Abstract:

I\_ Introduction:

The role of sentiment analysis has grown significantly with the rapid proliferation of social networks. One of the most popular microblogging platforms is Twitter. It has grown constantly over the last decade and becoming a place for expression and debate between a wide range of people from students to politicians. This popularity has made Twitter one of the largest and most dynamic datasets of user generated content with approximately 200 million users post 400 million tweets per day [1]. In these posts users usually express either positive or negative opinions toward various subjects. Applying sentimental analysis allow to predict size of the markets and results of marketing campaigns.

The purpose of this project is to train an innovative model using a dataset of 2.5 million tweets labelled as positive or negative if they contained a positive or negative smiley and to perform binary classification.

II\_ Method:

A\_ Baseline models

We started our work with simple approaches such as linear and simple neural networks models.

1. Linear baseline

To implement the baseline, we trained a linear regression classifier with TF-IDF [2] vectorization (from scikit-learn library) to calculate the TF-IDF score for every word in a sentence relative to the entire dataset and put that information into a vector. Then we compare their similarity by using cosine similarity.

Term frequency works by looking at the frequency of a particular term relative to the document. Inverse document frequency looks at how common a word is amongst the corpus; it gives information about the relative rarity in the collection of documents.   
IDF minimizes the weighting of frequent terms while making infrequent periods have a higher impact.

1. Neural network and word embedding baseline

We implemented a simple bidirectional LSTM recurrent neural network with word embedding.

We used Word2Vec as embedding which is an algorithm that uses shallow 2-layer, not deep, neural networks to ingest a corpus and produce sets of vectors [3].

Using a large enough dataset can make robust estimates about word meaning based on their occurrences in the text and then group together vectors of similar words. We used the CBOW architecture from the gensim library, which tries to predict a target word from a list of context words.

B\_ Preprocessing

We explored the various existing techniques for text preprocessing in the context of sentiment analysis and decided to implement the ones that appear to be the most suitable for our project.

**removes rare words**, occurring less than "rare" times

**remove user and url** from tweets

**remove words consisting solely of digits**

**remove words that occur nearly equally frequent for positive and negative tweets**

**spell-check** words that would be discarded otherwise (are rare)

**delete duplicate** tweets.

**word segmentation to hashtags**.

**Handle emoji:** The classic positive ‘☺’ and negative ‘☹’ emojis have been removed from the dataset but others such as ‘:p’, ‘<3’, ‘=(’ are still in the dataset. We replace all these emojis by a token EMO\_POS or EMO\_NEG.

**Conjunction rules handling [4]:** We use conjunction rules to extract the precise meaning from a given sentence. If a sentence has a conjunction such as ‘but’, ‘although’ or ‘however’: the phrase before the conjunction will be cut off, and the rest of the sentence will be remained to represent the emotional polarity of the whole sentence. Using conjunction rules can makes the sentence more explicit and comprehensible.

C\_ Vocabulary based approach

We decided to create a lexicon based on the training data using the pipeline ‘Automatic Lexicon Generation’ described by Olga Kolchyna et al. [5].   
First, we assign a POS tag to each word and select only verbs, nouns, adverbs, adjectives with a length between 2 and 12. Then we count the occurrence of each one of these words in the positive and in the negative sub datasets. The positive polarity of each word is calculated according to the following formula: the number of occurrences in positive sentences divided by the number of all occurrences. We discard all the words with a polarity in the range [0.4; 0.6], since they do not help to classify the text as positive or negative.   
We map the scores of the words into the range [-1;1], given negative scores for the words with negative polarity and contrary for positive words.   
We then classify a sentence as positive or negative if the total score (adding the score of each word and 0 if the word is not in the vocabulary) is positive or negative.

This approach has been implemented with the goal of being combined with a ML model to form an ensemble.

D\_

E\_ Ensemble method

III\_ Experiment:

We trained the baseline models on the full dataset and tested it on Kaggle.

For the preprocessing we tested the impact of the various method implemented on the baseline models with the small dataset taking the same random seed and the same split to allow a good comparison.

IV\_ Results:

A\_ Baseline

1. Linear baseline

Trained on the full dataset without preprocessing : 83.9 on Kaggle

1. NN with Word2vec

Trained on the almost full dataset (90%) without preprocessing : 71.3 on Kaggle

Poor result maybe because : 2.5M tweet dataset can be considered as small by today’s standards 🡪 NN might overfit when training from scratch : explore various regularization strategies

B\_ Preprocessing :

C\_ Vocabulary based :  
Vocabulary from the full dataset, applied on the test data : 74.3 on Kaggle

V\_ Discussion & future direction:

Conclusion

References :

[1]  A. Ritter, S. Clark, Mausam, O. Etzioni, Named entity recognition in tweets: an exper- imental study, Proceedings of the Conference on Empirical Methods in Natural Lan- guage Processing, EMNLP'11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 1524–1534.

[2] Bruno Trstenjak, Sasa Mikac, Dzenana Donko,

KNN with TF-IDF based Framework for Text Categorization,

Procedia Engineering,

Volume 69,

2014,

Pages 1356-1364,

ISSN 1877-7058,

<https://doi.org/10.1016/j.proeng.2014.03.129>.

[3] IJCCS (Indonesian Journal of Computing and Cybernetics Systems)   
Vol.14, No.2, April 2020, pp. 169~178  
ISSN (print): 1978-1520, ISSN (online): 2460-7258  
DOI: 10.22146/ijccs.51743

[4] Maximum Entropy-based Sentiment Analysis of Online Product Reviews in Chinese, Hanqian Wu et al.

[5] Methodology for Twitter Sentiment Analysis, Olga Kolchyna et al.