A classification approach for Sentiment Analysis

Sarina Takalloo
Politecnico di Torino
sarina.takalloo@studenti.polito.it

Faezeh Kazamihatami
Politecnico di Torino
faezeh.kazemihatami@studenti.polito.it

***Abstract*—In this report, a possible method has been proposed to address the problem of the Sentiment Analysis, associated with Tweeter dataset. The main goal is to classify the text into positive and negative views. Three approaches have been introduced for this problem including Logistic Regression(LR), Support Vector Machines (SVMs) and Naive Bayes (NB). In particular, the studied approach, consists of three main steps, Pre-processing the data and convert text data into vectors using Tf-IDF, model selection and classifying the model. Utilized method based on Logistic Regression obtains overall satisfactory results.**

***Index Terms*—SVM, Sentiment Analysis, Tf-IDF, wordToVector**

I. PROBLEM OVERVIEW

Sentiment Analysis is a sub-field of NLP that tries to identify and extract opinions within a given text across blogs, reviews, social media, etc. Twitter is a rich source to learn about people's opinion and sentimental analysis. [1] The goal of this project is to correctly identify the sentiment in each tweet by giving "0" to negative tweets and "1" to positive tweets. The data set is divided into two parts:

- A *development* set of 224,995 unique and non-empty entries characterised by a nominal feature as a numerical label class (sentiment), a text column contains the tweets (text) and a set of nominal (ids,user), interval(date) and categorical(flag) attributes. This dataset is needed to build a classification model for correctly labeling the points in the evaluation set.
- An *evaluation* set of 75,000 data points with the same structure of the development set, Using to examine the accuracy of model.

To more visualize, the frequency distribution of positive class and negatives is shown in Fig.1. .

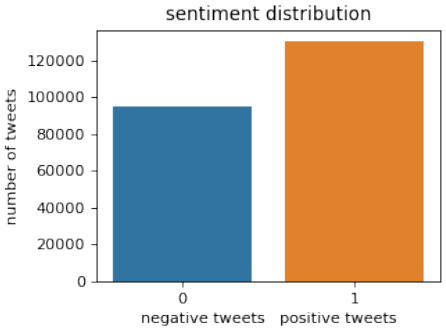

Fig. 1.

Based on the development set, some consideration can be obtained: First, all the labelled reviews in this set, have been distributed as 92220 negatives and 19532 positive, which can be considered as a balanced ratio. Second, for each text, there is a representation of a feeling, "0" as negative and "1" as positive emotions. To achieve a better understanding, two wordclouds showing different distribution word for each class have been illustrated which is a visualization technique in which the size of each word indicates its frequency or importance. most frequent negative and positive words can be seen in Fig.2. and Fig.3, respectively.

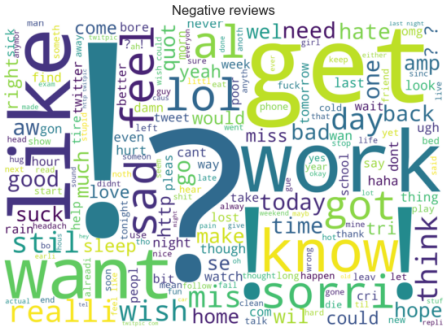

Fig. 2.

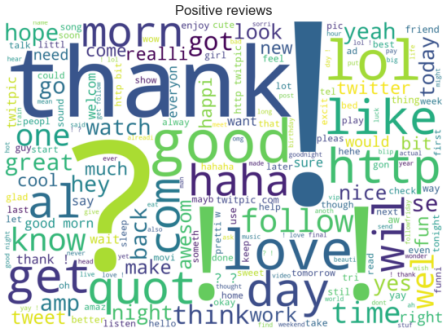

Fig. 3.

We can have better analysis, considering tweet length for each class. In our dataset, negative tweets have more length than positive tweets.

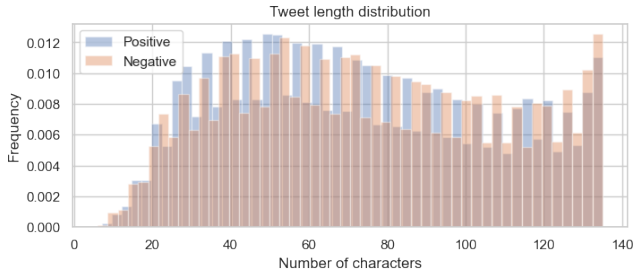

Fig. 4.

II. Proposed Approach

A. Data Preprocessing

We first focus on the data preparation steps needed for textual data. The pre-processing is used to clean the raw data, in order to obtain a suitable format for the classification algorithm. So, the primary step is assigned to determine how many missing values have to be worked on. Therefore, a function is represented to calculate the null items. The result showed Zero amount of missing values. Since the task is to detect emotions; one of the ways to present feelings is using emojis and emoticons. Therefor, these items are converted into text for a better intuitive. Next move is data cleaning, which consists of Tokenization, Case normalization, Stopword removal and stemming, that all will be explained in detail. [2]

1) Data cleaning: The process begins by clean-text function, a method to remove unrelated objects from the text. Including URLs, usernames, numbers, special characters and stopwords. This will cause elimination every irrelevant data. Then, it continues by stemming and returns each word to its root. After that, the Tokenizer-ours function divides the document into words, with the help of nltk library.

2) Weighting Schema: The Tf-IDF, term frequency inverse document frequency, is a tool to extract feautures. Every token produced by the prevoius step is considered as a separate feature, therefore a document is represented by a vector of weights, one for each distinct token. The weight of each token is computed with the term frequency-inverse document frequency (Tf-IDF) weighting scheme and it is computed as it follows:

$TF - IDF(t) = freq(t, d) \times \log \frac{m}{freq(t,D)}$

where t is the term, D is the collection of documents d and $|D|$ is its cardinality [2]. In other words, the frequent terms present in our collection of document have a higher Tf-IDF. To calculate this measure, a Tf-IDF vectorizer function imported from scikit-learn library is applied on the cleaned data.

B. Model Selection

In order to choose the appropriate model for this kind of task, some different classifiers have been evaluated. Here is a discussion about proposed models for the registration task:

1) Naive Bayes: Naive Bayes is a simple technique for constructing classifiers; models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. Since a fast algorithm was demand to determine which model would work better for our classifier, we used Naive Bayes. This function searches all different possible cases with assigned parameters and determined model. First, a NB function with auto parameters was used to evaluate the general performance of this method, the outcome was obtained 0.7656. Then, with help of a grid search, results showed that the alpha equals to 1.1 has the best performance among values between 0.1 to 2. The best score is obtained 0.7660.

2) Support Vector Machines: After evaluating model with NB, a more accurate solution was decided to being tried, so Support Vector Machine classification has been chosen for next classifier. SVM technique mostly is used for binary classes and gives a reasonable accuracy. So, based on the class distribution, a grid search is defined on parameter C to evaluate model accuracy. To achieve best possible performance, different parameters for C were compared, among 1,10,100 and 1000. he score obtained the value of 0.7565 for C=1 as best parameter. Then, C was changed for different amounts within range 0.9 to 1.3 and C=1.3 had the best score and it changed to 0.7696. In the next step, new Tf-IDF was performed with foremost parameters, the score changed to 0.7750.

3) Logistic Regression: In this project, we make use of the Logistic Regression algorithm to build a model, as it is one of the most common approaches for sentiment and social media analysis. It identifies the probability of occurrence of an event by fitting data to a logit function. [3] For this method to examine the score of each change in the algorithm and select the best one, various values were used for C among 1 to 2, with step=0.1. Best score has been gained equal to 0.7800 for C=1. Then, by applying a new Tf-IDF, the score of this algorithm improved to 0.7817.

C. Hyperparameter Tuning

Why hyperparameters should be tuned? Since computation cost is an important problem in classification algorithms; we need to select optimized parameters as possible as we can, and to avoid curse of dimensionality, we have to choose an acceptable amount of features. To assess each model, first we defined Tf-IDF vectorizer by initiating min-df=2,it selects only words that are present in more than 2 reviews . Also, we set max-features equal to 10000, to avoid overfitting. Then, we evaluate models and for the two best of them, SVM and Logistic Regression, we performed a search on vectorizer and Tf-IDF to determine best parameters for min-df, max-df and ngram-range. Result for SVM are max-df=0.3, min-df=1 and ngram-range=(1, 3), and for Logistic Regression are max-df=0.3, min-df=4, ngram-range=(1, 3). Basically, we split the original dataset into a training set and a test set. The classifier fits to the training set, while the test set is reserved for evaluation. To ensure that the class distribution of both sets are similar, we use scikit-learn's train test split with the parameter stratify. We hold out twenty percent of the available data as a test set and use the remaining eighty percent for training and validation. We used the grid search approach with cv=3

to search exhaustively a subset of the hyperparameter space for the best performing model, using the weighted F1 score as our metric. Hyperparameters control the learning process of the algorithms, therefore we should set their value before the training begins. The only hyperparameter for naive Bayes classifiers is the additive smoothing parameter . The Support vector Machine classification algorithm has a regularization parameter C and kernel coefficient which is required to be set. In fact, the choice of a kernel assigned to linear at the first because of class binary distribution. And, Logistic Regression, also has a parameter C, a regularization parameter, to be tuned. For the naive Bayes classifier, we searched for the best additive smoothing parameter value between 1 and 2, with increments of 0.1. The hyperparameter value that gives the best results is = 1.1 for the classifiers. For the support vector machine, we searched for the best regularization parameter value between 1, 10 100 and 1000. The combination of hyperparameters that performs best is C = 1 with the linear kernel. Therefor we examined values between 1 and 2, which C equals to 1.3 gave us best possible result. For Logistic Regression, different parameters for C, penalty and solver l1 and liblinear respectively has set manually. Best score has been achieved by C = 1.5, penalty=l1 and solver=liblinear. Then, Tf-IDF parameter tuning has been applied to achieve best outcomes. All used hyperparameters are shown in Table.1.

TABLE I
SUMMARY OF HYPERPARAMETER TUNING

Model	Hyperparameters	Values
Logistic Regression	'C'	range(1,2)
SVM	'C'	[0.9,1,1.3,10,100,1000]
Naive Bayes	'alpha'	range(0.1,2)

III. RESULTS

As you can see in Table II, All three classifiers achieve satisfactory performance. But we can see that Logistic Regression has a better accuracy, equals to 0.78. The other two classifiers gain 0.76 and 0.77 for accuracy respectively for NB and SVM. The result has been illustrated in Table.2.

TABLE II
BEST HYPERPARAMETER CONFIGURATIONS AND SCORES

Model	Best Parameters	F1-Score
Logistic Regression	C:1.5, penalty:'l1',solver:'liblinear'	0.781
SVM	C:1.3	0.775
Naive Bayes	alpha : 1.1	0.765

IV. DISCUSSION

The proposed approach obtains results that far outperform the Logistic Regression defined. We have empirically shown that the selected classifiers perform similarly for this specific task, achieving satisfactory results in terms of macro f1 score. The following are some aspects that might be worth considering to further improve the obtained results:

- Data preprocessing needs to be more accurate; although we tried our best to clean the data, but, the dataset is tricky. Social medias in text format are included of so many slangs, irony and informal conversations in the intended language. So, a very specific approach is required to detect and label this texts.
- Hashtags and trendy subjects are demanded to work on separately; which means each hashtag have a specific meaning concept that cannot be detect as a meaningful word. To overcome these two mentioned issues, a very strong pre-processor is required for this task.
- Utilizing the Artificial Neural Network algorithms; One of the robust methods for solving learning problems, is using RNN. While it can gives us more accurate approach, it will cost more time and resources. Therefor, if the time and hardware resources are available, it will be a very optimal solution.

REFERENCES

[1] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[2] Huma Parveen and Shikha Pandey. Sentiment analysis on twitter dataset using naive bayes algorithm. In *2016 2nd international conference on applied and theoretical computing and communication technology (iCATccT)*, pages 416–419. IEEE, 2016.

[3] WP Ramadhan, STMT Astri Novianty, and STMT Casi Setianingsih. Sentiment analysis using multinomial logistic regression. In *2017 International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC)*, pages 46–49. IEEE, 2017.