Readme

Introduction：

In Q5, to do sentiment analysis and opinion mining, we collect the data based on topics such as ‘gene editing’, ‘genetic engineering’, and ‘transgene’ from twitter and relevant website. And then we use the mature model used to do sentiment analysis and open labelled datasets to do classification for our data. And finally, we classify 100 pieces of news/comments/articles related to gene topic into 1: positive and 0: negative. And detailed information listed as below.

Methods:

1. Collect relevant datasets:

One of the major challenges in Sentiment Analysis of Twitter is to collect a labelled dataset. Researchers have made public the following datasets for training and testing classifiers. I get the corpus from the open source, and the corpus includes:

IMDB reviews:

This is another older and relatively small dataset for binary sentiment classification, featuring 25,000 movie reviews for training and testing.

Twitter US Airline Sentiment:

This dataset contains Twitter data on US airlines which was scraped from February 2015. Contributors classified the tweets as positive, negative, and neutral tweets.

Twitter Sentiment Corpus:

This is a collection of 5513 tweets collected for four different topics, namely, Apple, Google, Microsoft, Twitter It is collected and hand-classified by Sanders Analytics LLC. Each entry in the corpus contains, Tweet id, Topic and a Sentiment label.

Stanford Twitter:

The training set is collected by querying Twitter API for happy emoticons like ":)" and sad emoticons like ":(" and labelling them positive or negative. The emoticons were then stripped and Re-Tweets and duplicates removed. It also contains around 500 tweets manually collected and labelled for testing purposes. We randomly sample and use 5000 tweets from this dataset.

2. Data Preprocessing: Special Data Cleaning for Tweet

- Hashtags

A hashtag is a word or an un-spaced phrase prefixed with the hash symbol (#). These are used to both naming subjects and phrases that are currently in trending topics. For example, #gene, #gene features. The regular expression used to match hashtags is #(\S+)

- Handles

Every Twitter user has a unique username. Anything directed towards that user can be indicated be writing their username preceded by '@'. Thus, these are like proper nouns. For example, @hkust. The regular expression used to match user mention is @[\S]+

- URLs

The regular expression used to match URLs is ((www\.[\S]+)|(https?://[\S]+))

- Retweet

The regular expression used to match retweets is \brt\b.

- Missing Value

Because we need to do sentiment analysis for these contents, we need to deal with contents with small length: just delete the content that with the length of 20.

- Remove duplication

Because the news data is crawled according to the API, there are some duplicate data, so I deleted duplicate data with contents as identifier.

- Data Imbalance

I just down-sampling this dataset to solve this problem.

3. Feature Extraction

We extract two types of features from our dataset, namely unigrams and bigrams. We create a frequency distribution of the unigrams and bigrams present in the dataset and choose top N unigrams and bigrams for our analysis.

- Unigrams

Probably the simplest and the most commonly used features for text classification is the presence of single words or tokens in the the text. We extract single words from the training dataset and create a frequency distribution of these words. A total of 181232 unique words are extracted from the dataset. Out of these words, most of the words at end of frequency spectrum are noise and occur very few times to influence classification. We, therefore, only use top N words from these to create our vocabulary where N is 15000 for sparse vector classification and 90000 for dense vector classification. The frequency distribution of top 20 words in our vocabulary is shown in figure 1. We can observe in figure 2 that the frequency distribution follows Zipf's law which states that in

a large sample of words, the frequency of a word is inversely proportional to its rank in the frequency table. This can be seen by the fact that a linear trendline with a negative slope fits the plot of log(Frequency) vs. log(Rank). The equation of the trendline shown in figure 2 is log(Frequency) = −0.78log(Rank) + 13.31.

- Bigrams

Bigrams are word pairs in the dataset which occur in succession in the corpus. These features are a good way to model negation in natural language like in the phrase – This is not good. A total of 1954953 unique bigrams were extracted from the dataset. Out of these, most of the bigrams at end of frequency spectrum are noise and occur very few times to influence classifiction. We therefore use only top 10000 bigrams from these to create our vocabulary. The frequency distribution of top 20 bigrams in our vocabulary is shown in figure 3.

Raw	misses Swimming Class. http://plurk.com/p/12nt0b
Normalized	misses swimming class URL
Raw	@98PXYRochester HEYYYYYYYYY!! its Fer from Chile again
Normalized	USER_MENTION heyy its fer from chile again
Raw	Sometimes, You gotta hate #Windows updates.
Normalized	sometimes you gotta hate windows updates
Raw	@Santiago_Steph hii come talk to me i got candy :)
Normalized	USER_MENTION hii come talk to me i got candy EMO_POS
Raw	@bolly47 oh no :'(r.i.p. your bella
Normalized	USER_MENTION oh no EMO_NEG r.i.p your bella

Table 4: Example tweets from the dataset and their normalized versions.

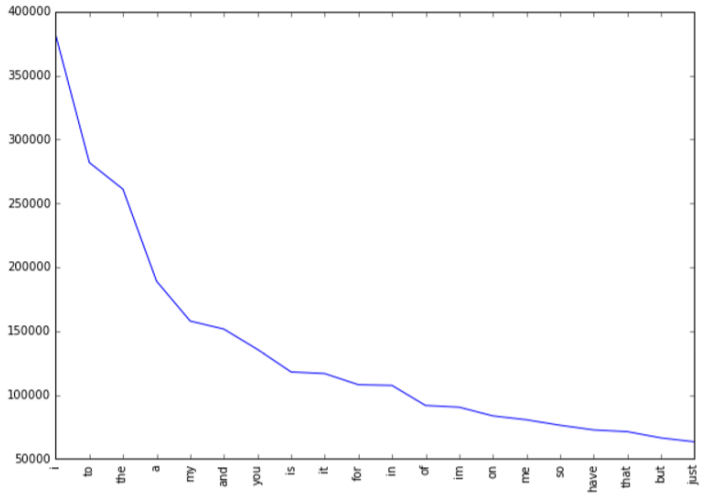

Figure 1: Frequencies of top 20 unigrams.

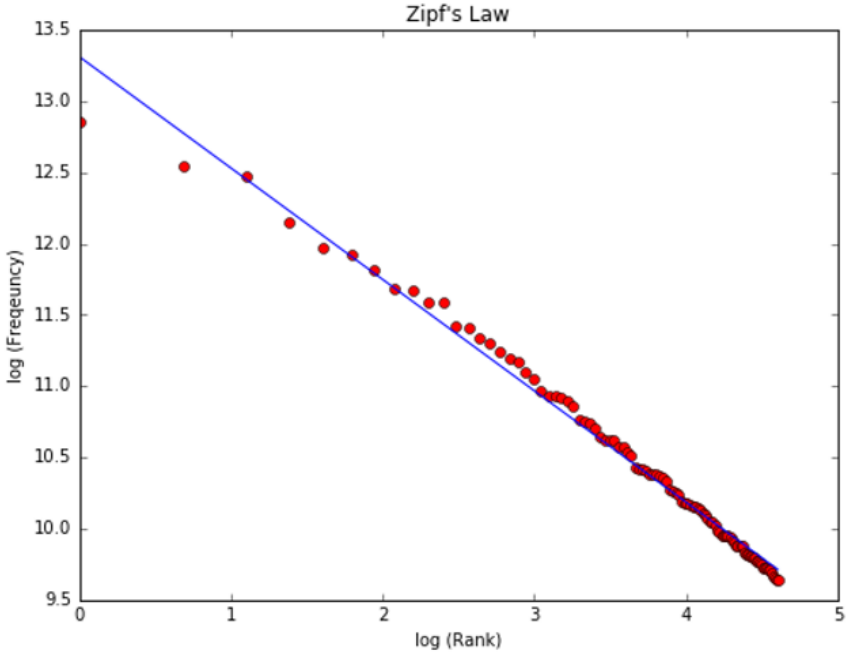

Figure 2: Unigrams frequencies follow Zipf's Law.

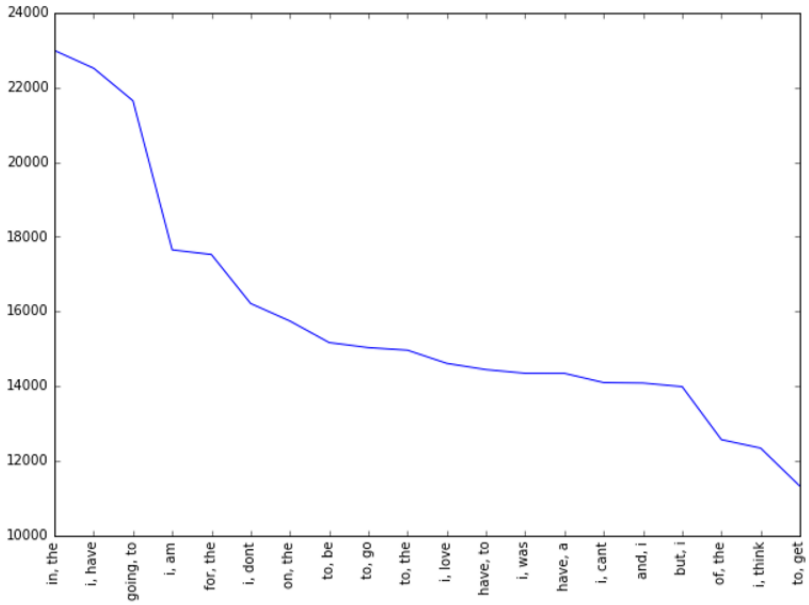

Figure 3: Frequencies of top 20 bigrams.

4. Feature Representation

After extracting the unigrams and bigrams, we represent each tweet as a feature vector in either sparse vector representation or dense vector representation depending on the classification method.

- Sparse Vector Representation

Depending on whether or not we are using bigram features, the sparse vector representation of each tweet is either of length 15000 (when considering only unigrams) or 25000 (when considering unigrams and bigrams). Each unigram (and bigram) is given a unique index depending on its rank. The feature vector for a tweet has a positive value at the indices of unigrams (and bigrams) which are present in that tweet and zero elsewhere which is why the vector is sparse. The positive value at the indices of unigrams (and bigrams) depends on the feature type we specify which is one of presence and frequency.

- Dense Vector Representation

For dense vector representation we use a vocabulary of unigrams of size 90000 i.e. the top 90000 words in the dataset. We assign an integer index to each word depending on its rank (starting from 1) which means that the most common word is assigned the number 1, the second most common word is assigned the number 2 and so on. Each tweet is then represented by a vector of these indices which is a dense vector.

5. Classifier

- Maximum Entropy:

Maximum Entropy Classifier model is based on the Principle of Maximum Entropy. The main idea behind it is to choose the most uniform probabilistic model that maximizes the entropy, with given constraints. Unlike Naive Bayes, it does not assume that features are conditionally independent of each other. So, we can add features like bigrams without worrying about feature overlap. In a binary classification problem like the one we are addressing, it is the same as using Logistic Regression to find a distribution over the classes. The model is represented by

$$P_{ME}(c|d,\lambda) = \frac{exp[\sum_i \lambda_i f_i(c,d)]}{\sum_{c'} exp[\sum_i \lambda_i f_i(c,d)]}$$

Here, c is the class, d is the tweet and λ is the weight vector. The weight vector is found by numerical optimization of the lambdas so as to maximize the conditional probability.

- SVM

SVM, also known as support vector machines, is a non-probabilistic binary linear classifier. For a training set of points (xi,yi) where x is the feature vector and y is the class, we want to find maximum-margin hyperplane that divides the points with yi = 1 and yi=-1. The equation of the hyperplane is as follow:

$$w \cdot x - b = 0$$

We want to maximize the margin, denoted by γ, as follows:

$$\max w,\gamma \ \gamma, s.t. \forall i, \gamma \leqslant yi(w \cdot xi + b)$$

In order to separate the points well.

Summary:

We train the SVM by labelled dataset and classify 100 pieces of news/comments/articles related to the gene topic, and show my result in the output.