NLP Final Project: Emoji Predictor

Rachel Connolly, Vicki Liu, Robin Mehta, and Pavithra Vetriselvan
University of Michigan

I. Project Description

Writing and reading short text has become an integral part of modern communication. With the explosion of social media platforms, such as Twitter, Instagram and Facebook, expressing opinions and feelings with a short amount of text has become more important than ever. Due to this, emojis have been increasingly used to enhance semantic meaning of a short text that wouldn't have otherwise existed. Moreover, use of emojis can even generate semantic meaning as opposed to enhancing it. For example: "feeling :)". In this case, understanding emojis is integral to finding the meaning of this text. Emojis are being used more and more on a regular basis to denote the sentiment and meaning of modern natural language.

Currently, consumer apps, such as Apple's iMessage and Venmo[1], suggest emojis by a direct word-mapping as you type out words in text messaging or Venmo payment descriptions. Google Inbox also suggests emojis, but seemingly on a more advanced basis. However, emojis are not used solely to replace words in communication.

For our project we would like to build a system that given a line of short text, similar to what you would find on Twitter, can predict emojis that best enhance and complement the intended meaning and motive of the writer.

Due the the very diverse use of emojis in text, we have decided to only focus on a certain use case of emojis for the scope of this project. Often, users will write out their sentence and then use an emoji at the end[2]. Those are the emojis that our system seeks to predict. Our system does not seek to "rebuild" sentences solely from emojis, nor does it seek translate emojis into semantically and grammatically correct text. Additionally, our system does not predict compound emojis, or pairs of emojis that represent a single concept. Rather, it will rank all possible emojis used and the emoji with the highest ranking will be what our system predicts as the best match for the text.

Although emojis are used in many forms of text such as cellular phone messaging, Facebook, or even email, we have decided to build our system to predict emojis for input from Twitter. Some examples of tweets that include emojis are:

Due to emojis becoming a larger part of our modern language, we believe these considerations will soon need to be incorporated into systems concerning natural language processing and hope this project will provide interesting observations concerning that transition.

II. Related Work

Several studies have tried to determine the sentiment of emojis through human annotation and classification. Novak et.al. (2015) manually annotated about 70,000 tweets in 13 European

[1] http://blog.venmo.com/hf2t3h4x98p5e13z82pl8j66ngcmry/2015/5/14/introducing-emoji-autocomplete

[2]

languages, including English, to create the Emoji Sentiment Ranking. They also found that, on average, emojis tend to be placed at two-thirds the length of a tweet. The emotional intensity of an emoji increases toward the end of a tweet. In addition, tweets that had emojis were more likely to be positive and less likely to be neutral or negative. Lu et.al. (2016) investigated the emoji usage trends in 6.06 billion messages from 3.88 million users in 212 countries. 119 out of the 1,281 emojis comprise around 90% of the usage; the *face with tears of joy* emoji, , makes up 15.4% of emoji usage alone. Moreover, the most popular emojis across different cultures and languages all conveyed emotion. Therefore, they used the Linguistic Inquiry and Word Count (LIWC) program to categorize the annotations of emojis as 'positive', 'negative', 'mixed', 'anxious', 'angry', and 'sad'. Of the 99 emojis with sentiment score, an overwhelming 70% of them were positive.

Barbieri et. al. (2016) used Twitter data specifically to create vector-space skip-gram models of emoji semantics, similar to Mikolov's word2vec. They also created clusters of emojis and words based on topics, such as 'sports and animals' and 'love and parties', by grouping emoji vectors. Their work illustrates the difficulty in understanding the different purposes and contexts that emojis are used in. Wijeratne et. al. (2016) built a machine readable sense inventory for emoji; Emojinet uses word-sense disambiguation strategies to assign English language meanings to emojis.

III. Data Collection

Emoji usage is widespread with the rise of smartphones, social networking sites, and the internet age. Although we had many options for datasets, we decided to use Twitter for a few distinct reasons. First and foremost, the Twitter API is accessible and easy to use, meaning that we should be able to extract tweets using a Python script. Second, emojis are widely used on Twitter and we assume it is because emojis often encompass a particular sentiment without having to use words. Given the character limit per tweet, emojis are a way for users to communicate without compromising the point. Third, tweets can still give us insight into a user's personality without violating privacy. We considered using Slack messages or texts, but ran into a few ethical concerns.

We will write a Python script that uses the Twitter API in order to scrape ~10,000 tweets randomly. These tweets will have to follow the format of "|beg of tweet| |text| |emoji| |end of tweet|" since our program will predict the single most optimal emoji for a given tweet. We will only be scraping tweets that are in English since we are the most familiar with its grammatical structures and slang terminology. There have been similar studies that analyze tweets and emoji usage in a single profile or a single area (i.e. Chicago area). We randomized our data set to have a broader range of types of tweets.

We also chose to exclude retweets since we only want to predict an emoji for unique instances. This allows for a more diverse dataset, which is better for training our algorithm.

In order to clean our data, we will strip tweets of trailing URLs, hashtags (#hashtag), and mentions (@username). We chose to exclude the latter two since they do not provide significant additional sentiment to the tweet. Hashtags often summarize the existing sentiment while mentions are primarily used to directly tag other users in a tweet. After stripping the extraneous information from the tweet, the remaining data will be written to a .csv file with an ID number and tweet text per line.

IV. Method Description

Methodology: Our methodology combines sentiment analysis, word sense disambiguation, and keyword matching to rank the set of emojis for each tweet.

An example of an emoji prediction based on sentiment analysis could be: "I am very **happy** with what we are learning in NLP! " To determine the sentiment of tweets, we will use VADER (Valence Aware Dictionary and sEntiment Reasoner), developed by C.J. Hutto and Eric Gilbert. An emoji's score for sentiment analysis will be the absolute value of the difference between its sentiment and the sentiment of the tweet. To determine the emojis' sentiments, we will use Novack et. al.'s Emoji Sentiment Ranking. Both sentiment scores will be normalized in order to compare them.

An example of an emoji prediction based on keyword matching could be: "I have to walk my **dog** before NLP class. " There are both annotations and keywords assigned by the Unicode Consortium associated with every emoji. The weights of the annotations and keywords are to be determined and could be variable.

An example of an emoji prediction based on word sense disambiguation could be: "**Pray** for my family. God gained an angel today. "

Deciphering the sense of the text of a tweet as well as the emoji is the most challenging task. To consider the semantic space of text and emojis (without building a neural network as has been explored in other examples), we are considering the frequency with which a sense occurs for a word instead of merely the word itself. NLTK provides a Lesk algorithm for Word Sense Disambiguation. According to the NLTK documentation, the Lesk algorithm will return a "Synset" with the highest number of between the context sentence and different definitions given an ambiguous word and the context in which it occurs[3]. We might also be able to relate the sense of each word to Wijeratne et. al's Emojinet sense dictionary.

Twitter data presents the additional challenge of words that do not exist in the English language dictionary, such as "ahahahaha" and "lol". However, we do not want to clean the tweets by eliminating these words because they could still be significant indicators of emoji usage. For these words, without a comprehensive dictionary of ever-evolving Internet slang, they will only have one sense.

The total score for each emoji could be the product of the absolute value of the difference between the sentiment of the tweet and emoji and the product of the probability of an emoji given the word sense for all words in the tweet plus a keyword-matching score. We want the rankings of emoji to reflect how close the sentiment of a tweet is to the annotated sentiment of emoji, how likely it is that the words in the tweets would appear with the emoji, and whether or not specific keywords are present.

Evaluation Methodology: Our goal is to produce a system that can accurately predict the use of an emoji given a sentence or tweet. In order to evaluate the performance of our emoji predictor we will build out a very naive predictor to use at our baseline. For this naive predictor, we will create a mapping of keywords to emoji based on the unicode descriptions[4]. In this case where no keywords appear in a sentence, we will predict some arbitrarily chosen generic emoji. We believe this is the most naive and basic approach to building an emoji predictor and have seen it

[3] http://www.nltk.org/howto/wsd.html
[4] http://unicode.org/emoji/charts/emoji-list.html

referenced as such in similar projects[5]. Thus, if our system does not do better than the baseline predictor, we will consider it a failed system. For our testing methodology, separate our scraped and tagged data into two datasets, one for training and one for testing. Using the training data set we will compare our systems predicted results to the "gold standard" (the emoji actually used in the text) to compute an accuracy. Once our system achieves better accuracy than the baseline predictor on our training set, we will then run our predictor on our test set in order to find the true accuracy range of our predicting system.

References

1. Barbieri, Francesco, Francesco Ronzano, and Horacio Saggion. "What Does This Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis." Language Resources Evaluation Conference (2016): 3967-972. Web.

2. Lu, Xuan, Wei Ai, Xuanzhe Liu, Qian Li, Ning Wang, Gang Huang, and Qiaozhu Mei. "Learning from the Ubiquitous Language." Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16 (2016): n. pag. Web.

3. Novak, Petra Kralj, Jasmina SmailoviÄ, Borut Sluban, and Igor MozetiÄ. "Sentiment of Emojis." PLOS ONE PLoS ONE 10.12 (2015): n. pag. Web.

4. Wijeratne, Sanjaya, Lakshika Balasuriya, Amit Sheth, and Derek Doran. "EmojiNet: Building a Machine Readable Sense Inventory for Emoji." Lecture Notes in Computer Science Social Informatics (2016): 527-41. Web.

[5] https://getdango.com/emoji-and-deep-learning/