Yik Yak Progress Report - EECS 349

Matthew Heston
Northwestern University

Jeremy Foote
Northwestern University

TASK

Our task is to predict the score that a given "yak" would receive on the anonymous mobile social app Yik Yak. Yik Yak is an anonymous, location-aware mobile application in which users can create short posts, and can view and upvote or downvote posts created near their location. While anonymous, location-based messages have been a part of the world for a long time (e.g., writing on the bathroom stall), the particular combination of anonymity and mass social feedback is enabled by GPS-enabled smartphones. Our goal is to understand how people are using this new "place", and whether the norms and uses differ by location.

DATA

In order to make data exploration simpler, we have decided to look at the yaks from two different campuses - Northwestern University and Florida State University. Northwestern was chosen because we are likely to understand local references and more easily interpret results of why some words or phrases are associated with low or high scores. FSU was chosen to offer a contrast in that it is a large state school. It is also the campus for which we have the largest of yaks, making it useful as we have a large amount of training data. We have approximately 40,000 NU yaks and approximately 160,000 FSU yaks. For current experiments, we have discretized score using fairly naive approaches, i.e., binning yaks based on somewhat arbitrary, though intuitive, thresholds (for example, binning all yaks that receive a score of less than 0.)

Features

Our primary features are mostly derived from the message text itself. We have primarily used bag of words representations in our experiments up to this point, though we are considering deriving other features from the text (e.g., length of the post.) We also have the timestamp of when the date was posted, which we have not utilized yet, but have considered using this to derive features as well (e.g., time of day posted.)

Partitioning

We have been using 5-fold cross validation in our experiments up to this point, but we also have held out data to test later. Also, our data includes time and date information and spans about 5 months. This is useful as we may want to train on earlier months and test on later months, as this more closely resembles how an ML system would work "in the wild."

PRELIMINARY RESULTS

We have used word vector representations, using both unigram and bigram models, and trained both naive bayes and logistic regression models as implemented in the Python module sklearn. We have been using both accuracy and F-score as metrics given the skewed nature of our class distributions. Our best average F-score is 67.0% from 5-fold cross validation using a bigram bag of words representation and training a logistic regression model.

We have also created a website which allows a user to input a yak, and to receive the score that we estimate that yak would receive (using a simple linear regression model).

FUTURE WORK

For the remainder of the quarter we plan to:

- Incorporate other types of features (time of day, etc.) and see how they affect performance of our classifiers.
- Extend current word vector approaches (e.g., try tf-idf, counts of words vs. binary whether or not a word exists, etc.) to see if this affects classifier performance.
- Compare features between NU and FSU models to see if they give us insight into differences between campuses.
- Try different realizations of our output variable, such as the median-based partitions that we explained in our proposal, or a log transformed version of the score.

QUESTIONS

We have thought about how best to combine different types of features in one model. Specifically, we are wondering if there are any dangers in combining word vector features with attributes of a message (such as length, time of day, etc.). The bag of words approach obviously creates a super high dimensional, sparse hypothesis space, while the attribute features apply to every yak. Intuitively, it feels like we may need to do something to treat these types of features differently for some of the algorithms that we might use.

Similarly, we have thought about adding higher-level attributes of the texts themselves, such as LIWC categories or sentiment analysis, and wondered if those can be combined with a word vector approach.