

# Project 2

Bilguun Chinzorig, Monika Avila, Lkham Nyambuu  
EPFL

**Abstract**—The aim of this project is to conduct a twitter sentiment analysis. Our goal is to classify untaged tweets into two groups: positive and negative. For this purpose, we begin our work by preprocessing the tweets and generate the dictionary. Following, we create the word space, i.e. matrix of embeddings using GloVe methodology. After, we span the tweet space by retrieving features from the embedding matrix. In addition to these features, we constructed additional variables that capture the meaning and the importance of the words as well as hashtags included in each tweet. Finally, we used our features in two types of classifiers: 1. Random Forests and 2. SVM. We conclude that random forests outperforms SVM considering both computing efficiency and classifying results.

## I. INTRODUCTION

### II. THE VOCABULARY CONSTRUCTION: WORD PREPROCESSING

We begin our work by preprocessing the available tweets such that we can define our vocabulary. However the words in the dataset are not easily separable by whitespaces due to following reasons:

**Separators:** Words can be separated by multiple characters including whitespace for words, period for sentence ending, comma for clause endings, dash for connected words and : or ; to for beginning independent clauses. The problem is that the last two characters can be used as emojis.

**Word contractions:** word contractions can be viewed as complete new token, but in the end it is just combination of two words. Common word contractions are related to to be's and modal verbs.

**Special words:** Due to freedom of writing tweets, we can observe multiple emphasizes on words including hashtags, and repeated characters like (hey to heyyyy). These words must have a special treatment but for vocabulary building these variations were eliminated.

**Stop words:** The full list of stop words can be found here <https://kb.yoast.com/kb/list-stop-words/>. In our case, we are assuming pronouns like "the" can represent some meaningful information since it emphasizes following nouns.

**Word variations:** In english word can take multiple forms like plural form, verb tenses, incorrect spelling etc. Hence, simple word separation is not enough. And also in english, people use "'s" or "s'" to represent possessions REPHRASE THIS!!

**Numbers:** we assumed that numbers usually conveys factual informations which is not helpful to identify the opinion of a person. Moreover, we need to treat numbers different from words. Thus, we have completely removed every numbers.

In order to overcome the problems mentioned above, we created a customized vocabulary building algorithm with the following procedure:

First we tokenized the text using white space, comma, period, new line. We excluded : and ; since they are maybe part of emojis. Following, for each token we look for word contractions. In our case, we only considered hashtags, common contraction list to separate tokens which contains multiple words. Moreover, we have removed possessions from each token. After this, we address the issue of word variation by shrinking consecutive repeated characters, e.g. we replaced "foooooot" by "foot". Finally, we used their stems to build our vocabulary.

This preprocessing lead to an increase of the tokens in 80,000. Moreover, the total number of unique words in decreased significantly. Indeed, we have successfully obtained almost 32,000 unique words. This means that we ended up with just 32% of all the tokens produced after splitting. (Clarify)

Now let's look at the distribution of each words. The graph 1 shows the distribution of the words of our final vocabulary. As expected, it shows similar relationship as Zipf's law, but note that the number of words with only 1 occurrence in the entire text is important. Indeed, almost 50% of the vocabulary takes only 1.2% of the text which implies that it is not worth keeping the 50% of the words for simplicity.

Finally, in order to increase efficiency of the information contained in our final co-occurrence matrix we apply term frequency- inverse document frequency (tf-idf) feature. This factor gives more importance to words that are repeated in a tweet but not highly recurrent in the whole corpora.

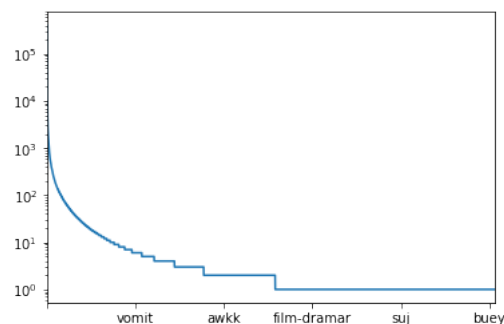


Fig. 1. Word Distribution

### III. THE WORD SPACE: EMBEDDING MATRIX

In order to obtain a numerical representation of the words  $w_i$  contained in the data set as  $\mathbf{w}_i \in \mathbb{R}^K$  we used GloVe model.

This model retrieves the numerical representation of the words that is closest to natural logarithm of the observed co-occurrence value  $x_{ij}$ . Indeed, Pennington et al. (2014) propose to obtain the real valued vector by minimizing a weighted sum of squared errors. Where the error is the difference between the log of co-occurrence value and the predicted one given by  $\mathbf{w}_i' \mathbf{w}_j$ . Thus, the model minimizes the following cost function:

$$L(\mathbf{w}_n, \mathbf{w}_c) = \sum_{j=1}^N \sum_{i=1}^D f(x_{ij}) (\log(x_{ij}) - \mathbf{w}_i' \mathbf{w}_j)^2$$

with:

$$f(x_{ij}) = \min(1, x_{ij}/x_{max})^{3/4}$$

### IV. THE TWEET SPACE: FEATURE RETRIEVAL

In order to obtain a numerical representation of the tweet space we retrieved two types of features. The first class is obtained using GloVe matrix of embeddings. On contrast, we developed the second type of features with the aim to capture the syntax.

Using the GloVe matrix of embeddings, we used the mean of the word embeddings.

On the other hand, the features that we developed are:

- 1) The mean of the tf-idf.
- 2) The weighted sum of the tf-idf, which means that we weighted hashtagged tf-idf, positive tags tf-idf, negative tags tf-idf and emphasis tf-idf.

### V. THE CLASSIFICATION MODEL

After obtaining the tweet space i.e. the features that represent the data set of tweets, we trained the classification model. For this aim, we used two models: 1. Random Forest and 2. SVM.

The first model is random forests, this is a method based in an ensemble approach composed of a set of decision trees. At each node, the variables used to divide the space of the independent variables are randomly selected and used to create the decision tree (Breiman (2001)).

The second classification algorithm used is SVM. In this method, we look to maximize the geometric margin between positive and negative tweets subject to the constraint that the classified vectors must lie outside this margin<sup>1</sup>. This optimization problem is equivalent to the following one for each training sample:

$$\begin{aligned} \min_w & ||w||^2 \\ \text{s.t.} & y_i(x_i'w + b) \geq 1 \end{aligned}$$

Now, setting the dual problem we can obtain the support vectors which are few training observations that lie on the margin.

### VI. THE DATA

The training data set is formed by 2 millions of tweets, half of them correspond to the positive emotions and the other half to negative ones. This data has been already tokenized, thus we start with the cleaning and the analysis of the data.

### VII. RESULTS

### VIII. DISCUSSION

### IX. SUMMARY

### REFERENCES

Breiman, L.

2001. Random forests. *Machine learning*, 45(1):5–32.

Pennington, J., R. Socher, and C. Manning

2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Pp. 1532–1543.

<sup>1</sup> AndrewNg, Support Vector Machines