Classification of Data based on Sentiment of Text using Machine Learning & Deep Learning

(High Integrity Systems, Master's Thesis)

Presented by: Rozana Alam

Advisor: Prof. Dr. rer.nat. Matthias F. Wagner

Co-advisor: Prof. Dr. Eicke Godehardt

Department of Computer Science and Engineering,
Frankfurt University of Applied Sciences, Frankfurt(Germany)

Outline

- Introduction 3
- Related Work 4
- Business Knowledge and proposed solution 6
- Theoretical & Technical Overview 8
- Data Collection & Analysis 19
- Implementation
- Results 30
- Conclusion 31

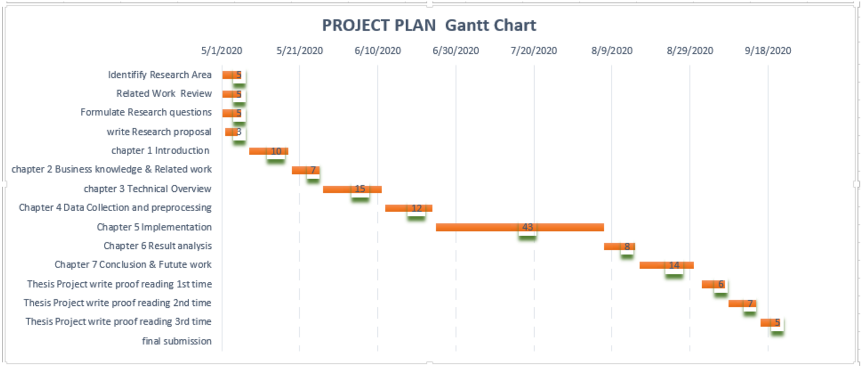
Introduction

- Overview
- Motivation
- Related Work
- Project plan

Related work

Result analysis of Related works		
Model Type	**Model name**	**Accuracy**
Machine Learning	Naïve Bayes	88%
	Support Vector machine	81%
	Random Forest	63%
Deep Learning	Deep Pyramid CNN (Amazon 2)	94.68%
	BLSTM-(SST2)	89.5%
Bert model	Bert base(IMBD)	95.63%
Classification	eight category	59%

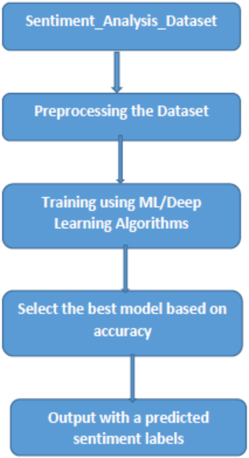
Project Plan



Business Knowledge and proposed solution

- Natural Language processing(NLP)
- Multi-class Classification
- Sentiment Analysis
- Applications of Sentiment Analysis
- Problem Statement
- Proposed Solution
- Objectives
- Process model of sentiment analysis

Process model of sentiment analysis



Sentiment_Analysis_Dataset

Preprocessing the Dataset

Training using ML/Deep Learning Algorithms

Select the best model based on accuracy

Output with a predicted sentiment labels

AI, Machine Learning and Deep Learning

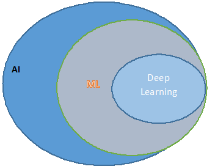
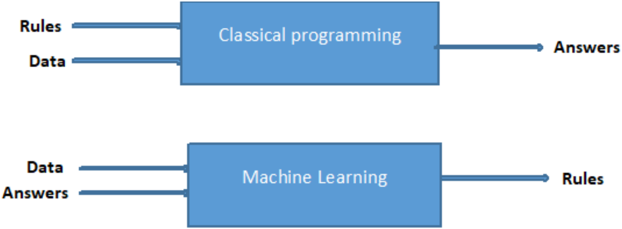
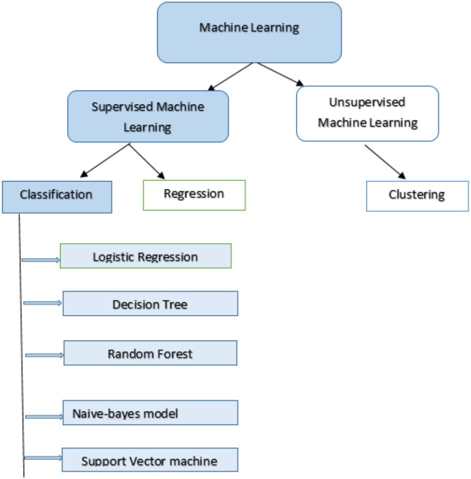


Figure: AI, ML, and Deep Learning relationship



Selected Methods for the experiment on our Data

Text classification with Machine Learning models



Selected Methods for the experiment on our Data

Text classification with Machine Learning models

- Naïve Bayes classifier 11
- Logistic Regression 12
- Decision Tree 13

Text classification with Deep Learning models

- General Neural Network 14
- Convolutional Neural Network 15
- Long Sort-term Memory(LSTM) 16
- Bert_base_uncased_model 17

Naïve Bayes Theorem ML model

Naïve Bayes:

$$P(A/B) = (P(B/A)P(A))/(P(B)) \tag{1}$$

Where P(A/B) and P(B/A) are conditional probabilities.

Multinomial NB model:

$$P(c/d) \ \alpha \prod_{1 \leq_k \leq_n d} P(c)P(tk/c) \tag{2}$$

Logistic Regression

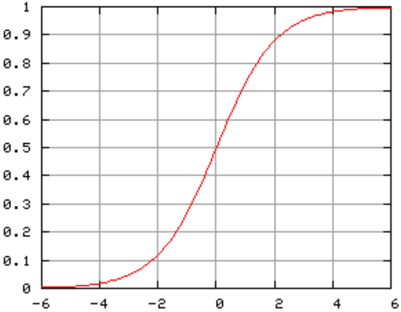


Figure: Logistic Curve, The values of y cannot be less than 0 or greater than 1

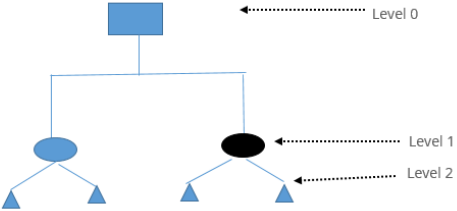
Decision Tree

■ Root node: is the beginning of the tree

⬮ Internal Node: Splits into further

▲ Leaf node: is a node that no longer splits

| Branches: is the link between nodes



Simple Neural Network

$$\sum_{i=1}^{3} x_i w_i = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 \qquad (3)$$

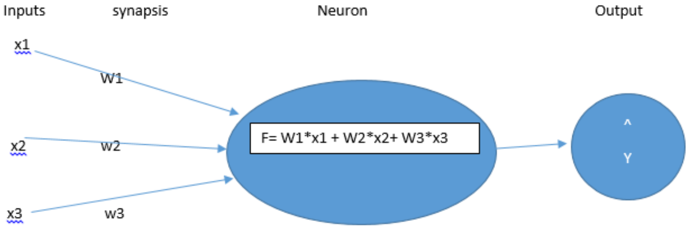


Figure: Perceptron

Convolutional Neural Network

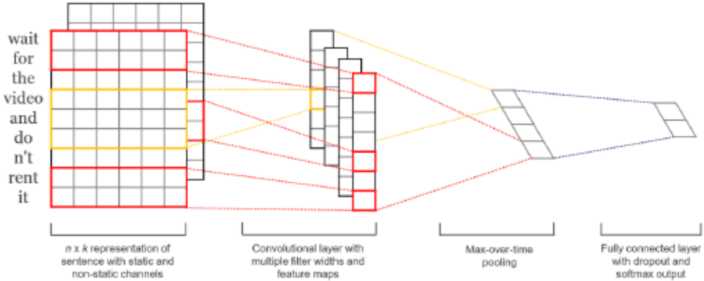

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure: The architecture of a sample CNN model for text classification

Long Short-Term Memory(LSTM)

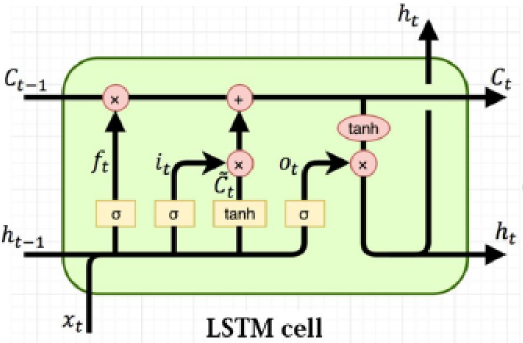


Figure: Structure of the LSTM cell

Bert_base_uncased model

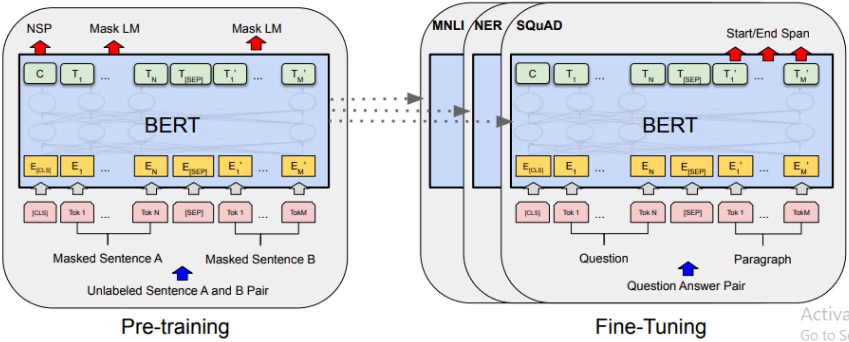
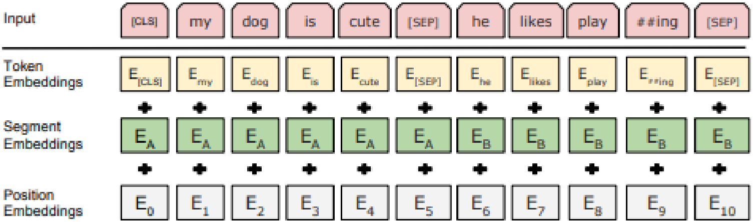


Figure: BERT model

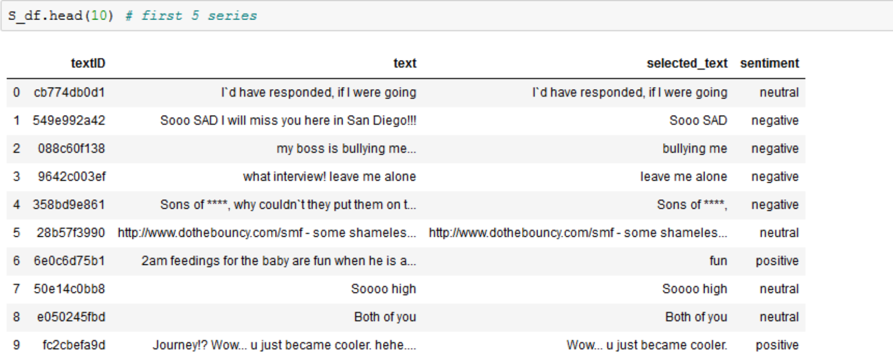
BERT input representation



Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\#\#ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

Figure: BERT input representation

Data Collection And Data Investigation



```
S_df.head(10) # first 5 series
```

	textID	text	selected_text	sentiment
0	cb774db0d1	I`d have responded, if I were going	I`d have responded, if I were going	neutral
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
2	088c60f138	my boss is bullying me...	bullying me	negative
3	9642c003ef	what interview! leave me alone	leave me alone	negative
4	358bd9e861	Sons of ****, why couldn`t they put them on t...	Sons of ****,	negative
5	28b57f3990	http://www.dothebouncy.com/smf - some shameles...	http://www.dothebouncy.com/smf - some shameles...	neutral
6	6e0c6d75b1	2am feedings for the baby are fun when he is a...	fun	positive
7	50e14c0bb8	Soooo high	Soooo high	neutral
8	e050245fbd	Both of you	Both of you	neutral
9	fc2cbefa9d	Journey!? Wow... u just became cooler. hehe....	Wow... u just became cooler.	positive

Figure: head lines of S_df

Data visualization of 'S_df' DataFrame

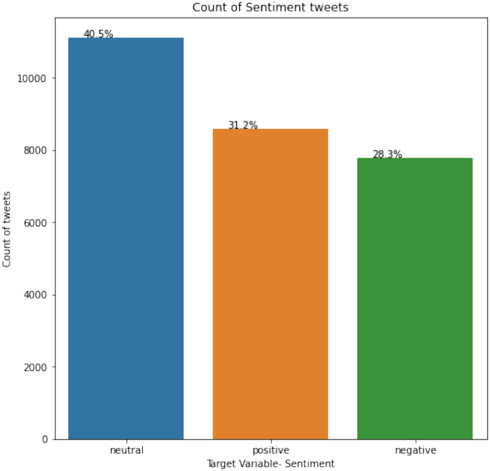


Figure: S_df visualization

'Selected_text' column WordCloud Visulalization

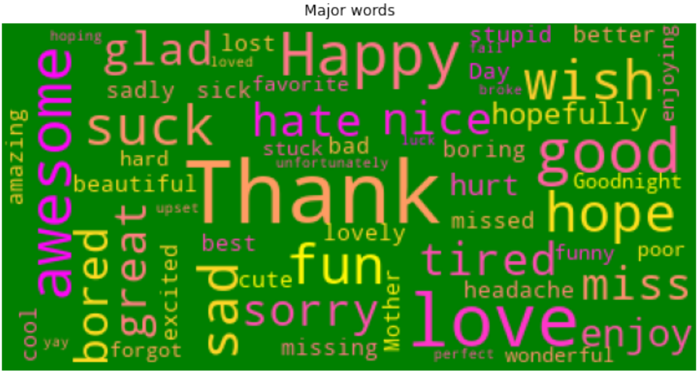
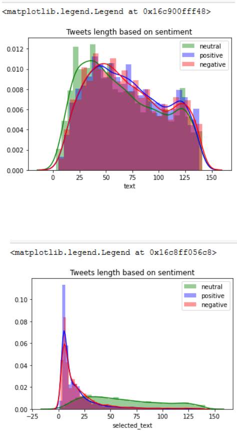


Figure: 'Selected_text' column WordCloud Visulalization

Comparison of 'selected_text' & 'text' column

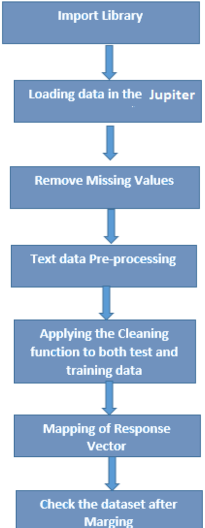


```
<matplotlib.legend.Legend at 0x16c900fff48>
```

```
<matplotlib.legend.Legend at 0x16c8ff056c8>
```

Figure: Value counts for 'selected_text' and 'text' column

Data pre-processing for ML models



Import Library

Loading data in the Jupiter

Remove Missing Values

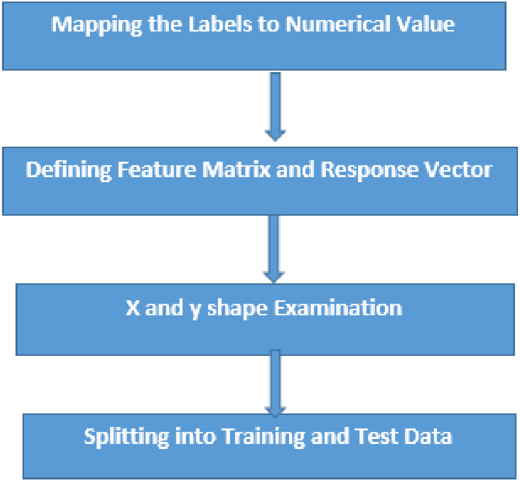
Text data Pre-processing

Applying the Cleaning function to both test and training data

Mapping of Response Vector

Check the dataset after Marging

Defining Training and Test Data for ML models



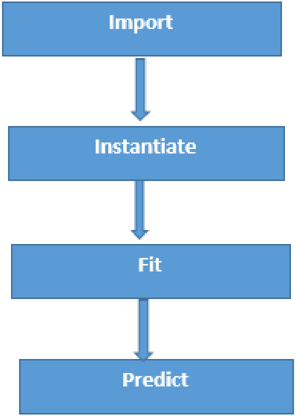
Mapping the Labels to Numerical Value

Defining Feature Matrix and Response Vector

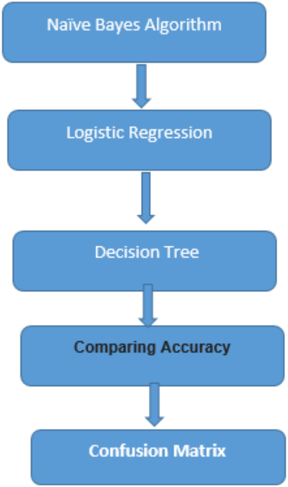
X and y shape Examination

Splitting into Training and Test Data

Flow of Feature Engineering steps



Steps for best Machine Learning model



Naïve Bayes Algorithm

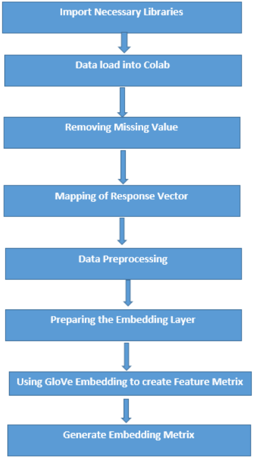
Logistic Regression

Decision Tree

Comparing Accuracy

Confusion Matrix

Data pre-processing for Simple NN, CNN and RNN



Import Necessary Libraries

Data load into Colab

Removing Missing Value

Mapping of Response Vector

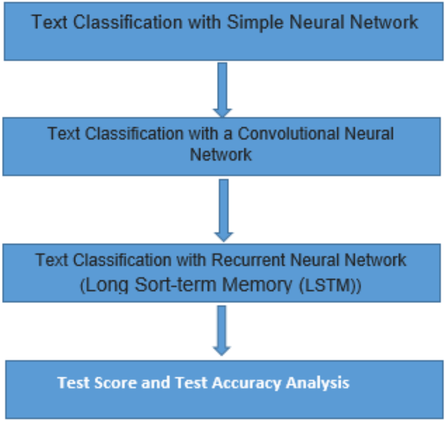
Data Preprocessing

Preparing the Embedding Layer

Using GloVe Embedding to create Feature Metrix

Generate Embedding Metrix

Flow of Deep Learning Best model



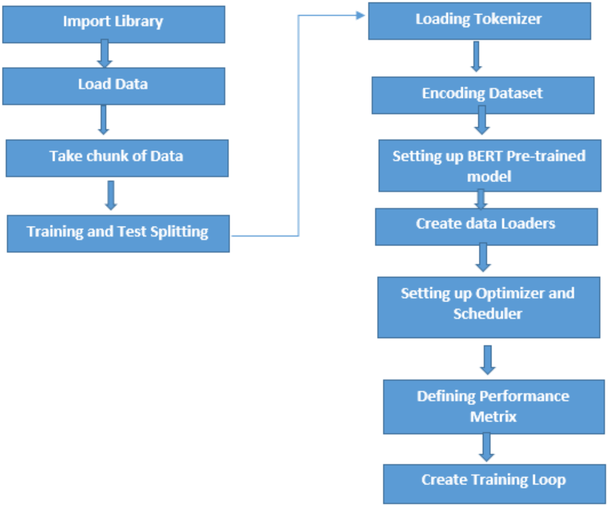
Text Classification with Simple Neural Network

Text Classification with a Convolutional Neural Network

Text Classification with Recurrent Neural Network (Long Sort-term Memory (LSTM))

Test Score and Test Accuracy Analysis

Flowchart of BERT model



Import Library

Load Data

Take chunk of Data

Training and Test Splitting

Loading Tokenizer

Encoding Dataset

Setting up BERT Pre-trained model

Create data Loaders

Setting up Optimizer and Scheduler

Defining Performance Metrix

Create Training Loop

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Result analysis of all the model

Model Type	Model name	Accuracy
ML model	Naïve Bayes	78.31%
	Logistic Regression	78.85%
	Decision Tree	75.96%
DL Model	Simple NN	48.66%
	CNN	49%
	LSTM	32.50%
Bert model	Bert_Base_Fine_Tuning	88.56%
Sentiment for Bert model	Neutral	86.028%
	negative	88.724%
	positive	90.29%

Conclusion

- Limitations
- Problem Faced
- Future Plan

Thank You

Questions??

FRANKFURT UNIVERSITY OF APPLIED SCIENCES