Deep Learning for Natural Language Processing

Homework 1

Eirini Kolimatsi

Student ID: 7115112200015
Email: eirini[dot]kolimatsi[at]di[dot]uoa[dot]gr

MSc Computer Science
University of Athens
Academic Year: 2022-2023

Contents

1 Data Preparation

Libraries Firstly, all required libraries should be loaded. For the needs of this assignments modules from scikit-learn (sklearn), nltk, pandas, numpy, re, matplotlib, contractions, google.colab and wordcloud.

Data Loading In order to access the data provided for the assignment, the csv file was uploaded in Google drive. Then the notebook got "connected" to Google Drive, to allow reading the file from a path. The path that leads to the file is /content/drive/path/to/file, where path/to/file should be replaced with the actual location in Google Drive where the file was uploaded.

1.1 Data Cleaning

The provided data have 3 columns: rating, URL and review not needed. Reviews are expressed in a scale from 1 to 10. As a result, all negative reviews (with a score below or equal to 4) are marked with 0 and all positive reviews (with a score above or equal to 7) are marked with 1.
Regarding the URL, we can separate the movie identifier from the rest of the URL. This feature cannot be used per se in the training, but someone may decide to use it to optimize the way data are split in the train and validation sets.
Going through the reviews, it is clear that the data need to be cleaned. Firstly, reviews include HTML tags (eg `<br>`) which cannot help understand the sentiment of the review and need to be removed. Similarly for punctuation points. Also, the characters should all be in lower case to avoid the same word being interpreted as a different one given the case. Lastly, given that reviews are everyday simple text there are contractions present in the text. For example, isn't is used instead of is not. To "expand" these abbreviated words, a dedicated library is used, that is called contractions.

On top of these steps, there are many more linguistic cleaning steps that can be applied, of which three have been selected to be examined. Firstly, the removal of stopwords, stemming and lemmatization.

What is suggested is to explore which of the aforementioned techniques of text cleaning better serves the given dataset and the purposes of model training to choose which ones to apply. Nevertheless, some of the cleaning steps ought to be done in all cases, eg HTML tags removal and lower case adaptation. To decide which methods to use, a quick model training will be performed to find out if it helps out the model or not.

1.2 Data Exploration

Before training the model, it is useful to explore further the dataset. We observe that the number of positive and negative reviews is very similar, so the dataset is balanced. Also, there are no missing values and as a result we should not remove them or fix them in another way.
To gain a deeper understanding of the dataset, wordcloud plots are generated for both positive and negative reviews. The positive wordcloud include words like great, love etc.

The negative wordcloud contains words like bad and worst.

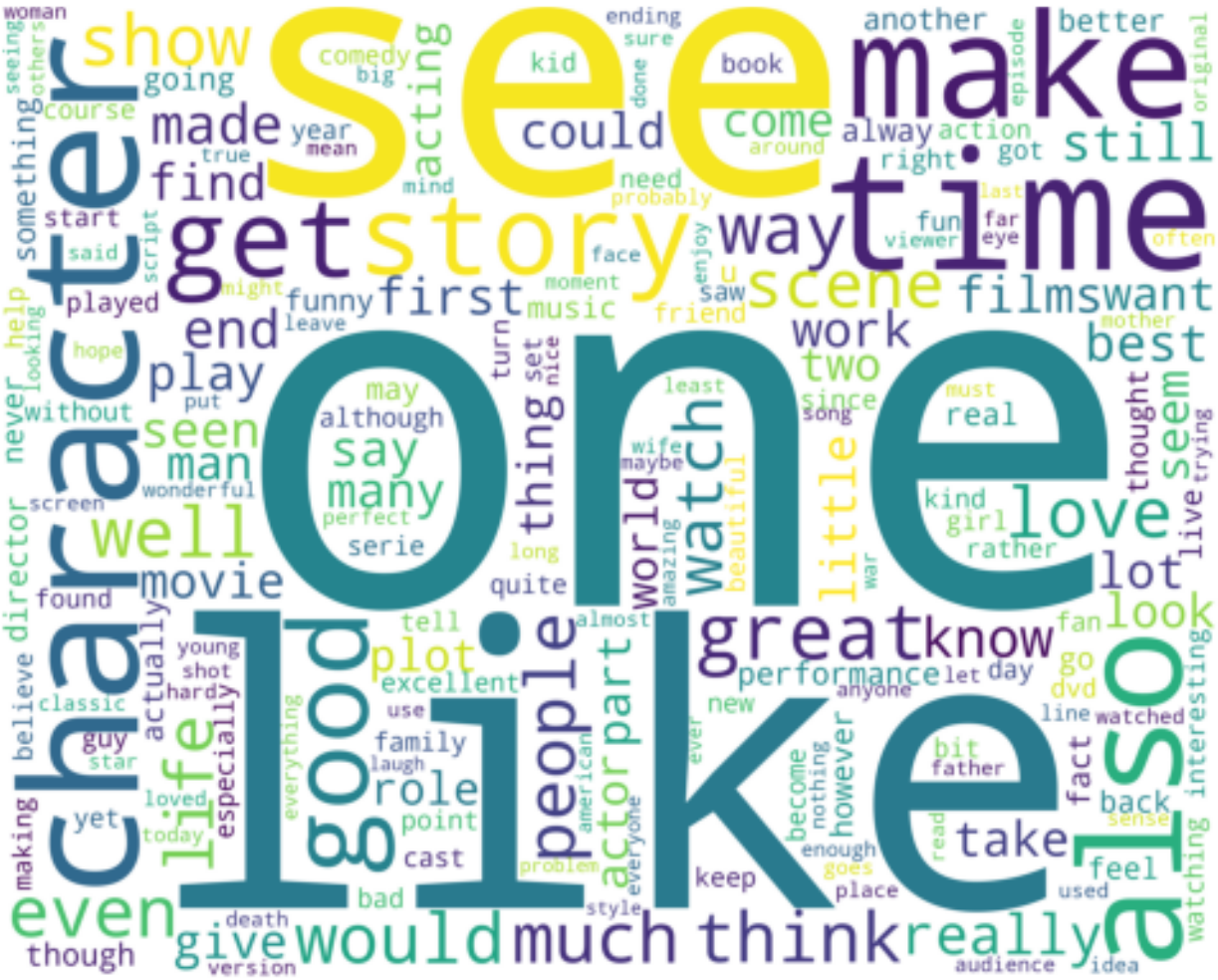


Figure 1: Positive Wordcloud

We might notice that the most frequent words from both wordclouds are similar, which leads to the assumption that words that are often mentioned might not be the most suitable to extract sentiment.

2 Training

For the training, the approach used is to try to build the features used, as well as the vectorisation methods from a simple selection to more complicated combinations and along the way evaluate the performance. At the end, the most suitable model is selected to be fine tuned and examined more for its performance.

Regarding the train test split, given that the dataset is balanced a 80-20 split is chosen.

The first attempt for model training was done only doing the necessary data cleaning, as described in the Data Cleaning section and with the help of a CountVectorizer to interpret the reviews. Two different values where given for C, 0.01 and 1. The model seems to perform at least fine given the preliminary results.

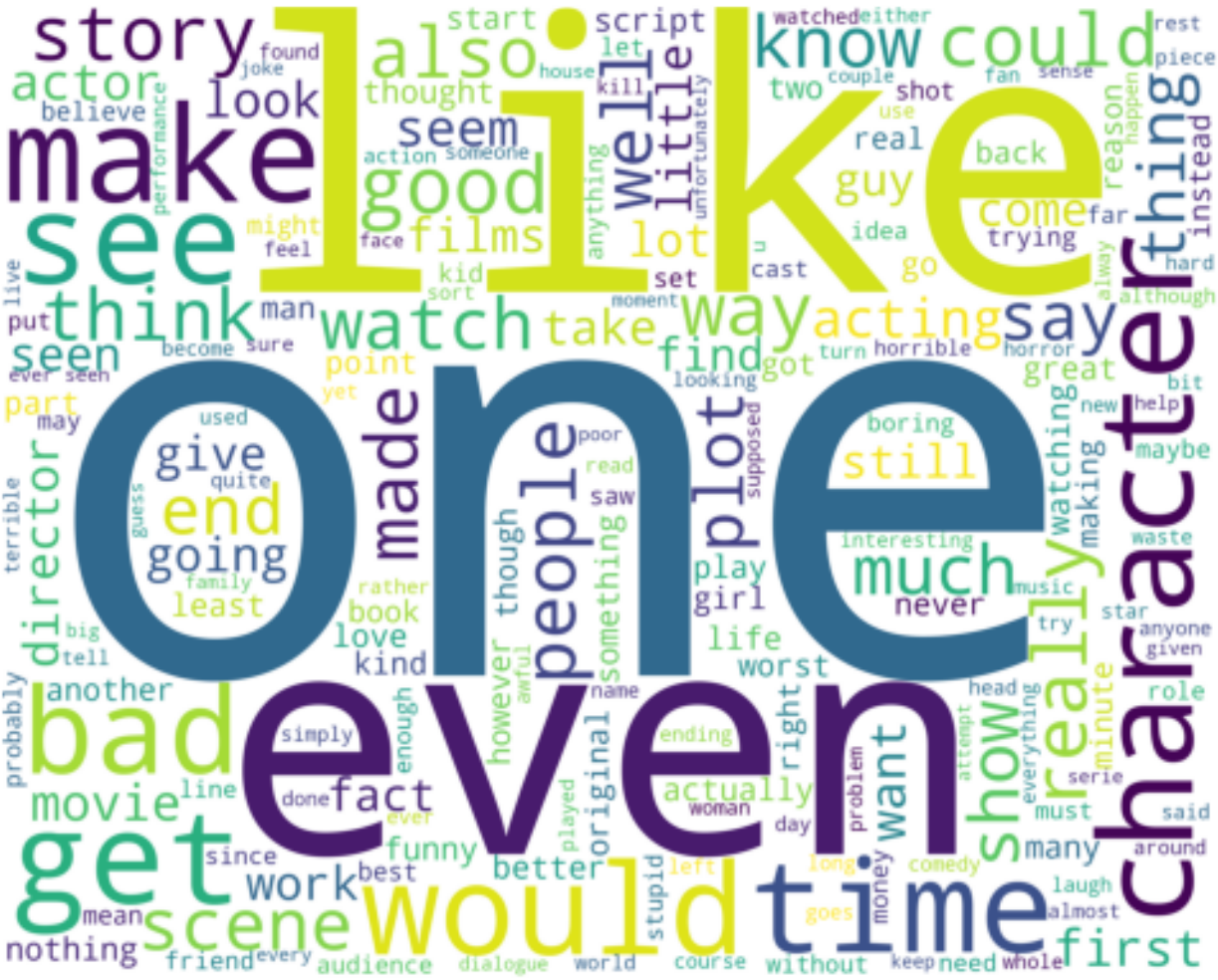


Figure 2: Negative Wordcloud

	Train Dataset	Validation Dataset
Precision	89.05%	87.45%
Recall	90.72%	89.19%
F1 Score	89.88%	88.31%

Then, for the second attempt instead of the CountVectorizer, TfidfVectorizer is used with different parameters (stopwords removal, min_df and ngram_range). This vectorizer performs worse than the CountVectorizer in the first attempt:

	Train Dataset	Validation Dataset
Precision	81.43%	80.45%
Recall	88.55%	88.72%
F1 Score	84.84%	84.39%

For the third attempt, stopwords removals, contractions removal and lemmatization was applied, as well as both CountrVectorizer and TfidfVectorizer. The CountVectorizer performs better:

	Train Dataset	Validation Dataset
Precision	88.56%	86.88%
Recall	90.50%	88.82%
F1 Score	89.52%	87.84%

For the fourth attempt, stopwords removals, contractions removal and stemming was applied, as well as both CountrVectorizer and TfidfVectorizer. The CountVectorizer performs better:

	Train Dataset	Validation Dataset
Precision	88.36%	86.94%
Recall	90.51%	89.86%
F1 Score	89.42%	88.38%

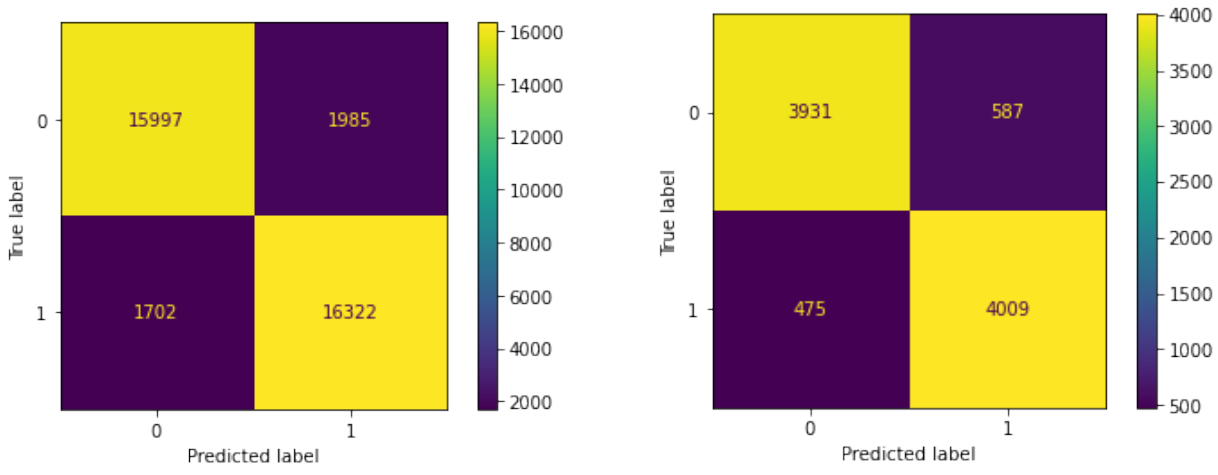
Finally, it seems like the 1st and the 4th model perform very close and they're the best approaches out of the 4. However, the 1st training attempt will be selected because it performs marginally better across mode metrics than the 4th.

3 Results

After selecting the model, it is tuned to select the best C and then it is cross validated (10-fold). Finally, the results are:

	Train Dataset	Validation Dataset
Precision	89.01%	87.53%
Recall	90.61%	89.15%
F1 Score	89.81%	88.33%

The confusion matrix has been generated for both training and validation results.



(a) Training Set

(b) Validation Set

Figure 3: Confusion Matrix

To run the produced model with test data, at the end of the notebook the path can be specified, and the following cells should be executed.

4 References

- Scikit-learn documentation
- Stackoverflow