UNVEILING SENTIMENT: A DEEP DIVE INTO ADVANCED SENTIMENT ANALYSIS MODELS ON SENTIMENT LABELLED SENTENCES

Abstract:

This article delves into a comprehensive exploration of sentiment analysis, employing advanced machine learning techniques on the Sentiment Labelled Sentences dataset. The code implementation encompasses a transducer method and the creation of a sentiment analysis class, utilizing pyformlang, scikit-learn and Keras.

The study proceeds with a detailed code demonstration for data collection, preprocessing, and supervised learning using the NLTK library. A DummyClassifier serves as a baseline model, paving the way for the creation and evaluation of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models. Hyperparameter tuning and metric assessments, including accuracy, precision, recall, and F1-score, reveal the superior performance of advanced deep learning architectures over the baseline.

A meticulous analysis of results showcases the strengths and weaknesses of each model, providing valuable insights into the nuances of sentiment capture.

Authors:

- Juan Esteban Duque Taborda
- Juan Camilo Bernal Marín
- Cristian Felipe Perafán

Inroduction

Sentiment analysis, also known as opinion mining, is a field of natural language processing (NLP) that aims to determine the sentiment expressed in a piece of text. It has significant applications in understanding customer opinions, social media analysis, and feedback analysis. In this report, we will explore the process of sentiment analysis using machine learning models, starting with a baseline using a DummyClassifier and progressing to more complex models like vanilla Recurrent Neural Networks (RNN) and Long Short-Term Memory networks (LSTM).

Objetives

1. Acquire and preprocess the Sentiment Labelled Sentences dataset, employing NLTK for tokenization, lowercase conversion, and stopwords removal to create a structured dataset conducive to supervised learning.
2. Establish a baseline for sentiment analysis using a DummyClassifier, training it on the preprocessed data, and evaluating its performance in terms of accuracy, precision, recall, and F1-score.
3. Develop and implement advanced sentiment analysis models, including vanilla Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) networks. Train both models on the preprocessed data, evaluate their performance metrics, and identify optimal hyperparameters through techniques like GridSearchCV.
4. Conduct a comparative analysis of the DummyClassifier, vanilla RNN, and LSTM models, unraveling their respective strengths and weaknesses in capturing sentiment. Offer insights into the potential advantages of advanced models over the baseline approach.

Methodology

The methodology to be developed in this paper starts with the acquisition and preprocessing of the Sentiment Labelled Sentences dataset using NLTK. In addition to a previous data analysis to know how many positive (1) and negative (0) data has the dataset, also the dominant words in each of the files, both from Amazon, IMBd and Yelp. This ensures a solid basis for further analysis. A DummyClassifier establishes a baseline, training on the preprocessed data to provide a benchmark for comparison. Advanced sentiment analysis models, namely the vanilla RNN and LSTM networks, are developed and trained on the preprocessed data set. Hyperparameter tuning employs GridSearchCV to improve the predictive capability of the models. A comparative analysis evaluates the performance of the DummyClassifier, RNN and LSTM models, providing insights into their respective strengths and weaknesses in sentiment capture. Visualization techniques, including word clouds, enrich the interpretation of the results.

Results

Accuracy = 47.27%

	Presicion	Recall	F1 - score
Class 0	0.0	0.0	0.0
Class 1	0.47	1.0	0.64

DummyClasifier

The DummyClassifier essentially predicts the majority class (Class 1) every time. As a result, it achieves high recall for Class 1 but performs poorly in terms of precision and accuracy. The warning about ill-defined precision indicates that precision for Class 0 is 0.0 because there are no predicted samples for that class. The DummyClassifier method is only used to have a baseline for how difficult the problem will be to deal with.

RNN

Accuracy = 72.47%

	Presicion	Recall	F1 - score
Class 0	71.10%	79.77%	75.35%
Class 1	74.04%	64.36%	68.86%

Accuracy = 70.42%

	Presicion	Recall	F1 - score
Class 0	71.75%	72.41%	72.08%
Class 1	68.91%	68.21%	68.56%

Accuracy: correctly classifies sentiments
Precision: class 0 and class 1 is balanced.
Recall: good capture true negatives, but difficult with all positive.
F1-score: reflecting a trade-off between precision and recall

The best model: hyperparameter tuning did not lead to a significant improvement

LSTM

Accuracy: higher accuracy on the test set.
Precision: class 0 and class 1 is balanced.
Recall: has good recall for both positive (Class 1) and negative (Class 0) sentiments
F1-score: a good trade-off between precision and recall

The best model: hyperparameter tuning led to a significant improvement. This underscores the effectiveness of LSTM in capturing sequential patterns and dependencies for sentiment analysis.

Accuracy = 73.45%

	Presicion	Recall	F1 - score
Class 0	76.47%	71.72%	74.02%
Class 1	70.50%	75.38%	72.86%

Accuracy = 75.76%

	Presicion	Recall	F1 - score
Class 0	74.95%	81.15%	77.92%
Class 1	76.84%	69.74%	73.12%

As seen in the results of each method in the sentiment analysis, let's see why LSTM is better than the others.

In this comparative analysis, the use of two libraries, sklearn and Keras, is pivotal in understanding the performance disparities among the implemented methods. Sklearn, a versatile machine learning library, offers simplicity and efficiency. On the other hand, Keras, a high-level deep learning library, excels in building and training neural networks.

In the context of sentiment analysis, the limitations of a simple RNN become apparent, especially in capturing long-term dependencies and contextual nuances within textual data. The superior performance of Keras with LSTM models stems from the latter's intrinsic design to overcome issues like the vanishing gradient problem. LSTMs, equipped with memory cells and specialized activation mechanisms, excel in retaining and utilizing information over extended sequences, a critical aspect in sentiment analysis where understanding context and long-range dependencies is paramount.

Conclusion

DummyClassifier served as a benchmark and the results showed the superior performance of the RNN and LSTM models, emphasizing the importance of deep learning architectures for capturing nuances of sentiment within textual data. Hyperparameter tuning, facilitated by GridSearchCV, allowed the identification of optimal model configurations.

The study achieved its goal of evaluating and comparing sentiment analysis models, it also provided valuable insights into the effectiveness of advanced models for sentiment analysis tasks. The findings highlight the advantages of employing deep learning techniques, particularly RNNs and LSTMs, to capture nuances of sentiment compared to simpler baseline approaches.