	0.789	0.754	0.764	68.00
160	0.783	0.676	0.673	0.42	0.788	0.755	0.765	81.52
180	0.751	0.672	0.675	0.48	0.791	0.758	0.768	112.70
200	0.740	0.671	0.682	0.59	0.791	0.758	0.768	91.65

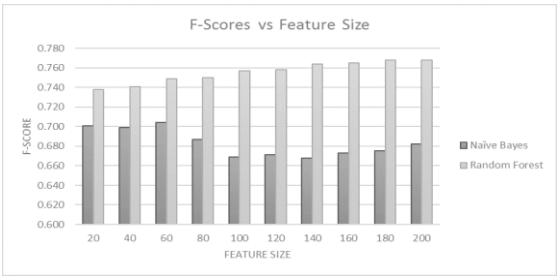

Fig. 4: The Results of Naive Bayes against Random Forest

References

[1] Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research, 50, 723-762.

[2] Suttles, J., & Ide, N. (2013, March). Distant supervision for emotion classification with discrete binary values. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 121-136). Springer Berlin Heidelberg.

[3] Vo, B. K. H., & Collier, N. I. G. E. L. (2013). Twitter emotion analysis in earthquake situations. International Journal of Computational Linguistics and Applications, 4(1), 159-173.

[4] Novak, P. K., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of emojis. PloS one, 10(12), e0144296.

[5] Amalanathan, A., & Anouncia, S. M. (2015). Social network user's content personalization based on emoticons. Indian Journal of Science and Technology, 8(23).

[6] Zhao, J., Dong, L., Wu, J., & Xu, K. (2012, August). 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 (pp. 1528-1531). ACM.

[7] Balabantaray, R. C., Mohammad, M., & Sharma, N. (2012). Multi-class twitter emotion classification: A new approach. International Journal of Applied Information Systems, 4(1), 48-53.

[8] Mohammad, S. M. (2012, June). # 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 (pp. 246-255). Association for Computational Linguistics.

[9] Mohammad, S. M., & Kiritchenko, S. (2015). Using hashtags to capture fine emotion categories from tweets. Computational Intelligence, 31(2), 301-326.

[10] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Scoring, term weighting and the vector space model. Introduction to information retrieval, 100, 2-4.