Report for Milestone 1: data preparation

Members of the Group: Hamid Pour Mohammad[1], Saeid Entezari,

Mehdi Naghi Lou (and Eza Bakhoda)

Topic: Text Analysis: social media topic analyzing and explanations on users' opinions

April 2021

[1] tbto.exps@gmail.com

A Short Report

The data has obtained from a lot of tweets on Twitter that have been collected for some specific tags (each tag is a class). We first created a developer account. Then, we collected thousands of tweets for 27 different tags such as #science, #love, #technology, etc. with the help of *tweepy* pakage. You can see the code to extract and work with the data named "Downloading_and_init_processing.ipynb". These tweets are all in English and were recorded in March 2021; the number of tweets is 98681. When we were collecting the data in the initial stage, the exact date of each tweet, ID, username, text of tweet, length of tweeted text, desired tag, and type of filtering (retweet) are recorded; we saved this file as "tweet_information.csv".

Some of the tweets also used emojis, uppercase letters, replicas, etc. To solve these issues, we changed or removed these things. So, what remains is just textual data. It should be noted that we converted each word to its root , in each tweet; for example, the root of 'making' is 'make'. Also, we searched for 266 frequently used words in each one of 27 classes (tags) and made a long list of these words; this list has 1785 words; some of them had common words. Then, in each tweet, we kept only the words that were among those frequently used words.

The final file has 98681 rows, and 452 columns. The first 451 columns are the features (binary digits), and the last column is the label of each class. To make this file, we first related each of 1785 common words to a binary string. For example, the binary string of 'desert' is '11011110110'. Since the maximum length of each tweet is 41 words, each tweet has $11 \times 41 = 451$ digits; we concatenated these 451 digits to make a single long string for each tweet. Please note that if the length of a tweet was less than 41, we added 'Space' characters to make the tweet longer; the binary string for each Space is '00000000000'. Also note that OneHot method did not work well for these data, because the number of elements for each array (for each word) was larger than 1600; thus, we had some problems about the dimension of data and saving the features. That's why we used binary strings instead of OneHot. So, we have had a map from 41 words in a single tweet, to a binary vector with 451 elements. The last column is for the label of each class. We have 27 classes; so the labels are from 0 to 26. These labels are the target tag of each tweet.

Finally, we created the X matrix as a $98{,}681 \times 451$ matrix, named "data_clean.py"[2]. So, the X matrix has 451 columns, each of which represents a single binary digit; each 11 digits, show a word, or the End Spaces. The Y matrix shows the label of each class (tag) as a number; Y is a column for 27 classes. So, we want to relate X to Y, as a classification problem, with 451 features and 27 classes.

[2] Please note that the final file on GitHub, has 25000 tweet of total 98681; because GitHub has file volume limit, and we can not put all the data there.

Basic Statistical Analysis on the Data

The original data was text, so we did text processing on it. The maximum text length of a tweet is 41 words, and the average length is 11.57 words. The distribution of some very common words (such as 'a') was so strange. For example, in the #bad class, ‘a' was the second most used word; while in the #love class, the word 'a' was not even one of the twenty most used words. That’s why we didn’t eliminate the ‘a’, or other prepositions. It should be noted that we found that tweets related to each tag have unique literature to that tag; each tag uses almost unique word distributions for its tweets (it’s related to the Bag of Words for that tag). About this case, in tweets with #bad, the words ‘alone’, ‘kill’ and ‘young’ are seen significantly. It seems that the relation between words can show new sight to the psychology of social networks. In addition, we considered all 4 tags together as a category and obtained the frequency of their use in tweets.

As we said, we only care about repetitive words in each class; thus, we kept 1785 words and deleted the rest—in each tweet. The distribution of these 1785 words among different classes is such that some of them were able to be among the most repetitive words in all 27 classes; some of them could only be in one class. So we can draw a histogram for these 1785 words to show how many words could be repeated in several classes. Fifty words have been frequent in all classes (such as prepositions); 270 words have been frequent in two classes; 920 words have been frequent in one class (such as ‘close’, ‘desert’, ‘monkey’, etc.). It means that the share of prepositions is not very large in our model. Instead, some less frequently used words have a large share.

A List of Articles

Brief paragraph of some articles, and their significance/relevance to our project
In this paper, they focus on the analysis of tweet posts in order to identify the opinion polarities as well as the topic-category to which they belong. Since standard NLP (Natural Language Processing) techniques fail to capture the information needed for extracting such information, a combination of NLP and Machine Learning techniques seems more promising for this task. That's why they propose to show that NLP techniques feed classifiers with richer features. Their contributions are on automated tweet classification of political tweets. On the other hand, we are trying to classify tweets according to topics and opinion. Also, it seems that NLP can be related to the root of the words used in tweets; we use the root of each word, and Machine Learning techniques [1].
Word and phrase frequencies, ignoring their order in documents, is the most basic and essential linguistic representation—the “bag of words” representation. Word frequencies capture lexical variation, which is extremely important since words are most basic linguistic units of meaning in text. They are also practical, since words and phrases can often be identified through relatively simple programs for text tokenization. Also, about the opinion, one important aspect of an opinion might be a positive or negative form. Opinions are important for many social analysis problems, but are often difficult to reliably extract. Much work in this area uses word frequency approaches, though ongoing work in structured sentiment analysis will be important going forward. We have provided 1785 tokenized words to focus on the frequencies and positions. We think the order of words can be important in a tweet, as the frequencies are [2].
In the research community, the dominant approach to “The automatic classification of texts into predefined categories” is based on machine learning techniques; a general inductive process automatically builds a classifier by learning, from a set of preclassified documents. Since we also want to categorize tweets into predefined categories, it is necessary to consider the process of his study. his research is about how to Filter the text and Categorize the text from different perspectives. According to his paper, a filtering process can be considered as ML classification process. A filtering process happens when given a predetermined integer r, techniques for term selection attempt to select, from the original set T, the set T′ of terms (with $\|T'\| \ll \|T\|$) that, when used for document indexing, yields the highest effectiveness; we have had the same approach for filtering [3].

References (List of articles; which are described above in the paragraphs):

1. NLP-based feature extraction for automated tweet classification. Proceedings of the 1st international conference on interactions between data mining and natural language processing, vol. 1202, DMNLP'14. Aachen: CEUR-WS.org; 2011. p. 145–146.
2. O'Connor, B. T. (2014). Statistical Text Analysis for Social Science. Ph.D. Thesis. Carnegie Mellon University.
3. Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 1–47.