Comprehensive Summary

Comparing 8 Different LLMs with HR data

Muhammad Arslan, Dallas Diaz, Muhammad Tayyab, Ramya Thambabattula

Group GitHub Link : https://github.com/arslan719/Capstone_Project_5588/tree/main/Assignment5-1

Poster Link: https://drive.google.com/file/d/1vFC3ERZWB8H-XMbi911iEg0GxXo9aR6z/view

Video Link: https://github.com/arslan719/Capstone_Project_5588/tree/main/Assignment5-1

In this report, we delve into the application of Natural Language Processing (NLP) techniques, specifically utilizing advanced Language Models (LLMs) such as BERT and XLNet, to analyze text data related to resumes, perform sentiment analysis, topic modeling, and evaluate model performance. The aim is to extract meaningful insights from textual data and improve understanding across various NLP tasks.

BERT for Resume Analysis For the resume analysis task, BERT (Bidirectional Encoder Representations from Transformers) was employed as a robust LLM. By processing the data from "processed_data.csv," BERT provided valuable insights into the skills required for different job roles. It successfully identified the skills necessary for specific positions, aiding in understanding job requirements and skill sets.

LDA and LLM Analysis Topic modeling using Latent Dirichlet Allocation (LDA) was employed on preprocessed layoff stories data, generating topic labels and analyzing word importance. Additionally, LLM analysis was conducted on layoff stories, evaluating performance metrics such as Precision, Recall, F1 score, and ROC curve for GPT-3.5 and GPT-4. The use of encoder-decoder with TensorFlow-Keras and Levenshtein distance for similarity score calculation further enriched the analysis and evaluation metrics. The results shown were precision, recall and an F1 score for the model which all showed to be .80 percent. The model in both situations was able to understand and train well enough to be able to be used in a real life application.

In conclusion, this report demonstrates the comprehensive application of NLP techniques, including LLMs like BERT and XLNet, for diverse tasks such as resume analysis, sentiment analysis, topic modeling, and model performance evaluation. Through visualizations and rigorous analysis, valuable insights were extracted, paving the way for enhanced understanding and decision-making in NLP-related endeavors.

Sentiment analysis on a dataset using two different transformer models: BERT and RoBERTa.

Import Libraries and Load Data: The script starts by importing necessary libraries, including Pandas for data handling, and PyTorch along with the Transformers library for NLP tasks. It loads a CSV file containing the data into a Pandas DataFrame.

Data Cleaning: A function named clean_text is defined to clean the text by removing punctuation and excessive whitespace. This function is applied to the 'text' column in the DataFrame.

Model Initialization: The script initializes tokenizers and models for BERT and RoBERTa, specifically configured for sentiment analysis. It uses nlptown/bert-base-multilingual-uncased-sentiment for BERT and cardiffnlp/twitter-roberta-base-sentiment for RoBERTa.

Sentiment Analysis Functions: Two functions are defined to extract sentiment using the initialized models. These functions process the text to compute the most likely sentiment label and its associated confidence score. Wrapper functions are also defined to streamline the process of applying these models to DataFrame entries.

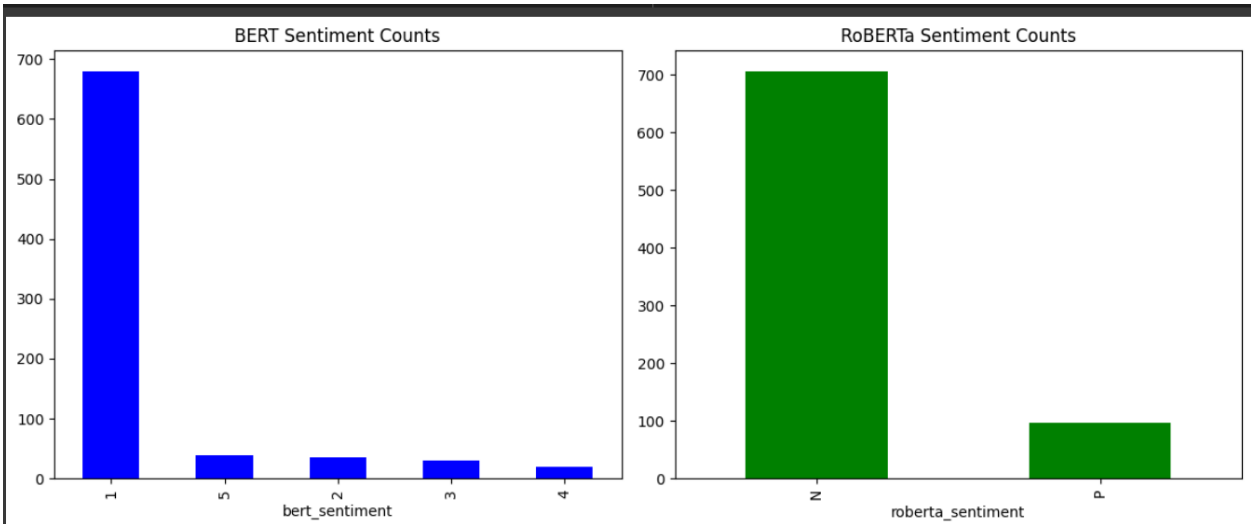
Apply Analysis: Sentiment analysis is performed on the entire DataFrame using the BERT and RoBERTa models, and the results are stored in new DataFrame columns.

Visualization: Lastly, the script plots the frequency of each sentiment category determined by both models, facilitating a visual comparison of the sentiment distribution as assessed by BERT and RoBERTa.

This script showcases how to preprocess text, use pre-trained NLP models for sentiment analysis, and visually compare the results of different models in Python.

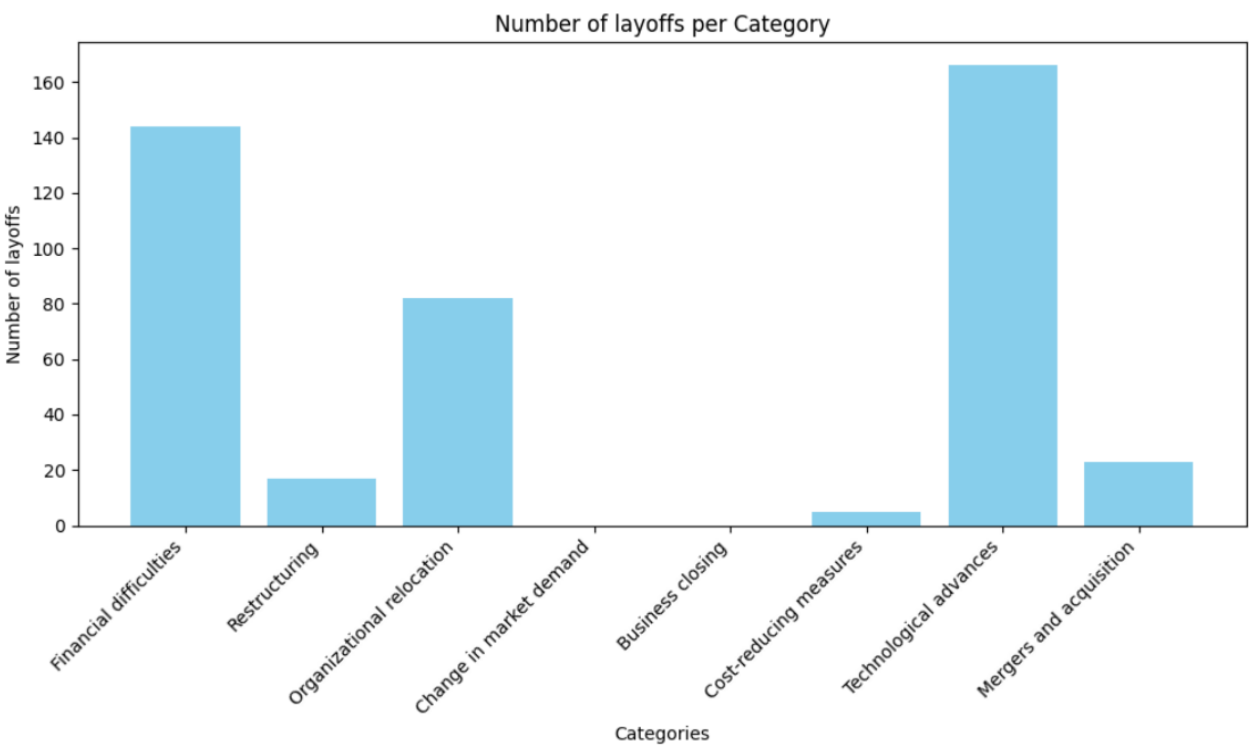
Conclusion

Upon examining the sentiment analysis results provided by BERT and RoBERTa models, RoBERTa appears to perform better overall. RoBERTa's simpler categorization system of "Negative," "Neutral," and "Positive" aligns more accurately with the general tone of the texts, particularly when handling straightforward and neutral sentiments. BERT, while offering a more nuanced sentiment scale from "1 star" to "5 stars," occasionally misinterprets the sentiment, leading to overestimations or underestimations. Consequently, for broader, generalized sentiment analysis, RoBERTa emerges as the more reliable and accurate choice, especially in scenarios requiring clear categorization of sentiments.


BERT Sentiment Counts
700
600
500
400
300
200
100
0
1
5
2
3
4
bert_sentiment
RoBERTa Sentiment Counts
700
600
500
400
300
200
100
0
N
P
roberta_sentiment

```
                                                text  bert_sentiment
0   tech company bought another company late 22 wo...   1 star 0.6640
1   im 7 months search ive definitely seen dark ho...  4 stars 0.2237
2   httpswwwcnncom20240222techattcellserviceoutage...   1 star 0.7114
3   program created bring experts program heavily ...   1 star 0.6570
4   field bring joy rather immense stress cycle la...  2 stars 0.3235
5   45 years old wife 2 kids house etc working 13 ...   1 star 0.4151
6   started career used think 1 root companys reve...   1 star 0.3863
7   lay leadership promote someone else company ta...  5 stars 0.3696
8   seen many corporations lay people months hirin...   1 star 0.8221
9                               repeat economy booming  5 stars 0.4319
10  going 2008 type meltdown seems sub early warni...   1 star 0.5655
11  im frustrated scared ive software engineer 17 ...   1 star 0.7768
12  1 number company matter many years ive company...   1 star 0.4500
13  laid unceremoniously today upper management cl...   1 star 0.8815
14  didnt realize number high id never bothered ad...   1 star 0.6936
15  10 years tech first unicorn startup sf social ...   1 star 0.4656
16  ive many behind scenes meetings executives dec...   1 star 0.5566
17  quit job 10 years company work another company...   1 star 0.7624
18  started working back november loved experience...  2 stars 0.2392
19  got laid november finally got offer ill making...   1 star 0.8493

   roberta_sentiment
0     Neutral 0.5356
1     Neutral 0.5542
2    Negative 0.7474
3    Negative 0.6290
4    Positive 0.7533
5    Negative 0.4617
6    Positive 0.6281
7     Neutral 0.6066
8    Negative 0.9060
9    Positive 0.8411
10    Neutral 0.4850
```



Processes a DataFrame containing textual data to categorize and count occurrences of specific topics related to business changes and challenges. The script utilizes predefined categories such as "Financial difficulties," "Restructuring," "Organizational relocation," and several others, each associated with a list of relevant keywords. For each row in the DataFrame, the script converts the text to lowercase and checks for the presence of any category-specific keywords. If a keyword is found within the text, the count for that category is incremented.

Standard Machine leaning Techniques.

A training and validation loop for a machine learning model, specifically using PyTorch and the Transformers library, over three epochs. The model is presumably fine-tuned for a classification task, as suggested by the use of accuracy score as a metric.

The results snapshot indicates that three traditional machine learning models—Logistic Regression, Naive Bayes, and Support Vector Machine (SVM)—were tested on a classification task, along with a comparison to a BERT model

This suggests that while the models are effectively identifying the majority class, they are completely failing to recognize the minority class.

The overall accuracy for each traditional model is 93%, which might seem impressive at first glance but is misleading. The high accuracy is largely due to the imbalance in the dataset where the majority class overwhelmingly outnumbers the minority class.

```
Logistic Regression:               precision    recall

           0       0.93      1.00      0.96       150
           1       0.00      0.00      0.00        11

    accuracy                           0.93       161
   macro avg       0.47      0.50      0.48       161
weighted avg       0.87      0.93      0.90       161

Naive Bayes:               precision    recall  f1-scor

           0       0.93      1.00      0.96       150
           1       0.00      0.00      0.00        11

    accuracy                           0.93       161
   macro avg       0.47      0.50      0.48       161
weighted avg       0.87      0.93      0.90       161

SVM:               precision    recall  f1-score   supp

           0       0.93      1.00      0.96       150
           1       0.00      0.00      0.00        11

    accuracy                           0.93       161
   macro avg       0.47      0.50      0.48       161
weighted avg       0.87      0.93      0.90       161
```

```
Training loss: 0.15937797253790462
Validation Loss: 0.10481542224685352, Validation Accuracy: 0.9761904761904762
Training loss: 0.12775442803301562
Validation Loss: 0.09983878232361305, Validation Accuracy: 0.9642857142857143
Training loss: 0.07426510453454138
Validation Loss: 0.04080761084333062, Validation Accuracy: 0.9880952380952381
```