Homonyms Problem in the text

By/ Ahmed Essam

Homonyms are words that share the same spelling or form (characters) but possess distinct meanings. For instance, the term "bank" can assume two disparate contexts, denoting both a financial institution and the edge of a river.

These homonyms hold significant relevance in sentiment analysis, given their capacity to alter a sentence's meaning or emotional tone entirely. Consider the following examples that highlight this challenge:

Sentence	Label
I hate the selfishness in you	NEGATIVE
I hate anyone who hurts you	POSITIVE

In the first sentence, the word "hate" renders the sentiment as NEGATIVE. Conversely, the same word, "hate" appears in the second sentence, shaping the sentiment of the sentence as POSITIVE. This poses a considerable challenge to models relying on fixed word embeddings. Therefore, employing contextualized embeddings leveraging attention mechanisms from transformers becomes crucial to grasping the comprehensive context within a sentence.

- Tools Used:

The project is implemented using the following Python packages:

Tool	Description
pandas	Data manipulation and analysis library
matplotlib	Data visualization library
seaborn	Statistical data visualization library
sklearn	Machine learning library for tasks like classification, regression, and clustering
WordCloud	Library for generating word clouds
CountVectorizer	Text preprocessing tool for converting text into numerical features
TensorFlow.Keras	High-level API for building and training deep learning models
Transformers	Library for working with transformer-based models like BERT
Datasets	Library for loading and processing datasets, including Hugging Face datasets

- Dataset:

The Stanford Sentiment Treebank (SST) is a corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. The corpus is based on the dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser and includes a total of 215,154 unique phrases from those parse trees, each annotated by 3 human judges.

Binary classification experiments on full sentences (negative or somewhat negative vs somewhat positive or positive with neutral sentences discarded) refer to the dataset as SST-2 or SST binary.

The dataset contains 3 features [idx, sentence, and label] and it comes in 3 different splits [train, validation, and test]. Here is the number of samples per split:

Split	Number of Samples
train	67349
validation	872
test	1821

- Methodology:

This phase encompasses dataset preparation preceding the modeling phase. It involves transforming the sentence column into integers utilizing the Tokenizer class from the keras library. Subsequently, the integer sequences are padded to conform to the maximum sequence length within the dataset.

- EDA:

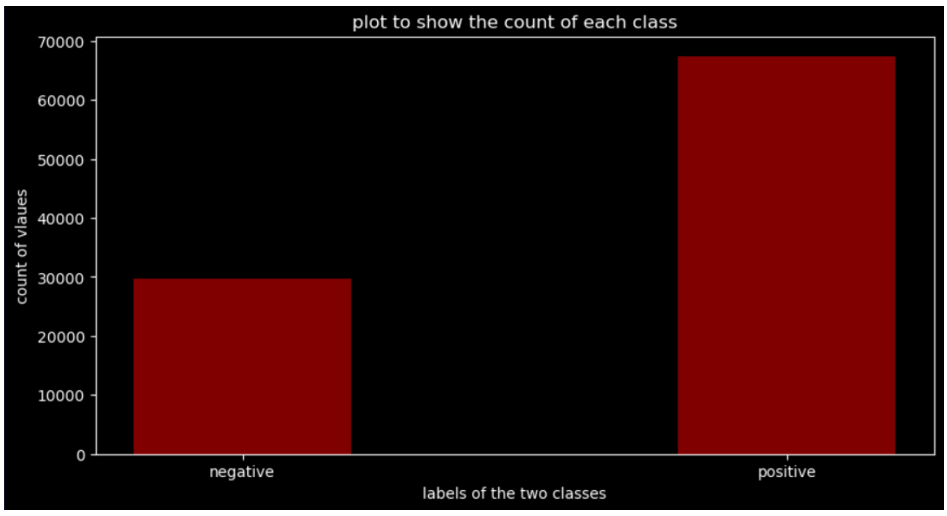


Figure 1, Class Distribution to check if the data is imbalanced

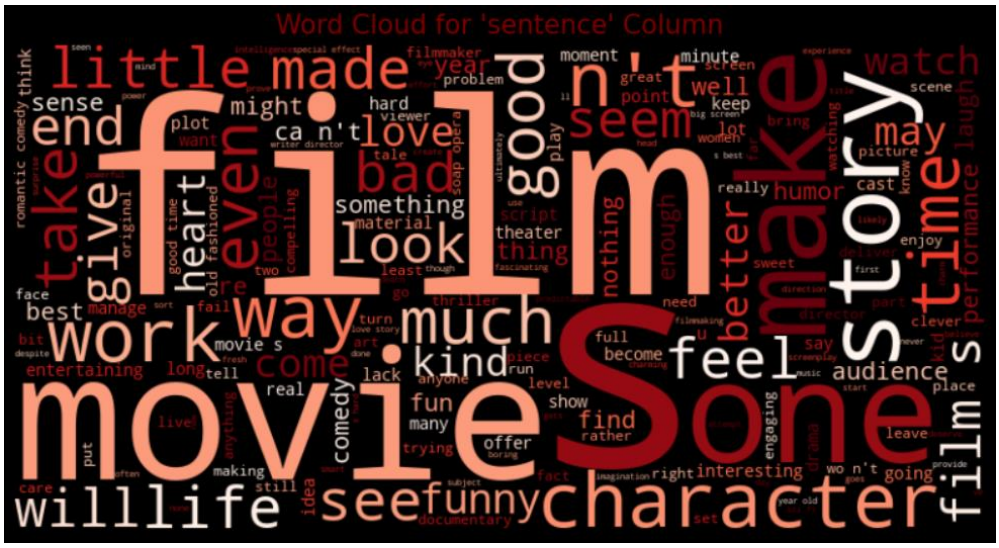


Figure 2, Most frequent words using wordcloud

- Model Selection:

The two models selected are LSTM-based and DistilBERT-based. The rationale behind choosing these models lies in the necessity to observe distinctions between fixed-embedding models, exemplified by LSTM, and contextualized embedding models, exemplified by DistilBERT. Additionally, two iterations of the LSTM-based model were trained: one before addressing the dataset imbalance and another after resolving it.

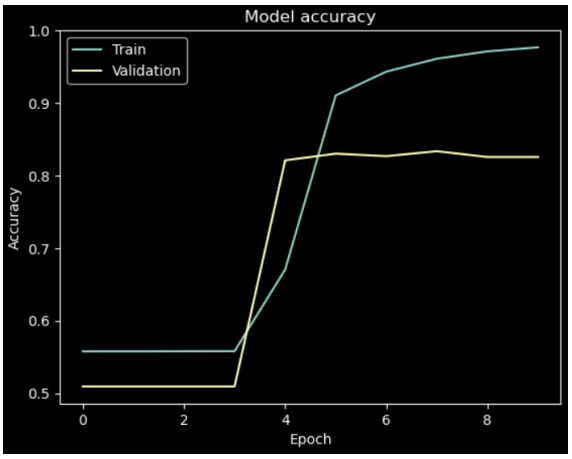
- Results:

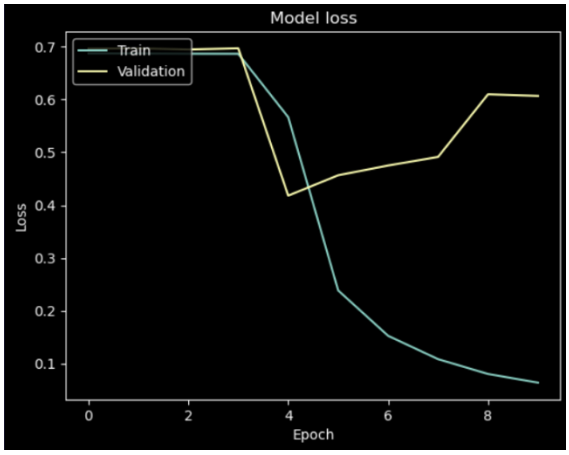
Before training, I conducted a straightforward grid search to determine the optimal hyperparameters. This included parameters such as the embedding dimension for the Keras Embedding layer and the number of hidden units in the LSTM cell. The detailed results, ordered by the highest validation accuracy.

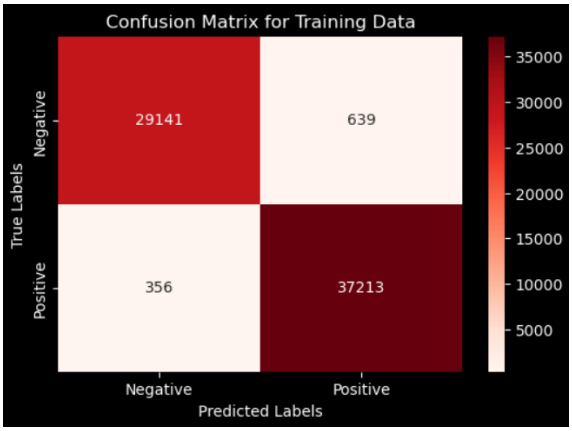
The model with hyperparameters used in it:

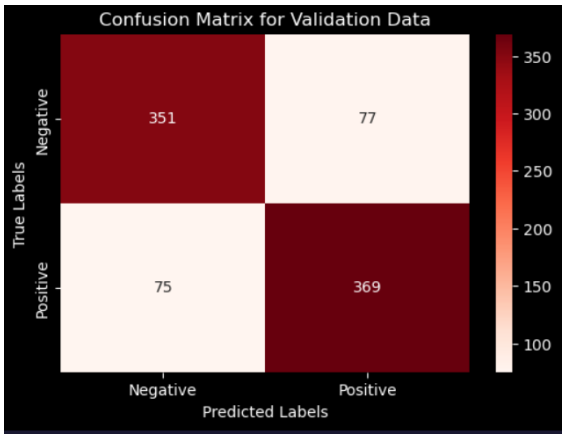
Parameter	Value
Model Type	LSTM
Embedding Dimension	128
Units	32
Dropout Rate	0.3

The results of this model was**:**









Training Classification Report:

	precision	recall	f1-score	support
0	0.99	0.98	0.98	29780
1	0.98	0.99	0.99	37569
accuracy			0.99	67349
macro avg	0.99	0.98	0.99	67349
weighted avg	0.99	0.99	0.99	67349

Validation Classification Report:

	precision	recall	f1-score	support
0	0.82	0.82	0.82	428
1	0.83	0.83	0.83	444
accuracy			0.83	872
macro avg	0.83	0.83	0.83	872
weighted avg	0.83	0.83	0.83	872

Conclusion:

I focused on addressing the complexity of words with multiple meanings and how they can impact natural language understanding. This project aimed to disambiguate homonyms within various contexts using contextual embeddings, such as BERT or Word2Vec, and rule-based or supervised learning techniques to understand the appropriate meanings of words in different sentences.

- Testing the model:

Text	Predicted Score	Predicted Label	TRUE Label
The bass was amazing to listen to at the concert	0.999711	POSITIVE	POSITIVE
The farmer went to produce a new variety of apple	0.471692	POSITIVE	POSITIVE
The tear in her dress was unfortunate	0.013828	NEGATIVE	NEGATIVE
He wound up the clock and prepared for the day	0.459765	POSITIVE	POSITIVE
I'll park by the river to read a good book	0.868612	POSITIVE	POSITIVE
The artist decided to rock the stage tonight	0.001069	NEGATIVE	POSITIVE
She shed a tear when she saw the results	0.36632	POSITIVE	NEGATIVE
The bow was tied neatly around the gift	0.997682	POSITIVE	POSITIVE
He was asked to lead the meeting	0.008404	NEGATIVE	POSITIVE
I can't bear the thought of losing	0.995474	POSITIVE	NEGATIVE
The duck refused to budge from the pond	0.247102	POSITIVE	POSITIVE
She refused to desert her friend in need	0.982428	POSITIVE	POSITIVE
I didn't mind the cold wind, but the wind in t...	0.019448	NEGATIVE	NEGATIVE
He got the permit to work in the construction ...	0.779522	POSITIVE	POSITIVE
I had to duck when the ball came flying at me	0.001984	NEGATIVE	NEGATIVE

The Biggest challenge was achieving accurate disambiguation of homonyms across diverse contexts. Homonyms can take on vastly different meanings depending on their surrounding words and overall sentence structure, making it difficult to create a model that generalizes well. Selecting and fine-tuning the right contextual embeddings and models to capture these subtle distinctions, especially with limited labeled data for less common meanings, required extensive experimentation and careful model selection. Balancing these aspects to ensure consistent, accurate interpretation across contexts was both the most complex and rewarding part of the project.

I have learned valuable insights into the intricacies of natural language processing, particularly in handling words with multiple meanings. I learned how context is crucial in interpreting language accurately, especially for ambiguous words, and how various NLP models, such as contextual embeddings like BERT, are designed to capture these nuances. Additionally, I developed a deeper understanding of data preprocessing and annotation for disambiguation tasks and learned how to approach and structure projects where nuanced, context-sensitive accuracy is critical. This project strengthened my ability to refine model performance through targeted tuning and careful data handling.