Project 2 on Machine Learning
Text classification
Team Yoor

Sergei Volodin[1], Baran Nama[1], and Omar Mehio[1]

[1]EPFL
{*sergei.volodin,baran.nama,omar.mehio*}*@epfl.ch*

***Abstract*—A classification dataset consisting of Tweets is being studied. First, the data is thoroughly explored using visual aids. Several basic Natural Language Processing methods are applied. Results are evaluated using cross-validation. Model overview is given and the best model is chosen.**

I. Introduction

This paper investigates into improving the quality of sentiment analysis on Tweets dataset [7]. It consists of $N_1 = N_2 = 1250000$ positive/negative tweets, each of them representing a message in English and numerical alphabet Σ with no longer than 140 characters. This way, each of $N = N_1 + N_2 = 2500000$ tweets is assigned to one of the classes $\mathcal{C} = \{:(, :)\}$. The task is to minimize the classification error. In other words, if $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ is the dataset with tweets $x_i \in \Sigma^*$ being messages and $y_n \in \mathcal{C}$ being class labels, the goal is to train a classifier $f\colon \Sigma^* \to \mathcal{C}$ which minimizes the loss function $l(y, \hat{y}) = [y \neq \hat{y}]$.

The task of sentiment analysis of tweets was thoroughly studied [2], [3], [4], [5]. These solutions usually give accuracy around 80%. Several techniques were applied, mostly consisting of two steps. First, the words are converted to dense vectors using Glove, word2vec, cbow or skip-gram models. After that, the resulting word vectors are used to construct features for the whole tweet. At the end, the vector is feeded into a classifier, such as SVM or Logistic Regression. Two latter steps might be replaces with a neural network accepting variable-length input such as RNN or CNN. Moreover, the embeddings themselves might be trained using backpropagation while training the classifier.

Claim: it is possible to find a model which fits the data better than the current state-of-the-art using expert knowledge on the Tweets dataset.

Next sections describe in details our approaches and compare them to various baselines.

II. Models and Methods

This paragraph describes the data being studied. Tweets are short messages no longer than 140 characters long [1]. The table I represents a few examples from the small dataset. Being considered an informal way of communication, tweets often contain misspelled words and letter repetitions, grammatical and other writing mistakes. Besides plain text with punctuation, tweets also contain hashtags, two types of tags, *user* and *url*, which are tokens for replaced user mentions and URL links, respectively. In addition, tweets sometimes contain emoticons. Despite the fact that objects in the training dataset mostly comply with its classes, some tweets cannot be determined as positive/negative even by a human (appear neutral). Moreover, some of the tweets are clearly mislabeled in the training data. Dataset contains repeating tweets.

First, data was preprocessed by ... **Omar part**

A several models were considered. First, the baseline was the GloVe [6] model, which is considers training embeddings on the co-occurence matrix. Unfortunately, the model did train quite slow on the dataset gave meaningless embeddings (closest words for a given word did not correspond to it in meaning). Therefore, it was not considered.

Secondly, a word count model was considered as a replacement. Here, a tweet is represented by a high-dimension sparse vector, each item corresponds to a number of occurences of this word in the tweet. This vector was further classified using several techniques, such as SVM and Neural Network. This method gave higher accuracy and allowed for tweaking its parameters with relatively small training time (a Google Cloud instance with 8 CPU was used)

III. Results

The following models were considered: *aba, caba*, the best one is *aba* This (does not) correspond to already conducted experiments [99, 98, 97]. Our contribution consists of running *method* with *xxx* modified with *yyy* and this does (not) give an improvement of 0.01231%

IV. Discussion

Our experiments lack *zzz*, which can be improved by doing also *ttt*

V. Summary

We have shown that it is possible to predict tweets using *aba* better than state-of-the-art.

Class	Row	Message	Comment
:(	171	<user> 5k i could cry i'm so unfit !	User tag
:(	99804	if i feel like this tomorrow i'm going to the er	Contraction, missed comma
:)	524	its the weekend ! ! <user> s coming home anddd <user> s baby shower , excited dd	Letter repetitions
:)	99615	<user> aren't you just an adorable granny <url>	URL tag
:)	443	<user> :d :d :d :d wish jocelyn had a twitter . #kudos for her too .	Hashtag, emoticons
:)	468	retweet if you was born in the 90 ' s ! #90's babies	Grammar mistake
:(	14	<user> i'm white . #aw	Appear neutral to human
:(	99594	<user> <user> <user> they're there tonight ! ! !	Does appear positive or negative to human

Table I
EXAMPLES OF TWEETS IN THE SMALL DATASET

REFERENCES

[1] Twitter

[2] Go, Alec, Lei Huang, and Richa Bhayani. "Twitter sentiment analysis." Entropy 17 (2009): 252.

[3] Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. "Twitter sentiment analysis: The good the bad and the omg!." Icwsm 11.538-541 (2011): 164.

[4] Tapan Sahni, Chinmay Chandak, Naveen Reddy, Manish Singh. "Efficient Twitter Sentiment Classification using Subjective Distant Supervision"

[5] Alec Go, Richa Bhayani, Lei Huang. "Twitter Sentiment Classification using Distant Supervision"

[6] Jeffrey Pennington, Richard Socher, Christopher D. Manning. "GloVe: Global Vectors for Word Representation"

[7] Competition and data downloads on kaggle.com