Sentiment analysis and visualization of data from social networks using Machine learning algorithms

by

Aru Omarali

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

at the

NAZARBAYEV UNIVERSITY

Apr 2021

Author .....
Department of Computer Science
Apr 29, 2021

Certified by .....
Askar Boranbayev
Assistant Professor
Thesis Supervisor

Certified by .....
Mark Sterling
Assistant Professor
Thesis Supervisor

Accepted by .....
Vassilios D. Tourassis
Dean, School of Science and Technology

Sentiment analysis and visualization of data from social networks using Machine learning algorithms

by

Aru Omarali

Submitted to the Department of Computer Science
on Apr 29, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science

Abstract

In today's data-driven world, it is possible to gain access to significant amounts of data from different sources, and even share this data for various purposes. In modern everyday life, people use social networks extensively, reading tweets and posts, leaving comments, sharing their views on findings through comments and posts, or getting feedback from other users. As social networks are enhancing a source of abundant information flows, it is becoming difficult and time-consuming to filter the information. The correct analysis of information is important since the way we communicate and establish various kinds of relationships can heavily rely on correct interpretations.

This thesis aims to introduce the methods for sentiment analysis, investigating the application of the Machine Learning Approach for the sentiment classification problem by comparison of the Machine Learning and Statistical approaches, especially defining the importance of the Machine Learning approach for our purpose. Moreover, this research paper intends to explore the effectiveness of the pre-trained models over other approaches. Logistic Regression, Long Short-Term Memory, and BERT models will be demonstrated as methods of explaining this topic. And there will be an observation of what is the performance of the python libraries next to these methods. The analysis will show how the results are different and how the first approach outperforms the second one and will test whether ML algorithms show good performance and best results. Training experimental work will take place on the open-source dataset Sentiment140 extracted from Twitter.

Thesis Supervisor: Askar Boranbayev
Title: Assistant Professor

Thesis Supervisor: Mark Sterling
Title: Assistant Professor

Contents

1 Introduction **6**

1.1 Background 7

2 Related work **9**

2.1 Sentiment analysis 10

2.1.1 Data collection 11

2.1.2 Text preprocessing and preparation 11

2.1.3 Sentiment detection 11

2.1.4 Sentiment (feature) selection 11

2.1.5 Sentiment classification 12

3 Proposed methods. Algorithms and techniques **13**

3.0.1 Logistic regression 13

3.0.2 BERT (Bidirectional Encoder Representations from Transformers) 13

3.0.3 LSTM (Long-Short Term Memory) 14

3.0.4 TextBlob 14

3.0.5 Polyglot 15

4 Experiment **16**

4.0.1 Data collection 16

4.0.2 Text preprocessing 17

4.0.3 Extracting features 18

4.0.4 Model building 19
4.0.5 Training 19
4.0.6 Testing 22
4.0.7 Evaluation 23

5 Results & Discussion **25**
5.0.1 Result 25
5.0.2 Data visualization 27
5.0.3 Discussion 29

6 Conclusions **31**

List of Figures

3-1 Tokenization example 15

4-1 Detailed statistics of the data 17
4-2 Dataset 1 and Dataset 2 target class balance 17
4-3 Data before text preprocessing 18
4-4 Data after text preprocessing 18
4-5 Train accuracy history of LSTM model 20
4-6 Train accuracy history of BERT model 21
4-7 Negative sentiment detection 21
4-8 Positive sentiment detection 21
4-9 Randomly generated sentences 22

5-1 Evaluation results 26
5-2 Test cases with Logistic Regression model 26
5-3 Web interface. Positive and Neural sentiments 28
5-4 Web interface. Negative and Neural sentiments 28
5-5 Web interface. Neural and Positive sentiments 28
5-6 Web interface. Negative and Neural sentiments 28
5-7 Sentiment analysis graph 29

Chapter 1

Introduction

Social networks are taking an essential part of our life, we may sit there for hours not only for chatting with our friends or looking for publications of acquaintances but also for sharing news, getting media, leaving our own opinions, and contributing to problem solutions discussed there. The importance of data from social networks getting valuable so that we are used to browsing posts to see products, clothes, foods and look through comments of others to know about the real quality of the service, then we will think about buying or visit a certain place. This is the new role of social networks as a recommendation system. Online communication creates a relation between people, businesses and other parts of our society. The creation of such connections is directly dependent on the data flow in the social networks. Data flow can be described in terms of text, photo, audio, or video. These kinds of information streams may contain not only positive knowledge contributions but also negative materials. The issue behind this topic is that data analysis should be improved so that we can get valuable data. Our interactions will be meaningful if there is a positive environment in our digital society. However, this process is enough complicated because of the scaling of data, as we can get millions of messages every day. Despite this growth, plenty of modern solutions for data analysis are suggested, starting with a special social network analysis studies, continuing with software and tools. In most cases special tools are designed for marketing purposes, investigating the user flows through a social network, user behavior, and user actions with the information

provided. Depending on the observation results, they will enhance and straighten out their customer services or remove products that gain low rates. Furthermore, there may be a commitment to revise the text which received negative opinions from users as the textual information is assessed based on the attitude. Time to proceed with the text, study, and capturing the correct outcome boosts the demand for optimization of the whole mechanism of text analysis. One of the current innovative solutions is the usage of Machine Learning algorithms. They help to stimulate the automatization of human activity regarding sentiment analysis. Sentiment analysis is a Natural Language Processing task, explained as a process of defining emotions from words to understand the feeling or sentiment towards some concept. Emotions can be extracted from image, audio, video, and text [1, 2, 3]. Automation advances provide various optimization models that may facilitate the task performance by giving a quick responses. These improvements may be advantageous, for a process that requires the data to be utilized. This proposed project will investigate an experimental work on sentimental analysis by using a dataset from a social network, as people free to write everything there, and in most cases, they are not thinking about how proper texts they are writing.

1.1 Background

Machine Learning is a study of learning systems involving experience, covering the areas, such as information theory, recognition, data mining, and statistics [4]. The idea behind Machine Learning is to learn the data/tasks using neural networks of multiple layers and produce predictive results [5]. The learning can be supervised, unsupervised and reinforced by its learning style. Supervised and unsupervised learning depends on whether the data is labeled or not and the first learning type needs to get both input and output, while the second one requires only inputs to be processed. Supervised learning introduces a classifier that will be trained on labeled data and then will predict the future values. Reinforcement learning is about learning through interactions "feedback" with a dynamic environment [6]. Machine Learning can be de-

ductive and inductive depending on the learning information, first deals with existed evidence and assumes new knowledge from that, while the inductive learning type focuses on building instructions by extracting features and patterns from the dataset [7]. Natural Language Processing (NLP) is a field of Machine Learning about the understanding human language by a non-human scheme. NLP introduces syntactic and semantic analysis, sentiment analysis, there are rules and interpretations to work and manage text. Tokenization, part of speech tagging, sentiment and lemmatization techniques are about syntactic and defining the relationship between texts and word sense tend to be a semantic analysis. Sentiment analysis extracts semantics from the text. A recurrent neural network is a part of a neural network which makes connections between nodes and able to store and pass the data forward without feedback communication between layers. RNN can be one-to-one, one-to-many, many-to-one, and many-to-many. This type of network is used for speech recognition, video tagging, machine translation, and language modelling tasks.

Chapter 2

Related work

Recent works and findings around this topic related to the introduction and use of sentiment analysis algorithms in textual content, give detailed information on how the algorithm is implemented, what kind of neural network architecture is used. First of all, the sentiment analysis problem has in general two directions, Machine Learning and Lexicon - based, but modern study introduced a new approach, which combines a hybrid approach and a Deep Learning approach [8]. The Machine Learning approach is divided into supervised learning, which combines Decision Tree Classifiers, Probabilistic Classifiers, Linear Classifiers, Rule-based Classifiers, and unsupervised learning. The lexicon-based approach is described with dictionary-based and corpus-based methods [9]. Supervised learning has three popular methods as Naïve Bayes, Bayesian Network, and Support Vector Machine. The most used ones are Support Vector Machines and Naïve Bayes classifiers because they output good results in terms of performance. They work with labeled data, while unsupervised machine learning work with data containing no labels [10]. The main task behind using Machine Learning algorithms is to find features from context [11], while the statistical approach looks for the occurrence of a part of speeches, the appearance of polarities in a corpus [12]. The word polarity is then measured by the frequency of negative and positive words in the text [13]. If the word occurs more frequently among positive texts, then its polarity is positive. If it occurs more frequently among negative texts, then its polarity is negative. If it has equal frequencies, then it is a neutral

word [14]. The sentiment analysis process starts with the identification of the emotion model – sentiment identification. There are six emotions as anger, disgust, fear, happiness, sadness, and surprise. However, the emotional model differs depending on the language. Best practices show that a combination of different emotion models gives significant results [15]. Sentiment analysis practice with the Kazakh language is not properly studied, it is reported that baseline methods at most show 60 percent accuracy [16]. In most cases Kazakh language is used with Russian Language and analysis for dual languages is set by using Deep Recurrent Neural Networks. This type of method shows over 70 percent of accuracy, where applied two-lingual embedding of words [17]. As the Kazakh language is in the group of agglutinative languages, it can be described not only in terms of phonetic, but also morphological and syntactic features should be taken into consideration [18]. However, it is well formalized and an analyzer for morphological features can be set in the early stage of sentiment analysis [19]. Due to the complexity of natural languages, simple approaches are likely to fail since many facets of the language are not taken into account, as the presence of the negation. The following problems and limitations may arise during sentiment analysis: mistakes in words, unstructured text, and lack of labeled learning examples. Texts that contain mistakes in spelling make the word processing complicated and slows the recognition part. Notwithstanding these limitations, there is a subjectivity issue as the word may be for one person positive, for second negative or word is neural by its definition [20].

2.1 Sentiment analysis

Sentiment analysis combines the following steps to be done:

1. Data collection
2. Text preprocessing and preparation
3. Sentiment detection
4. Sentiment classification

5. Visualization of results

In this part, we will give a piece of detailed information about sentiment detection and classification methods that we selected for our purpose.

2.1.1 Data collection

Data collection is an important process in the data management area as there is a lack of labeled data. This is not only getting data, but it also covers cleaning, analyzing, and feature engineering topics [21].

2.1.2 Text preprocessing and preparation

Text preprocessor covers removing all non-letter characters, numbers, punctuation, stop-words, and transforming the text to lowercase. If there is empty data, we should also get rid of them. This process will help to improve the text quality for text classification tasks.

2.1.3 Sentiment detection

In this step, the subjectivity and objectivity of the words are defined. For example, the subjectivity of the sentence 'The film was not interesting' – 0.5, the subjectivity of the sentence 'Life goes on' – 0, and the sentence 'I feel sad' has a subjectivity of 1. Only the sentences that have subjectivity should be passed for the next step.

2.1.4 Sentiment (feature) selection

After removing not-relevant characters from the dataset, we go to the next step, called sentiment selection. This part is consolidated with Machine Learning algorithms, and its main task is to reduce the scale of the feature field [22]. In this step, the whole sentence is divided into separate words and this technique can be divided into supervised, semi-supervised, and unsupervised. The feature selection part will define features that may indicate that words in one category will differ from words

in the second [23] and techniques that used in this part may be narrowed due to the vocabulary resources [24]. That is why, each word is then transformed into a vector of numbers by using methods, like TF-IDF. Moreover, feature selection will influence the computational cost of algorithms.

2.1.5 Sentiment classification

Then goes sentiment classification, which can be divided into three levels: document level, sentence level, aspect level [25]. Document-level aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document a basic information unit. Sentence level aims to classify sentiment expressed in each sentence, first it checks whether the sentence is subjective or objective, if subjective, it will determine whether the sentence expresses positive or negative opinions. The aspect level aims to classify the sentiment concerning for the specific aspects of entities. On the output of the sentiment classification step, we get sentiment polarity, sentence/word classified as positive, negative or neutral.

Chapter 3

Proposed methods. Algorithms and techniques

3.0.1 Logistic regression

Logistic regression is a discriminative model, which does predictive analysis for the classification problems, when the target value is categorical, explaining the relationship between target-dependent value and other independent values. Logistic regression is considered to be a feature-based method and it calculates the odds, dividing the probability of an event by the probability of not event. It gets as input some value and on the output gives 0 or 1, probabilities are found by logistic function, outputting a binomial result. In this algorithm, Count Vectorizer is used for converting text input into feature output, token counts provided by scikit-learn.

3.0.2 BERT (Bidirectional Encoder Representations from Transformers)

BERT is a machine learning framework, meaning that reading a sequence of tokens, understanding, learning relations between them, and encoding the text based on meaning. It provides a pre-trained language model for 104 different languages, relying on two powerful technologies, a deep Transformer network and bidirectional.

BERT helps to conduct NLP tasks such as sentence-level classification, token level classification, and question answering. BERT aims to generate a language model, training by masking only 10-15 percent of tokens and predicting the next tokens. The first part is called Masked Language Modeling, the second is Next Sentence Prediction. Masked Language modeling aims to predict the masked tokens, while Next Sentence Prediction aims to predict the flow sequence of sentences. BERT base encoders consist of 12 encoders and BERT large 24. In this work, we will use a pre-trained model on the English language – "bert_base_cased".

3.0.3 LSTM (Long-Short Term Memory)

LSTM is one of the popular methods for text classification problems as a part of Recurrent Neural Networks, which can learn long-term dependencies. It contains memory blocks, each containing an input and output gate. They manage the input and output of the network by measuring the input and cell activations [26]. In this algorithm, Tokenizer is used for vectorizing a text by converting them into a sequence of integer values, provided by Keras, we can pass the number of most frequent words and get the unique words in train data. First, it creates a vocabulary with word frequencies, then transforms each word in the text to a sequence of integers. A sequential model is selected because we have one input and one output. Out LSTM model created with 128 neurons.

3.0.4 TextBlob

TextBlob is a text processing open-source python library described as a rule-based method, which focuses on pattern analyzer. It returns polarity and subjectivity values for a given text. It has the following features: tokenization, phase extraction, part-of-speech tagging, parsing, spelling corrections, sentiment analysis, and classification. In our practical work, TextBlob is used to test the sentence polarities of each word.

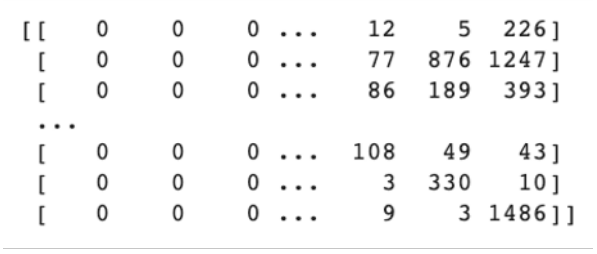

Figure 3-1: Tokenization example

3.0.5 Polyglot

This is an open-source python library, which characterizes itself as a unigram modeling approach and helps to perform NLP operations. It has polarity lexicons for 136 languages and in this project, it is used for testing the corpus-based statistical approach. Polyglot has the following opportunities: tokenization, part-of-speech tagging, language detection, transliteration morphological, and sentiment analysis. For a given sentence we can output each word's polarity and then the sentiment of the whole sentence determined by comparing positive and negative values.

Chapter 4

Experiment

In order to do our experiments, we used Keras (https://keras.io/), Scikit-learn (https://scikit-learn.org/), and PyTorch (https://pytorch.org/) python libraries. We set the number of epochs to be 10 and batch size – to be 128 and 32. To evaluate the performance of our models we used widely used assessment methods: Accuracy, Precision, Confusion matrix, Recall, and F score.

4.0.1 Data collection

For our experimental part, we took two open-source datasets in csv format, each containing tweets from different users about different topics in the English language. The first dataset consists of 1700 lines of data, containing index, sentence, sentiment, polarity, and sentiment type. The second dataset consists of 10 000 lines of data and is organized in terms of three columns: ItemId, Sentiment, Sentence. The length of the tweets in both datasets is mostly one or two-sentence. The first thing we needed to do is to process the text, prepare our data for the modeling and training part.

The first dataset contains 580 lines of negative sentences and 1120 lines of positive sentences. The second dataset contains 5800 lines of negative text and 4200 lines of positive text. The graphical information about target class distribution is given below:

Dataset	Samples	Train samples	Test samples	Classes	Target class balance
Twitter dataset 1	1700	1360	340	2	0 – 0.3420 1 – 0.6579
Twitter dataset 2	10 000	8 000	2 000	2	0 – 0.5812 1 – 0.4188

Figure 4-1: Detailed statistics of the data

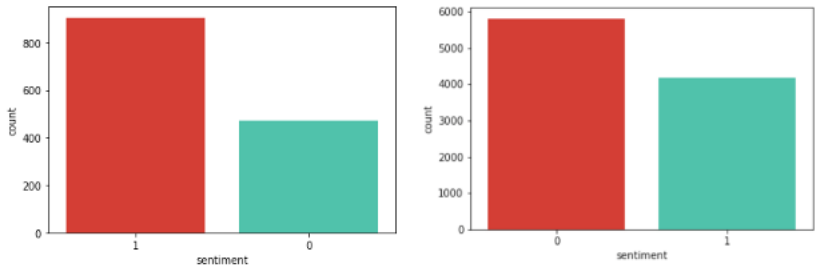

Figure 4-2: Dataset 1 and Dataset 2 target class balance

4.0.2 Text preprocessing

In this part, we used regular expressions to clean out data, did the following preprocessing steps which will help to remove the characters, as they do not contain any sentiment.

- Removing the number of white space characters followed by a comma
- Removing the literal question mark
- Removing the underscore
- Removing symbols
- Removing the one or more white spaces

- Removing one or more digits
- Removing punctuations
- Removing empty sentences (on non-empty sentiment value)
- Removing HTML tags
- Converting text to lower case

This is a dataset column before text preprocessing:

	sentence	sentiment
0	is so sad for my APL friend.....	0
1	I missed the New Moon trailer...	0
2	omg its already 7:30 :O	1
3	.. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was suposed 2 just get a crown put on (30mins)...	0
4	i think mi bf is cheating on me!!! T_T	0
5	or i just worry too much?	0
6	Juuuuuuuuuuuuuuussssst Chillin!!	1
7	Sunny Again Work Tomorrow :-I TV Tonight	0
8	handed in my uniform today . i miss you already	1
9	hmmmm.... i wonder how she my number @-)	1

Figure 4-3: Data before text preprocessing

After text preprocessing, we got the following cleaned text:

	sentence	sentiment
0	is so sad for my apl friend	0
1	i missed the new moon trailer	0
2	omg its already o	1
3	omgaga im sooo im gunna cry i ve been at this dentist since i was suposed just get a crown put on mins	0
4	i think mi bf is cheating on me t t	0
5	or i just worry too much	0
6	juuuuuuuuuuuuuuussssst chillin	1
7	sunny again work tomorrow tv tonight	0
8	handed in my uniform today i miss you already	1
9	hmmmm i wonder how she my number	1

Figure 4-4: Data after text preprocessing

4.0.3 Extracting features

Extracting features is one of the important parts of the model building process, as we need to find valuable data that contains more information to fit our model. This

part includes a method called TF-IDF (Term Frequency Inverse Document). TF-IDF counts the frequency of the word in a document, first TF is calculated by dividing the number of times a word appears in a document by total number of words in a document. The second term IDF measures the weighted importance of the word in a whole document. In BERT, word embeddings are extracted with the help of the Keras library. In Logistic regression algorithms it is done by Sklearn library CountVectorizer and in LSTM method features extracted by multiple layers using tokenizer functions. Feature selection techniques like removing objective sentences part of speech (POS) tagging (unsupervised learning), opinion words and phrases, finding negations can be also practiced [27].

4.0.4 Model building

LSTM model is four-layered Sequential models, containing an LSTM layer with 196 memory units. Categorical cross-entropy used as a loss function and Adam is used as an optimization function. To overcome the overfitting problem, we added a Dropout layer. The splitting proportion of the dataset into testing and training is 1/5. Logistic regression model constructed using sklearn linear model, defining 'lbfgs' (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) solver as an estimation, which guesses weights that will encourage to decrease the cost function, and other default parameters. Model fits the input that was converted into feature vectors. BERT model consists of 12 layers, 768 hidden sizes, and 12 attention masks. Adam is used as an optimizer and cross-entropy loss to fine-tuning the weight parameter. The model is called pre-trained because it is trained on data that not connected with the task.

4.0.5 Training

Training is a process of learning the data. We trained the models with two different numbers of data, first with 1700 lines and then 10 000 lines of data. The model training process consisted of 10 iterations, in order to prevent overfitting, and as we have not large data enough. After each iteration, we calculated the train, validation

accuracy, and loss. The validation split ratio for both LSTM and BERT was 0.5, while test size was in ratio 0.1.

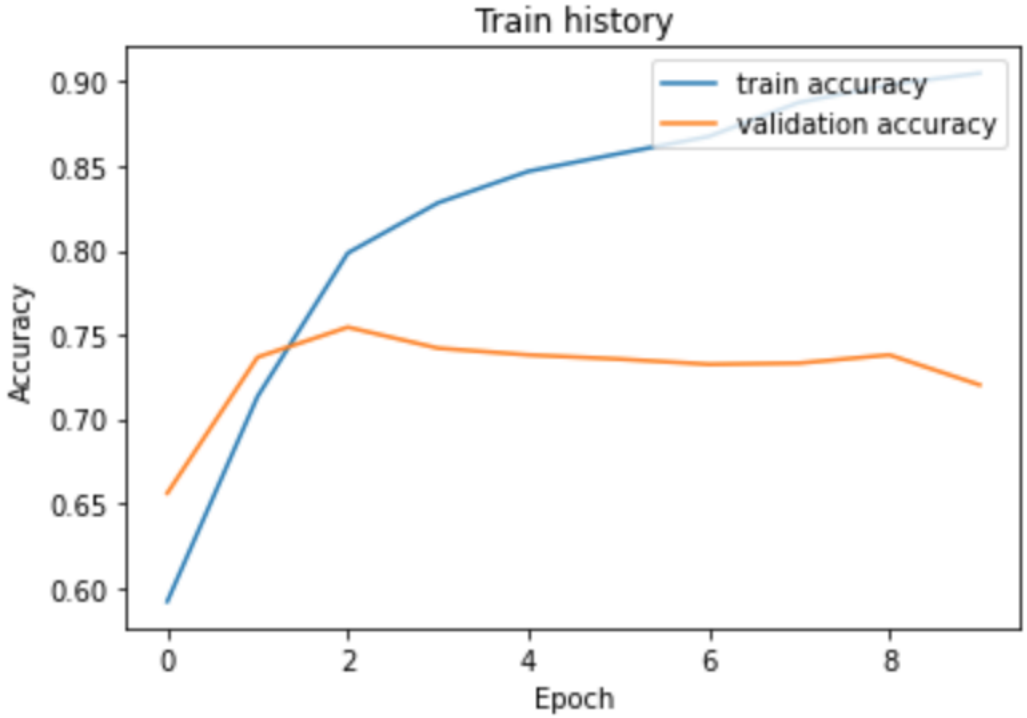

Figure 4-5: Train accuracy history of LSTM model

The accuracy of training the model after 10 iterations were the following: LSTM – 0.82, Logistic Regression – 0.77, BERT – 0.81. Working with the Polyglot and TextBlob library, they do not require the training part, so we took the examples from the dataset and passed them through our code. One thing to be noted is that some language resources were not available in Polyglot and we had difficulties with sentiment detection.

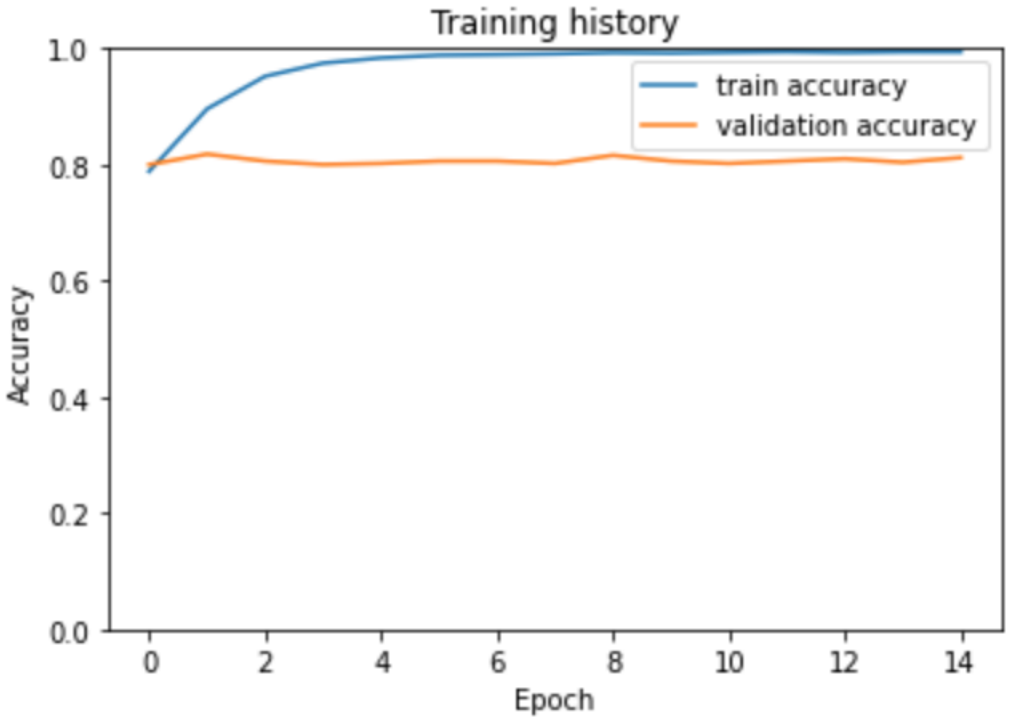

Figure 4-6: Train accuracy history of BERT model

Below examples of detecting "Negative" and "Positive" sentiment:

is so sad for my apl friend	
is	0
so	0
sad	-1
for	0
my	0
apl	0
friend	0
Negative	

i missed the new moon trailer	
i	0
missed	-1
the	0
new	0
moon	0
trailer	0
Negative	

or i just worry too much	
or	0
i	0
just	0
worry	-1
too	0
much	0
Negative	

i think mi bf is cheating on me t t	
i	0
think	0
mi	0
bf	0
is	0
cheating	-1
on	0
me	0
t	0
t	0
Negative	

Figure 4-7: Negative sentiment detection

sunny again work tomorrow tv tonight	
sunny	0
again	0
work	1
tomorrow	0
tv	0
tonight	0
Positive	

hmmmm i wonder how she my number	
hmmmm	0
i	0
wonder	1
how	0
she	0
my	0
number	0
Positive	

it this is the way i feel right now	
it	0
this	0
is	0
the	0
way	0
i	0
feel	0
right	1
now	0
Positive	

Figure 4-8: Positive sentiment detection

4.0.6 Testing

We generated 10 random sentences using the Document Generator library (https://pypi.org/project/esse generators/) to test our models, how they predict their polarities and whether they show different results.

```
1 Persons. The the lower-density surface zone is known as the length and movement
2 Density zones: two existing customs unions: Mercosur and the Mediterranean trade.
3 Owls, Carolina with sporadic rainfall while parts of
4 Midtown, and a move into the ground in what is right. Evil or bad
5 Italian sausage. than a place name.
6 Physician Asaph downdrafts within the Boreal Kingdom and Empire), and the Arabian
7 XML dialect. entrance to
8 And testified colloquial use of effect size statistics, rather than the speed of light in
9 English languages. explain properties of the
10 Ten floors has rather warm summers, with a salad
```

Figure 4-9: Randomly generated sentences

We passed these sentences to our four methods, labeled the output, if the assumed value is positive, then 1, if it is negative then 0. For the Polyglot case, we also highlighted the neural values.

1. Persons. The the lower-density surface zone is known as the length and movement [BERT – 0, LSTM – 1, Logistic Regression – 1, Polyglot – 1 (neural)]

2. Density zones: two existing customs unions: Mercosur and the Mediterranean trade. [BERT – 0, LSTM – 1, Logistic Regression – 0, Polyglot – 1 (neural)]

3. Owls, Carolina with sporadic rainfall while parts of [BERT – 0, LSTM – 0, Logistic Regression – 1, Polyglot – 0]

4. Midtown, and a move into the ground in what is right. Evil or bad [BERT – 0, LSTM – 0, Logistic Regression – 0, Polyglot - 1 (neural)]

5. Italian sausage. than a place name. [BERT – 0, LSTM – 1, Logistic Regression – 1, Polyglot – 1 (neural)]

6. Physician Asaph downdrafts within the Boreal Kingdom and Empire), and the Arabian. [BERT – 0, LSTM – 1, Logistic Regression – 0, Polyglot – 1 (neural)]

7. XML dialect. entrance to. [BERT – 1, LSTM – 1, Logistic Regression – 0, Polyglot – 1 (neural)]

8. And testified colloquial use of effect size statistics, rather than the speed of light in [BERT – 0, LSTM – 1, Logistic Regression – 0, Polyglot – 1 (neural)]

9. English languages. explain properties of the [BERT – 1, LSTM – 0, Logistic Regression – 0, Polyglot – 1 (neural)]

10. Ten floors has rather warm summers, with a salad [BERT – 1, LSTM – 0, Logistic Regression – 0, Polyglot – 1]

4.0.7 Evaluation

To test whether our model works appropriately, we need to make an evaluation. In this work, we looked for the most commonly used evaluation metrics:

- Accuracy (TP+TN)/Total – meaning the proportions of correct predictions

- Precision TP/(TP+FP) - meaning the proportion of points that the model classifes as positives are actually positives, how many values are predicted correctly.

- Confusion matrix represents a table of four value combinations about predicted and real values, used for evaluation of the classification model. The predicted values are described in positive and negative, while actual values are true and false.

    1. True Positive (TP): model predicted the actual value correctly and it shows a positive result

    2. True Negative (TN): model predicted the actual value correctly and it shows a negative result

    3. False Positive (FP): model predicted the actual value to be positive and it is incorrect

    4. False Negative (FN): model predicted the actual value to be negative and it is incorrect

- Recall (TP/(TP+FN)) - meaning the proportion of actual positives that are correctly classified by the model, how many actual values predicted correctly
- F score – helps to compare models, simultaneously taking into account recall and precision, calculated as the harmonic mean of precision and recall.

Chapter 5

Results & Discussion

5.0.1 Result

We applied two datasets to the proposed LSTM, Logistic Regression, BERT models, and library Polyglot. We did Machine Learning based and statistical approach. The table below shows the experimental results of ML approaches. We established that the result of experimental work with a small dataset does not achieve the results that a large dataset got, so in this part, only the result of working with large dataset will be discussed. In the Figure 5-1, LSTM and BERT models show the best accuracy, showing about 0.8, however, BERT pre-trained model shows better performance in terms of precision, recall and F score over two algorithms.

We ran 5 positive sentences and 5 negative sentences from the dataset for the Logistic Regression model to see predicted and expected sentiment values. Then we obtained the following results in Figure 5-2.

By summarizing achieved results, we can admit that LSTM presented a more compatible model learning rate, showing an accuracy of 0.82. Despite a well-trained model, LSTM failed in terms of precision and recall metrics. BERT model demonstrated consistent output for accuracy and recall, showing a value of 0.81. Talking about the Logistic regression model, we conclude that there was lagging behind the accuracy of the training history than other models, showing - 0.77. We looked to the results of others that did sentiment analysis with this open-source

Evaluation metrics	LSTM	Logistic regression	BERT
Accuracy	0.82	0.77	0.81
Precision	0.74	0.76	Negative – 0.7 Positive – 0.61
Recall	0.74	0.64	Positive - 0.81 Negative – 0.79
F score	0.74	0.67	0.82

Figure 5-1: Evaluation results

	Sentence	Predictions	Expect	Results
0	feeling strangely fine now i m gonna go listen to some semisonic to celebrate	False	True	False Negative
1	handed in my uniform today i miss you already	False	True	False Negative
2	you re the only one who can see this cause no one else is following me this is for you because you re pretty awesome	False	True	False Negative
3	uploading pictures on friendster	False	True	False Negative
4	thanks to all the haters up in my face all day	False	True	False Negative
5	this weekend has sucked so far	False	False	False Positive
6	just worry too much	False	False	False Positive
7	i missed the new moon trailer	False	False	False Positive
8	is so sad for my apl friend	False	False	False Positive
9	isnt showing in australia any more	False	False	False Positive

Figure 5-2: Test cases with Logistic Regression model

dataset and found that their Logistic regression model hit accuracy of 0.82 [Kritika Rupauliha's solution from Github (https://github.com/rkritika1508/Sentiment-Analysis/blob/master/Fifth.ipynb)]. Talking about the prediction of the sentiment of the ten generated sentences, we observed that there was only one case, when we got the same result, in other cases, they differed. Sometimes we observed that BERT with LSTM, BERT with Logistic regression, or LSTM with Logistic regression predicted in the same way. In most cases, prediction by using Polyglot displayed that the text is Neural.

Talking about the models themselves, we considered that the LSTM model is one

of the widely used and studied methods, as over 3000 papers were found only from one source (https://paperswithcode.com/), and this rate is multiple times bigger than other method related papers. Papers about the application of LSTM for sentiment analysis task are in the second place after time series papers. The popularity of the LSTM model can be described by fact that it is simple to implement. The logistic regression model can be advantageous if there is low dimensional data and their feature are linearly separable, but it requires a large dataset to get better results. BERT pre-trained model gives a better result as it is already trained and has calculated weights, but it requires more computational power and time. Polyglot and TextBlob does not need prior training, and we can get the result quickly, because of execution time, but it has no learning competence.

5.0.2 Data visualization

The operation of the sentiment analysis may go through various applications where the user can easily detect what does the word expresses. One of the approaches can be web applications. Part of the hands-on experience was creating a simple web application that can demonstrate the idea behind this approach. Web application based on TextBlob and Polyglot python libraries, LSTM, and BERT approach for sentiment analysis. LSTM and BERT models were selected because of their performance while the training and testing phases. Those trained models saved and were imported to web projects and used for sentiment prediction. The interface of the application was written in JavaScript and used the Flask framework for connecting our python libraries for a web application. One thing that should be mentioned is that TextBlob and Polyglot may output "Neural" values too. User enters word or sentences, then clicks the button "Define sentiment" and get results for every block of methods. We can observe that one sentence can be from one perspective "Neural" and another perspective "Positive".

In the web application, we allowed loading data from the file and see the sentiment chart, count how much data tend to be positive and negative. Sentiment prediction is done by the LSTM algorithm. On the right side, we can see the filtered list of data

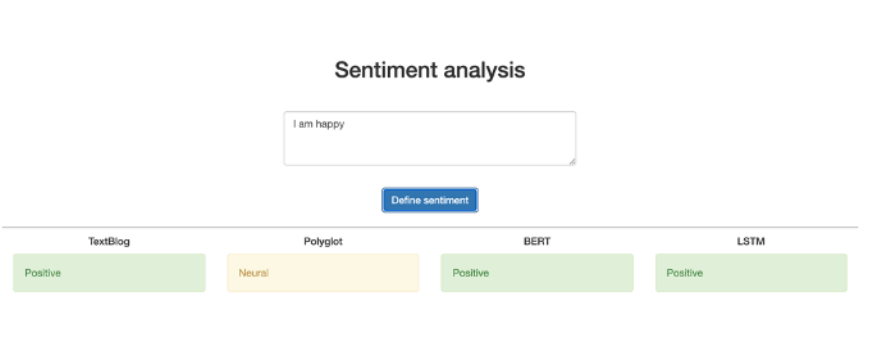

Figure 5-3: Web interface. Positive and Neural sentiments

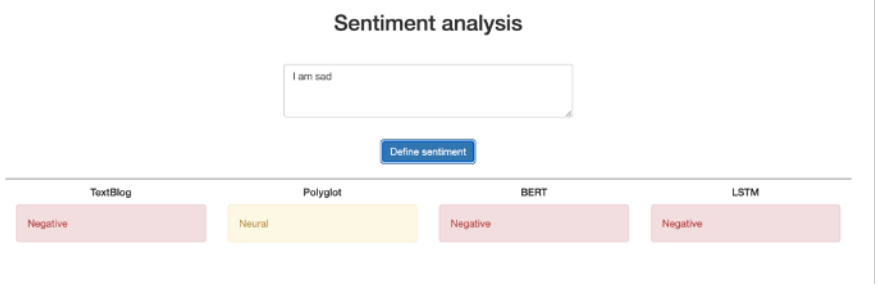

Figure 5-4: Web interface. Negative and Neural sentiments

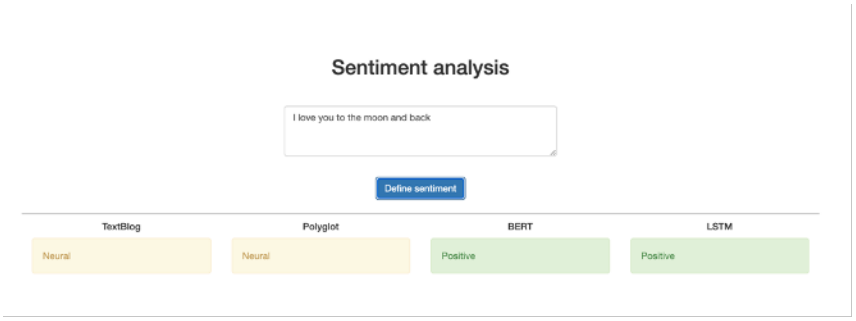

Figure 5-5: Web interface. Neural and Positive sentiments

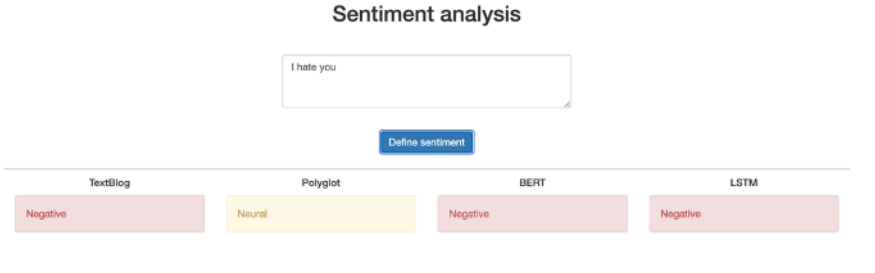

Figure 5-6: Web interface. Negative and Neural sentiments

by sentiment. This might be an example of how you can filter data by sentiment. For example, a user uploads a post on Instagram receives comments, and can see how others react to that post. The post can be an advertisement, recommendation, or sale of a product. With the help of this analysis, it will be possible to collect feedback and delete negative comments.

Figure 5-7: Sentiment analysis graph

5.0.3 Discussion

In this paper, we observed how sentiment analysis work and how different technique can be applied for this NLP problem. Sentiment prediction from different methods represented different values.

The first reason may be that our class distribution was imbalanced, that is why accuracy may not found appropriate. The first training set class balance was Negative – 0.34201, Positive – 0.6579, while the second dataset class distribution was following, Negative – 0.5812 and Positive – 0.4188. In order to solve this issue, we should truncate and pad the input sequences, use the class weighted loss function, and up sample our class sharing.

The possible second reason is that we targeted only Positive and Negative classes, not taking into consideration Neural class sentences. This may also alter the prediction sentiment if the actual value of the test sentence is Neural, as there is no option for forecasting of third class. This is a shred of evidence, when we passed text to ML

algorithms and got positive or negative sentiment, however Polyglot guess that the target class of the text is neural.

The third reason may be that we did not consider sarcastic sentences, negations and we did not perform the stemming process. Finding negations from sentences and analyzing them during sentiment classification is still an open question in the sentiment analysis domain. The stemming process can be done using Natural Language Tool Kit, meaning this preprocessing is replacing one rooted word in different words with a root word. For example, the words waiting, waits, waited are replaced with the word 'wait'.

The final reason may be the domain of the dataset that we used to test our models. 10 randomly generated sentences were not especially from social networks, it may cover different subjects, even difficult terms. A random sentence generator was chosen because attempts to connect Twitter API were unsuccessful. Our models were trained only on Twitter reports, that is why it is obvious that they may face confusion.

In the interest of increasing the progress of proposed models, we can do the following operations:

- Removing words that do not contain sentiment. For example, nouns and pronouns.
- Tuning the hyperparameters. For example, using popular Grid Search.
- Scaling the feature and normalization, they will decrease the computational cost. For example, BERT needed more computational power for training the data than other models.
- Handling Part of Speech and Point-Wise mutual information. For example, there is a hypothesis that adjectives have more value in the sentence sentiment rather than adjectives with adverbs.

Chapter 6

Conclusions

In this paper, we tried to introduce methods for sentiment analysis task by experimental work and by comparing their performance investigated the hypothesis which approach shows appropriate results. The main contribution of this paper was a revision of three algorithms, which have different individual objectives but applied to the one sentiment analysis task. Furthermore, we used the Polyglot and TextBlob libraries to see how it flies with this task, how we can output the sentiment of the whole sentence knowing separate words sentiment. We come up with the idea that pre-trained models and ready libraries give more precise results. To raise the power Machine Learning approach, we should build the model so that we took into consideration all the features of the selected language, improve preprocessing and do experiments on large datasets.

Bibliography

[1] Chaturvedi I. Cambria E. Hussain Poria, S. A convolutional mkl based multimodal emotion recognition and sentiment analysis. *IEEE International Conference on Data Mining*, 41(7837868):439–448, 2017. This is a full ARTICLE entry.

[2] Kalaiselvi Geetha M Arunnehru, J. Automatic human emotion recognition in surveil- lance video. *Studies in Computational Intelligence*, 660:321–342, 2017. This is a full ARTICLE entry.

[3] Ho A.T.S. Cheheb I. Al-Maadeed N. Al-Maadeed S. Bouridane A Jiang, R. Emotion recognition from scrambled facial images via many graph embedding. *Pattern Recognition*, 64:245–251. This is a full ARTICLE entry.

[4] Guoru Ding Yuhua Xu-Shuo Feng Junfei Qiu, Qihui. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 64, 2016. This is a full ARTICLE entry.

[5] Bing Liu Lei Zhang, Shuai Wang. Deep learning for sentiment analysis: A survey. This is a full ARTICLE entry.

[6] Andrew W.Moore Leslie Pack Kaelbing, Michael L.Littman. Reinforcement learning: A survey. *EJournal of Artificial Intelligence Reseach*, 4:232–285, 1996. This is a full ARTICLE entry.

[7] Omprakash Sangwan Yogesh Singh, Pradeep Kumar Bhatia. A review of studies on machine learning techniques. *International Journal of Computer Science and Security*, 1. This is a full ARTICLE entry.

[8] Li Wei Kun Guo Yong Shi, Luyao Zhu. Survey on classic and latest textual sentiment analysis articles and techniques. *International Journal of Information and Decision Making*, 2019. This is a full ARTICLE entry.

[9] Hassan A. Korashy H Medhat, W. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 4(5):1093–1113. This is a full ARTICLE entry.

[10] Ravi V Ravi K. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, (89):14–46. This is a full ARTICLE entry.

[11] Lee L Pang B. Opinion mining and sentiment analysis. *Foundations and Trends in Information retrieval*, (2):1–135. This is a full ARTICLE entry.

[12] Klenner M Fahrni A. Old wine or warm beer: target-specific sentiment analysis of adjectives. *Proceedings of the symposium on affective language in human and machine, AISB*, (2):1–135, 2008. This is a full ARTICLE entry.

[13] Carroll J Read J. Weakly supervised techniques for domain-independent sentiment classification. *Proceeding of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion*, page 45–52, 2009. This is a full ARTICLE entry.

[14] Ahmed Hassanb Hoda Korashyb Walaa Medhata. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5:1093–1113, 2014. This is a full ARTICLE entry.

[15] Mendoza M. Poblete B Bravo-Marquez, F. Combining strength, emotions and polarities for boosting twitter sentiment analysis. *Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining*. This is a full ARTICLE entry.

[16] Altynbek A Banu Yergesh, Gulmira Bekmanova. Sentiment analysis of kazakh text and their polarity. 2019. This is a full ARTICLE entry.

[17] Ivanov V.V Abdullin, Y.B. Deep learning model for bilingual sentiment classification of short texts. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 17:129–136. This is a full ARTICLE entry.

[18] Ivanov V.V Abdullin, Y.B. Deep learning model for bilingual sentiment classification of short texts. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 17:129–136. This is a full ARTICLE entry.

[19] Mukanova A. Sharipbay A. Bekmanova G. Razakhova B Yergesh, B. Semantic hypergraph based representation of nouns in the kazakh language. *Computacion y Sistemas*, 3(18):627–635. This is a full ARTICLE entry.

[20] Chaturvedi I. Distunguishing between facts and opinions for sentiment analysis: Survey and challenges. *Information Fusion*, (44):65–77. This is a full ARTICLE entry.

[21] Steven Euijong Whang Yuji Roh, Geon Heo. A survey on data collection for machine learning. *A Big Data - AI Integration Perspective*. This is a full ARTICLE entry.

[22] Savoy J Kummer O. Feature selection in sentiment analysis. 2000. This is a full ARTICLE entry.

[23] Lingfeng Niub Jianyu Miaoa, c. A survey on feature selection. *Information Technology and Quantitative Management*, 2016. This is a full ARTICLE entry.

[24] Saravanakumar Kandasamy Dishi Jain, Bitra Harsha Vardhan. Sentiment analysis of product reviews – a survey. *International Journal of Scientific Technology Research*, 8, 2019. This is a full ARTICLE entry.

[25] Liu B. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, Morgan Claypool Publishers*, 2012. This is a full ARTICLE entry.

[26] Franc¸oise Beaufays Has¸im Sak, Andrew Senior. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. This is a full ARTICLE entry.

[27] Zhai Cheng Xiang Aggarwal Charu C. Mining text data. *Springer New York Dordrecht Heidelberg London: © Springer Science+Business Media*, 12, 2012. This is a full ARTICLE entry.