Sentiment Analysis on IMDB Movie Reviews

Evaluation and comparison of multi machine learning approaches to predict a binary sentiment of a movie review

Roman Studer   
BSc Data Science Student,

FHNW Brugg-Windisch, Switzerland roman.studer1@students.fhnw.ch

Lukas Gehrig   
BSc Data Science Student,

FHNW Brugg-Windisch, Switzerland lukas.gehrig@students.fhnw.ch

Vincenzo Timmel   
BSc Data Science Student,

FHNW Brugg-Windisch, Switzerland vincenzo.timmel @students.fhnw.ch

*Abstract*—

Keywords—component, formatting, style, styling, insert (key words)

# Introduction

## Goal of the analysis

This text classification task is focused on the evaluation and comparison of several natural language processing methods for sentiment classification. For this purpose, we use a dataset with 50 thousand film reviews from the IMDB[[1]](#footnote-1) platform. A dataset that is often used for benchmarks in the NLP environment. The dataset is suitable for a binary sentiment classification as it is labeled with the binary feature "sentiment" which either takes on the value "positive", if the sentiment of the review is positive, or "negative" if the sentiment is negative [1]. An example record of the dataset would be:

Table 1, Example of a review in the IMDB dataset

|  |  |  |
| --- | --- | --- |
| **Index** | **Review** | **Sentiment** |
| 124 | “I like this movie” | positive |

## Process

At the start of the process, functions were defined that allow a uniform pre-processing of the data. This ensures that all employed models receive the same data as input. In chapter II we describe the pre-processing in more detail. With the pre-processed data several models were trained and their accuracy on train and test set was recorded. In a further step, the models have been compared. Assumptions about the results, how they were obtained and whether they are robust, are given at the end of the report in chapter 5.

# Preprocessing

Die Sprache bietet für uns Menschen eine möglichkeit, komplexe Themen in einem anpassungsfähigem und nationalen, zum Teil gar universellem, Weg verständlich zu vermitteln. Diese Komplexität spiegelt sich in den schier endlosen Möglichkeiten des Sprachgebrauchs wider. Da Machine Learning Modelle als Input numerische Werte verlangen muss eine Transformation des Textes stattfinden. Meist werden die einzelen Wörter eines Satzes in eine Liste aufgeteilt und jeweils in ihre Grundform gebracht.Der Prozess des umwandelns in die Grundform wird Stemming genannt.Stemming (stem form reduction, normal form reduction) is the term used in information retrieval as well as in linguistic computer science to describe a procedure   
by which different morphological variants of a word are reduced to their common root, e.g. the declension of Wortes or words to Wort and conjugation of "gesehen" or "sah" to "seh".Modelle wie BERT oder T5 können auch unpräprozessierte Texte als Input verwenden [2]. Folgende Schritte wurden beim IMDB Movie Review-Dataset vorgenommen:

1. Entfernen der HTML Tags. Diese sind für das Verständnis des Inputs nicht relevant
2. Entfernen von Klammern um den Text
3. Entfernen von speziellen Charakteren. E. g. ?,!,/ etc.
4. Entfernen von Stoppwörtern. Stoppwörter sind Wörter die kaum Informationen beinhalten. Zum Beispiel Verbindungswörter. Diese werden entfernt.
5. Wörter in die Grundform bringen. Stemming.

Für Transformer Modelle wie BERT werden die Schritte 4. und 5. Nicht angewandt. Auch die Zielvariable *Sentiment* wurde angepasst. Ein Encoding der binären Zielvariable wurde vorgenommen. Ein positives Review wird nun mit dem Integer 1 annotiert, ein negatives Review mit dem Integer 0. Tabelle 2 zeigt ein Beispiel des vorbereiteten Textes:

Table 2, Example of a review after preprocessing

|  |  |  |
| --- | --- | --- |
| **Index** | **Review** | **Sentiment** |
| 15594 | [“saw”, “thi”, “movi”, “wa”,..] | 0 |

## Train-Test-Split

Zur reproduzierbarkeit wurde eine Funktion bereitgestellt die für alle Modelle denselben Split der Inputdaten retourniert. Die 50 Tausend Reviews werden zu 80% (40 Tausend Reviews) in ein Testset aufgeteilt. Das Trainingset besteht aus den restlichen 20% (10 Tausend Reviews).

# Modelling

## Baseline Model

Als grundlegenden Vergleich wurde ein Baseline Modell erstellt. Der Ansatz dieses Modelles ist es positive und negative Wörter in einem Review zu zählen und basierend auf diesem Count ein positives oder negatives Sentiment vorzuschlagen. Eine Liste mit positiven Wörtern sowie eine mit negativen Wörtern wurde importiert. Da hinter dieser Methode kein lernender Algorithmus steckt, wurde der Score nur auf dem Testdatensatz berechnet.

## Matrizenrepräsentation

### Bag of Words

Bag of Words, ein Modell welches oft zur Dokumentenklassifikation basierend auf der Häufigkeit von Begriffen verwendet wird, wandelt eine Kollektion von Texten/Dokumenten in eine Matrix mit der Tokenhäufigkeit als Elemente um. Wobei die Zeile das jeweilige Review darstellt und die Spalte ein Wort. Grammatik und Anordnung des Inputs wird in diesem Modell nicht mit einbezogen. Im Trainset wurden nach dem Pre-Processing insgesamt 6209089 unterschiedliche Wörter gezählt. Bag of Words wurde als Input einer Linearen Regression, SVM sowie Multinomialen Naive Bayes verwendet.

### TF-IDF

TF-IDF steht für Term Frequency – Inverse Document Frequency und wird für jedes Wort im Corpus berechnet. Dieser Wert gibt einen Einblick in die Wichtigkeit eines Begriffes. Die Term Frequency, also die relative Häufigkeit eines Begriffes in einem Dokument kombiniert mit der relativen, logarithmierten Häufigkeit über alle Dokumente (Formel 1) gibt Wörtern die häufig und in vielen Dokumenten vorkommen weniger Gewicht als Wörtern die nur in wenigen Dokumenten vorkommen.

Wobei TF-IDF eine Kombination der beiden Formeln darstellt:

Das Resultat von TF-IDF ist ebenfalls eine Matrix mit den Dimensionen .

## Klassische Modelle

Neben dem Baseline Modell wurden zu Beginn klassische Machine Learning Modelle verwendet. Darunter OLS[[2]](#footnote-2) Linear Regression mit l2 Regularisierung, Suport Vector Machines mit Hinge Loss und ein Multinomiales Naive Bayes Modell. Bei den drei Modellen wurde die Accuracy mit Bag of Words und TF-IDF als Input gemessen.

## Deepl Learning Models

Als eine erweiterung wurden zwei Deep Learning Modelle trainiert. Die verwendeten Daten wurden mit denselben Präprozessierungsschritten wie in Kapitel II beschrieben vorbereitet.

### Bidirectional LSTM

LSTM[[3]](#footnote-3) verwendet ein vortrainertes Word-Embedding. Aufgrund von Memory Constraints wurde für diesen Fall die fastText Mini-Version verwendet. GloVe (Global Vectors), ein weiteres Word-Embedding wurde als Vergleich ebenfalls importiert. Somit wurden zwei Modelle mit unterschiedlichen Embeddings trainiert. Das LSTM Modell wurde mit einer maximalen Länge von 128, einer Trainingbatchsize von 16, Validierungsbatchsize von 6 und mit 5 Epochen trainiert.

### BERT

Ein weiteres language Modell, BERT, wurde auf einem Corpus mit 800 Millionen Wörtern aus BooksCorpus und 2'500 Millionen Wörtern von Wikipedia trainiert. Die Input Daten wurden wie in Kapitel II vorbereitet, mit Ausnahme der Schritte 4 und 5. Eine maximale Länge von 512, eine Batchsize von 32 wurde bei 4 Epochen verwendet.

Für alle Modelle wurde die Accuracy auf Training- und Testset festgehalten. Die Resultate werden im folgenden Kapitel besprochen.

# Evaluation

Die Evaluation der Modelle wurde jeweils durch die Metrik Accuracy vorgenommen. Accuracy gibt die relative Häufigkeit einer korrekten Vorhersage an. Eine bekannte Metrik für eine binäre Klassifikation. Bei lernenden Algorithmen wie Bidirektionale LSTMs oder BERT ist neben dem finalen Score auch der Verlauf der Metrik über die Trainingsepochen interessant und wichtig um Overfitting zu erkennen.

|  |  |  |
| --- | --- | --- |
| **Modell** | **Accuracy** | |
| **Train** | **Test** |
| Baseline | - | 0.7131 |
| Linear Regression (TF-IDF) |  | 0.7512 |
| Linear Regression (BoW) |  | 0.75 |
| SVM (TF-IDF) |  | 0.5829 |
| SVM (BoW) |  | 0.5112 |
| Multinomial Naïve Bayes (TF-IDF) |  | 0.751 |
| Multinomial Naïve Bayes (BoW) |  | 0.7509 |
| LSTM (fastText) |  | 0.8543 |
| LSTM (GloVe) |  | 0.8746 |
| BERT | 0.87663 | 0.7186 |

##### References

[1] “Sentiment Analysis.” http://ai.stanford.edu/~amaas/data/sentiment/ (accessed Oct. 31, 2021).

[2] “nlp - Using trained BERT Model and Data Preprocessing,” *Stack Overflow*. https://stackoverflow.com/questions/63979544/using-trained-bert-model-and-data-preprocessing (accessed Nov. 03, 2021).

1. IMDB: Internet Movie Database [↑](#footnote-ref-1)
2. OLS; Ordinary Least Squares [↑](#footnote-ref-2)
3. LSTM: Long Short-Term Memory [↑](#footnote-ref-3)