Report for Natural Language Processing Term Project

Xuebo Lai (xl1638), Pu Zhao(pz551)

Goal:

We intend to build a sentimental analyzer that can show people's attitude towards a current trending event.

Overviews:

The project will be developed by following the Cross-Industrial Standard Process for Data Mining, which roughly consists of six steps: Business Insight, Data Understanding, Data Processing, Modeling, Evaluation and Deployment. Most of the time for this project would be spent in processing text (data processing). The techniques being applied to this project are what we have learnt in NLP classes for this semester, including tf-idf, pca, logistic modeling, bag of words, word embedding and neural network, etc.

Details of the Projects:

Program

This is really important. Please note that all the coding and results are in the file called "result_program.ipynb". The unlabled tweets data collected by the program is name "outcome1.csv". The label for the first 800 tweets in outcome1.csv is in 200y.csv.

Business Insight:

Today, knowing people's opinion towards a trending event/ political event is very important in multiple aspects. For example, for politician, they might want to know how the public think about their opponents and themselves; for singers, they might hope to see people's reaction to their newly released album; for public relation company, they might intend to see how people react to their clients in order to make better strategies and decisions. As a response to those needs listed above, we create a sentimental analyzer that can manifest how many percent of people are feeling positive towards a topic, how many neutral and how many negative.

Data Understanding:

One of the best way to obtain people's thought is through twitter. Twitter is a platform for individuals to post their real-time believes and thoughts about ongoing events (or simply just

their lives). It is therefore a great tool to see what people currently have in their mind. There are three types of twitter account that can access twitter's api: Enterprise account, Premium account, Regular account. Because we do not have access to enterprise or premium types of account, we are only able to collect data within the past ten days. After careful discussion, we decided to use the word "maga (make America great again)" to query the twitter database. The data we ended up getting is the raw unformatted and unorganized users' data with the tag #maga (make America great again).

Data Processing:

Obtaining the Data:

We decided to use a library called Tweepy which already had many pre-defined function for extracting information and tweets from twitter. Because of the privacy, I have taken out all the information, authentication and credentials that is needed for accessing twitter database. If this application needs to be run or reproduced, please either contact the creator Xuebo Lai for his information or go to twitter application website and apply for those credentials. Sorry for the inconvenience. All the results are still preserved in the Python Jupyter Notebook for grading purposes.

Data preprocessing:

We introduce several factors and eliminate unnecessary tokens to preprocess the raw data. The unnecessary parts of raw data got from Twitter Inc. include

1) the status whether "tweet" or "retweet"
2) the users mentioned in the tweets
3) the author name of tweets
4) the web link and urls inside the tweet
5) some punctuation marks unrelated with sentiments and attitudes of the tweets
6) other implicit natural language usages (for simplicity)

To achieve our goal, we split the raw data each line into tokens and manually matches corresponding features. Also, in order to improve the accuracy, we also eliminate some trivial or negative features — especially we consider those stop words by introducing an exhaustive list of it for matching. Consequently, the overall preprocessing dramatically improves the efficiency and accuracy than direct tokenization from the raw data.

Mapping Sentence to Vector:

For each sentence, we need to map them into vector in order to perform machine learning algorithm on it. In the very beginning, we used the most naïve way, bag-of-word method, to map each sentence to a vector by the CountVectorizer class in sklearn library. Later, we decided to incorporate the value of TF-IDF in building the matrix, so we used the tfidfModleing class in sklearn library instead. The reason we determined to include TF-IDF is because we want to weigh each sentence vector by the term frequency of each word inside the sentence, which should allow the result vector to stress on the important key word more. We ended up mapping around 17,000 sentences about the topic #maga to bag-of-word vectors with their tf-idf as value.

Dimension Reduction(Principle Component Analysis):

One problem of using the bag-of-word approach is that the vectors generated would be sparse matrix. Hence, the dimension reduction is very important in this case. I applied the PCA class in sklearn library. After we observed the explained_variance matrix generated after applying the PCA method, we decided to keep around 60 dimensions, because even after applying PCA, the value is still very sparse in matrix. The most relevant column(dimension) after dimension reduction was only able to manifest around 4% percent information of the original data (normally, this number should be over 50%). In order to preserve enough information from data for modeling, we decided to keep all the columns that reflect more than 0.3% percent information of the original matrix. That's why we decided to preserve 60 dimensions. In order to normalize the data and center them around 0, the de-mean method was adopted.

Labeling Training Data:

When we manually label the data, we only consider the sentiments and attitudes of the posters regardless of political views. In spite of the nature of the tag we use — "MAGA" which might contain a strong political leaning, the major consideration is their sentiments of language; in another word, positive, neutral and negative tag can be assigned to any political opinion or not at all. We ended up training 800 tweets.

Visualization of Data:

As discussed above, the most informative column (dimension) only reflects around 4% of the original data's information so the visualization was not a great success. We used the two most informative dimensions as x and y to map each training sentence vector on the graph. Each dot represents a sentence. The dot's color represents the class that the sentence vectors belong to. Yellow represents positive class; deep blue represents neutral class; purple represents negative class.

Separation of Data into different sets:

Because of the limitation of time, we were only able to labeled around 800 tweets. We used 750 tweets as training corpus and 50 tweets as the testing and validation data. Our goal is to reach around 80% in the modeling stage of accuracy and then apply the model we trained to all the rest of the tweets which are around 17,000.

Modeling and Evaluation

For the modeling part, we used two models: Logistic and Neural Network. We adopted both of the algorithms from sklearn library. As shown by the program, the training accuracy (empirical cost) for logistic is 77.2% and the validation accuracy is 80%. The guess why validation accuracy is higher than training's is that the regularization term in logistic that lower the training accuracy of the data. The empirical cost (training accuracy) of neural network is 82.67% and validation is 78%, which is pretty similar. However, given the time and memory performance of these two algorithms, we decided to go with logistic in the end.

Result(Deployment)

The modeling results were pretty satisfactory, so we went straight into prediction for the unlabeled data. In the end, we utilized the model obtained from logistic algorithm, and obtained the result that 66.3 % of the people think the tag #maga is positive; 7.55% of the users in twitter are neutral towards it; 26.2% find the tag negative.

Work Cited

Joshua Roesslein, Tweepy, http://tweepy.readthedocs.io/en/v3.5.0/getting_started.html

Vicky Qian, Sentiment Analysis blog, http://ipullrank.com/step-step-twitter-sentiment-analysis-visualizing-united-airlines-pr-crisis/

Word2Vect-tutorial, https://rare-technologies.com/word2vec-tutorial/

Prepare text data, https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/ & https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

Essential Training for Data Science, https://www.lynda.com/Python-tutorials/Naive-Bayes-classifiers/520233/601978-4.html