SCHOOL OF SOCIAL SCIENCES

**EXTENDED RESEARCH PROJECT:**

**ADDITIONAL MATERIAL**

**Large Language Models (LLMs) in Financial Application**

Performance of business-centric LLMs and distilled LLMs, within the context of finance, particularly focusing on financial forecasting.

**Student ID: 11358614**

**Supervisor: Eghbal Rahimikia**

2024

An extended research project report submitted to the University of Manchester for the degree of MSc. Data Science with Business and Management in the Faculty of Humanities.

**Overview**

The repository contains the code and resources necessary to reproduce the research conducted on the application of Large Language Models (LLMs) for financial forecasting using textual data. The project investigates the efficacy of various LLMs, both pre-trained and fine-tuned, in predicting market trends and improving portfolio management strategies.

**Repository Structure**

* **BM16\_LLM\_Finance\_WRDS\_Tuned.ipynb**: Notebook for fine-tuning LLMs on financial news data and assessing their predictive capabilities.
* **BM16\_LLM\_WRDS\_Finance\_RollingWindow.ipynb**: Notebook for processing financial news data, extracting features using untuned LLMs, and evaluating their performance.

**Setup Instructions (Running the Notebooks)**

1. **Setup**

* Clone the Repository and Navigate to Project Directory

***Command: git clone <repository-url>***

* Create a Python Virtual Environment and Install All Required Packages

**Set Up a Virtual Environment**

Navigate to the project directory

***Command: cd path\_to\_project\_directory***

*Create a virtual environment*

***Command: python -m venv venv***

*Activate the virtual environment*

***a. On Windows:\venv\Scripts\activate***

***b. On macOS/Linux: source venv/bin/activate***

1. **Execution**

* Laun Jupyter Notebook or Jupyter Lab.
* Open **BM16\_LLM\_Finance\_WRDS\_Tuned.ipynb** to execute fine-tuning of models**.**
* Open **BM16\_LLM\_WRDS\_Finance\_RollingWindow.ipynb** to run analysis with untuned models.
* Follow instructions in each notebook to reproduce the results. The Code is self-explanatory.

**Technical Appendix**

This section provides a comprehensive guide on the steps, methods, and tools used to carry out the analysis in this project.

**1. Data Collection and Preprocessing**

**Data Source**

* **Source**: Financial news articles/headlines were retrieved from the Wharton Research Data Services (WRDS).
* **Access**: Ensure you have the necessary credentials to access WRDS. For information on access, visit the WRDS website.

**Data Capture**

* **Data Extraction**: Data is extracted using SQL queries using the WRDS API. The SQL queries are tailored to filter and retrieve relevant financial news articles and stock data for analysis.
* **Datasets extracted from WRDS API**:
  + 1. CRSP
    2. Capital IQ
    3. GVKey
* **File Format**: Post running the queries, data is exported to CSV format and stored in the **‘{user\_specified\_path}/LLMS/Finance/data’** directory.
* **Required Files**: The code expects the dataset files to be named appropriately and placed in the specified directory (**‘{user\_specified\_path}/BM16\_LLMs/Finance/data’).** Update the paths in the notebooks.

**Preprocessing Steps**

1. **Text Cleaning**

Remove non-alphanumeric characters, stopwords, and punctuation using Python’s **‘nltk’** and **‘re’** libraries.

1. **Tokenization and Embeddings**

Tokenization is performed using the ‘transformers’ library, specifically the tokenizers for each LLM model.

* + - 1. BertTokenizer
      2. RobertaTokenizer
      3. DistilBertTokenizer
      4. AutoTokenizer (for DistilRoBERTa and FinBERT)

For embeddings, pre-trained models from Hugging Face (transformers library) are used, including

1. BERT
2. RoBERTa
3. DistilBERT
4. DistilRoBERTa
5. FinBERT
6. **Feature Engineering**

* Extract embeddings from textual data to be used as features for machine learning models.
* Sentiment scores are calculated using the pre-trained LLMs and appended to the feature set for portfolio construction.

1. **Exploratory Data Analysis**

* Descriptive statistics and visualizations (e.g. Bar plots, Distribution plots, words clouds) to understand the data characteristics.
* Time Series analysis to observe trends and patterns in financial news over time.

**2. MODEL TRAINING and EVALUATION**

**Untuned Models (‘LLM\_Finance\_WRDS\_Untuned.ipynb’)**

**Model Setup:**

* Load pre-trained LLMs using the ‘transformers’ library from Huggin Face.

Models used

1. BERT: *bert-base-uncased*
2. RoBERTa: *roberta-base*
3. DistilBERT: *distilbert-base-uncased*
4. DistilRoBERTa: *distilroberta-base*
5. FinBERT: *yiyanghkust/finbert-tone)*

**Training Process:**

* Use extracted embeddings as features for Logistic Regression to predict market trends.

**Evaluation:**

* Evaluate model performance using metrics like accuracy, precision, recall, F1-score.

**Fine-Tuned Models (‘BM16\_LLM\_Finance\_WRDS\_Tuned.ipynb)**

**Fine-Tuning Setup:**

* Fine-tune LLMs using the ‘transformers’ library built-in functionality for sequence classification tasks.
* Hyperparameters such as learning rate, batch size, and epochs are optimized based on prior experiments.

**Fine-Tuning Process:**

* Models are fine-tuned on financial news headlines with labelled price directions (up, down)

**Post-Training Analysis**

* Similar evaluation metrics as untuned models (accuracy, F1-score etc.) are used for comparison.

**3. PORTFOLIO CONSTRUCTION and ANALYSIS**

Both pre-trained and fine-tuned models are used to generate predictions that inform portfolio decisions.

**Portfolio Construction**

**Strategy Setup:**

The study employs three main categories: Long(L), Short(S), and Long-Short (LS) for both Equal Weighted (EW) and Value-Weighted (VW) portfolios

* **Long (L) Strategy**: invest in top 3 predicted stocks. (based on highest positive sentiment)
* **Short(S) Strategy**: invest in bottom 3 predicted stocks. (based on least positive sentiment)
* **Long-Short (LS) Strategy:** Combine long and short positions.

**Transaction Costs**

The study also considers transaction costs in portfolio returns.

* **Large Cap**: A transaction cost of 11.21 basis points.
* **Small Cap**: A transaction cost of 21.27 basis points.

**Sharpe Ratio Calculation**

The Sharpe Ratio, a measure of risk-adjusted return, is calculated for each strategy to evaluate the performance of portfolios. Higher Sharpe Ratios indicate better risk-adjusted returns.

**Evaluation**

Graphs and charts (**‘{user\_specified\_path}/LLMs/Finance/results/figures’** directory) provide a visual representation of portfolio performance.

**PARAMETER SETTINGS**

* **Learning Rate**: Typically set to ‘**2e-5’** for fine-tuning LLMs.
* **Batch Size**: Set to ‘**16’** for training efficiency.
* **Epochs**: Default is **3**, but the user can adjust as per model performance.

**TOOLS and PACKAGES**

Ensure the following packages with same versions are installed in your environment. Use ‘**requirements.txt**’ file to install all packages.

**General Libraries**

1. *Python – 3.10.12*
2. *numpy – 1.26.4*
3. *pandas – 2.1.4*
4. *matplotlib -3.7.1*
5. *seaborn – 0.13.1*
6. *scikit-learn – 1.3.2*
7. *plotly – 5.15.0*
8. *wrds (for accessing WRDS)*

**NLP and Deep Learning Libraries**:

1. *transformers – 4.24.4*
2. *torch – 2.4.0+cu121*
3. *nltk – 3.8.1*
4. *datasets – 2.21.0*
5. *tensorflow – 2.17.0*

**GPU SPECIFICATIONS**

* **A100** GPU:

Memory: 40GB to 80GB

* **L4** GPU:

Memory: 24 GB GDDR6

**NOTES**

* Data Access: Access to WRDS. Make sure you have the credentials.
* Computational Resources: Fine-tuning requires significant computational power. It is highly recommended to use a GPU-enabled environment.
* Logging and Checkpoints: All logs and model checkpoints are stored in the **‘{user\_specified\_path}/BM16\_LLMs/Finance/results/’** and **‘{user\_specified\_path}/BM16\_LLMs/Finance/models/fine\_tuned/’** directories.
* Data for Market Analysis is already present in the repo in the **‘BM16\_LLMs/finance/data’** directory as the data extraction process was manual.
* Due to size restrictions, data for transaction cannot be uploaded in Git. You can download the data from Drive and place it in the data directory.   
  **link:** <https://drive.google.com/drive/folders/116uYNtZZoTl29x4cIMU5czmt92m4brDR?usp=drive_link>

**LINK TO REPOSITORY**

Link: <https://github.com/Abhishek050898/BM16_LLMs>