CINNAMON AI

**2020 TAIWAN BOOTCAMP**

7th JULY 2020 – 14TH AUGUST 2020

**PROJECT FINAL REPORT:**

**INTEGRATING LANGUAGE MODEL WITH ASSET ALLOCATION MODEL**

**Group 1**

**Name and Email Address of Members:**

1. sam0213442dif@gmail.com

2. henry32144@hotmail.com

3. boriscat99@gmail.com

4. penguinwang@smail.nchu.edu.tw

**Date Submitted:**

*July 22, 2020*

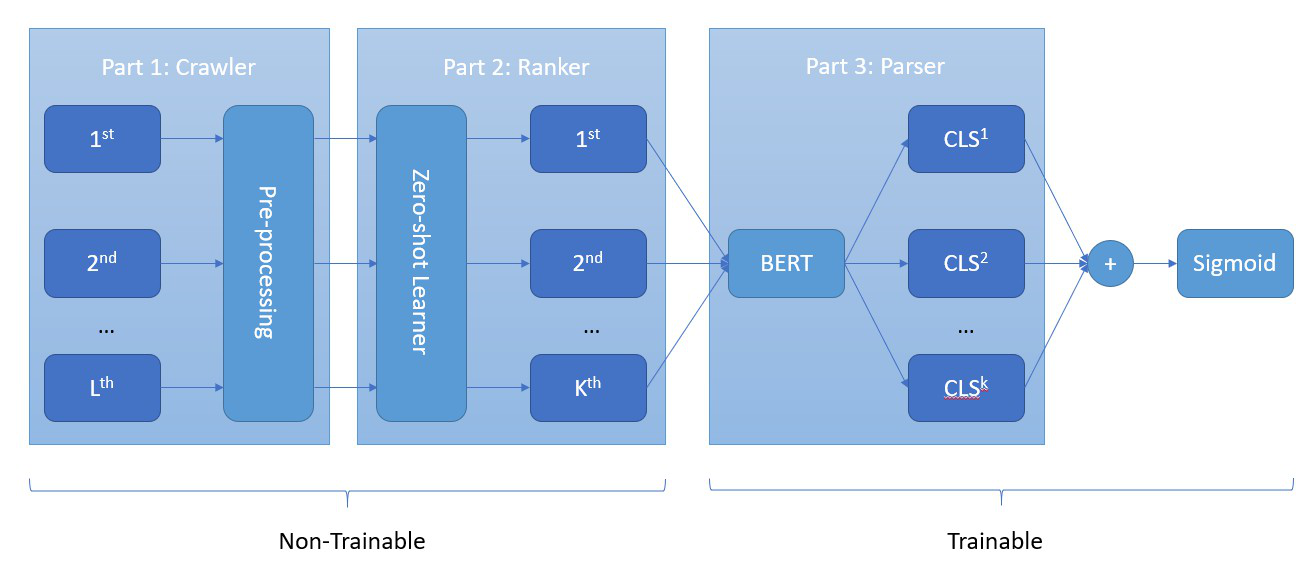
Table of Contents

(Your table of contents)

Introduction

The increasing demand and dominance of NLP makes it a target to financial market where investors hope trade execution be carried out automatically. This work addresses the problem by combining language model and asset allocation model. First, we utilize a smaller general-purpose language representation model called DistilBERT to extract sentence embedding for financial news of S&P500. Second, we exploit semantics from DistilBERT as unique views inputs for Black-Litterman model. The resulting news vector of returns leads to intuitive portfolios with sensible portfolio weights. One of the challenging parts is that there may be over hundreds of news on a daily basis, however, we cannot select all of them as our input to feed into our language model. Therefore, we conduct a zero-shot learner based on Sentence-BERT, and select the most crucial news related to finance and stock. Next, we design a siamese-like model which uses the same weights while working in tandem on several different financial news input to compute comparable output semantics vectors.

# Problem Statement



The overview of our language model is displayed above. The procedure is divided into three steps: (1) news crawler (2) news ranker (3) news parser.

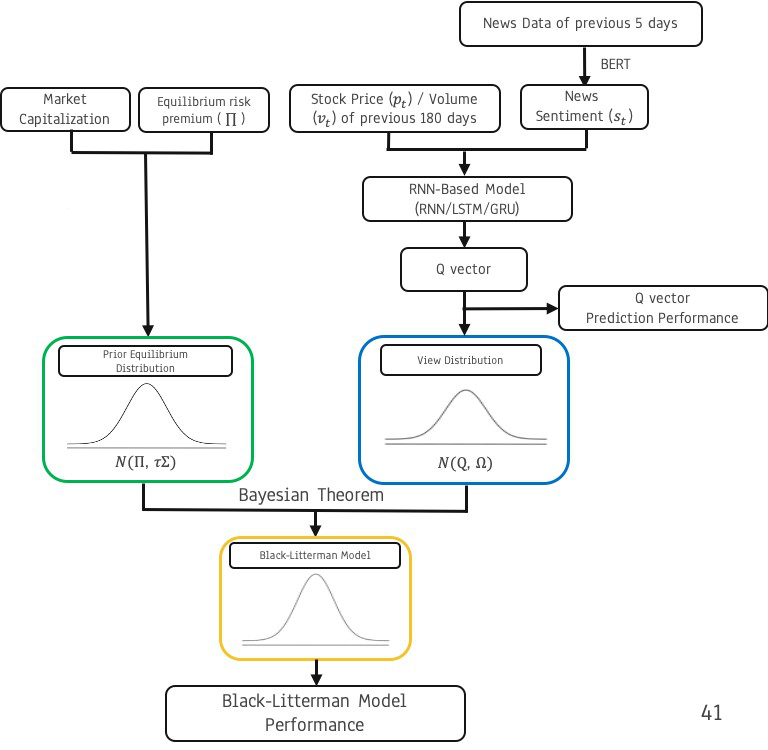
In order to get daily news data, we build a web crawler. Next, we then store them into database. To better select the most important financial news on a daily basis, we construct a ranking model. By doing so, we can rank the news documents based on cosine similarity. After top *k* news are filtered, those news documents are fed into language model to get their sentence embedding. Combining all sentence embedding, final representation of the top *k* news contains the semantic information from the most important news on the same day. And the output is the stock movement label telling whether the stock trade price is up or down after a certain time (we call it prediction delay).

# Object Model

The aim of this project is to provide recommendations for trading strategies using the results of DistilBERT model and Black Litterman model.

The main objectives of the system are:

* Utilize news sentiment from a language model as a feature to improve portfolio optimization



# Machine Intelligence and Methods

The main procedures of the system are:

* In the beginning, both selenium and beautifulsoup are used to crawl the news data from Reuters, which covers breaking news in markets, business, politics, entertainment and technology. Subsequently, for the news ranker, we utilize a modification of the pretrained BERT network called Sentence-BERT, which use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. For the final step, we propose a DistilBERT model to get news sentence embedding. Combining all sentence embedding, final representation of the top *k* news will contain the semantic information from the most important news on the same day.
* The Black Litterman model for portfolio optimization combines investors’ expectations with the Markowitz framework. The BL model is designed for investors with private information or knowledge of market behavior. In the subsequent stage, we propose a method where investors’ expectations are based on news sentiment extracting from language model. When compared to a market portfolio, a news sentiment-driven portfolio outperforms the market portfolio and the market index.

# Graphical User Interface

# System Testing and Evaluation

# Conclusions

# Lessons Learned & Recommendations

# Appendices

## Appendix A – Python Code of Your System

## Appendix B, C, D, … as needed.