Yong Kim, Ayse Ozdincer, Yi Xin Xiang
Word Count: 1396; Penalty: 0%

Project Proposal

Introduction

The stock price is sensitive to news publication, which accounts for 63.5% of substantial stock increases and 53.4% of decreases [1]. The goal of the project is to train a neural network that takes in a news article about a company and outputs a stock trend prediction. The model can be important for an automated trading system, which makes up 75% of US stock exchange transactions [2]. The algorithm can search through news articles, make predictions using the network and take the corresponding actions. A few seconds in entering orders can make a huge difference in the outcome, but traders may not necessarily see the news in time to respond, and this approach helps traders by processing articles instantly. Furthermore, the network will be trained using data that reflect true market reactions and include the decisions of all traders. This prevents individual emotions from disturbing a rational decision.

The vast amount of historical stock prices and news articles makes machine learning an appropriate tool as there are sufficient data to learn from. Moreover, automating the process of predicting stock trend from news using supervised learning allows traders to focus on more complicated tasks. More importantly, a person is not being directly affected by the model results, so there will be fewer controversies regarding machine learning usage.

Illustration / Figure

Step1. Raw data query

Input1: Company Name (Top 10 companies within each industry of S&P 500)

Input2: Critical stock deviation (± 3%)

Stock data API

- input: company name + news date range
- input: critical stock deviation (used for parsing of the received data)

-output: critical stock deviation dates

↓

Output:

- Company name (used as input)
- Critical date range (output)

Example

Stock_data (company = "AAPL", date_range = "6mo")

↓

result = [{Date : 2019-02-07, Market change : "3.9%"}
{Date : 2019-02-08, Market change : "-7.9%"}
{Date: 2019-02-09, Market change: "+2.3%"}
{Date: 2019-02-10 Market change: "+6.3%"} ...]

↓ Data parse using >+5% and <-5%

critical_dates = ["2019-02-08", "2019-02-10"...]

News article API

- input: company name
- input: date range

↓

Output:

list of news articles in paragraph format

Step2. Data Cleaning

Data Cleaning / Processing / Joining

- input: news article in paragraph
- input: stock deviation ([0] for decrease, [1] for increase)

NLTK Paragraph → Sentence tokenization
NLTK Sentence → Word tokenization
Google Word2Vec embedding

- output: Labelled dataset which includes;
  1. Article in series of google Word2Vec embedding
  2. Label of [0] or [1] for supervised learning

- data split: train_loader, valid_loader, test_loader
  0.6, 0.2, 0.2 ratio

Example

"Tesla failed to meet customer demand"

↓

Tesla	failed	to	meet	customer	demand

↓

2D tensor ([

0.28	0.28	0.28	0.28	0.28	0.28
0.48	0.48	0.48	0.48	0.48	0.48
1.24	1.24	1.24	1.24	1.24	1.24
-9.71	-9.71	-9.71	-9.71	-9.71	-9.71

...])

*row count is 300, as we use pretrained Google News vector of 300 sizing.

Label tensor ([

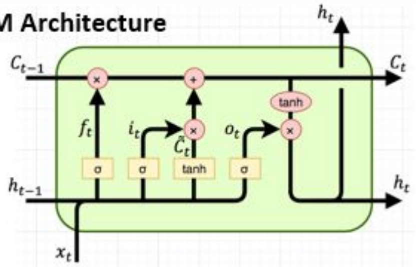
0	1	0	0	1	0

...])

*each cell indicates the stock trend of the overall article, not per single word or sentence. 1 represents +5% increase for the given news article and vice versa.

Step3. Training

LSTM Architecture



Parameters:

- input_size: 300
- hidden_size: 32
- num_layers: 1
- batch_first: True

*values prone to change depending on training result

Background & Related Work

Word Sense Disambiguation Application in Sentiment Analysis of News Headlines **by Saeed Seifollahi and Mehdi Shajari**
This model uses natural language processing in news headlines to predict movements in the FOREX market, specifically a currency pair. ‘Significant words’ in the headlines are identified to improve the determination of sentiment. The model inputs news headlines and EUR/USD exchange rates from the corresponding dates [3].

A Method of Measurement of The Impact of Japanese News on Stock Market **by Daisuke Katayama and Kazuhiko Tsudab**
This research focuses on the effects of news on Japanese companies stock and aims to obtain more efficient investment behaviour. A polarity dictionary is used to identify the polarity of the news article. Weights are assigned to both positive and negatively polarized words and averaged to find an overall categorization for the company [4].

Data Processing

Overall data objective

ML model’s expected input data must be in the following format (visualized in JSON format).

```
        {
                "news_article“ : tensor([…]),
                “label” : [0] or [1]
        }
```

Here, the tensor represents series of word2vec word embeddings, which composes the queried news article [Figure 1]. To reach the above result, the raw data of both news articles and stock market must be queried first.

Raw source data

Following APIs were used to construct base-data;

Table 1. APIs usage for raw data retrieval

REST API Title	Queries and input parameters
IEX Stock Market API https://iextrading.com/developer/docs/	Returns historical stock data. **Stock_symbol** stock/{NYSE_symbol} **Date_range** stock/{NYSE_symbol}/chart/{date_range} - 3 months. Prone to modification based on obtained data. **Output** Critical_date - Any date within the recent 3 months, where the company’s stock value has either increased or decreased by a value higher than the critical stock deviation (3%). Label - Either [1] or [0] depending on stock increase/decrease
News API https://newsapi.org/.	Returns news articles. **Keyword** q = {company name} - Top 10 companies within each S&P 500 industry. **Date_published** from = {critical date - 1}, to = {critical date} - One day interval is used. **Language** Language = en **Sorting_factor** sortBy = relevancy

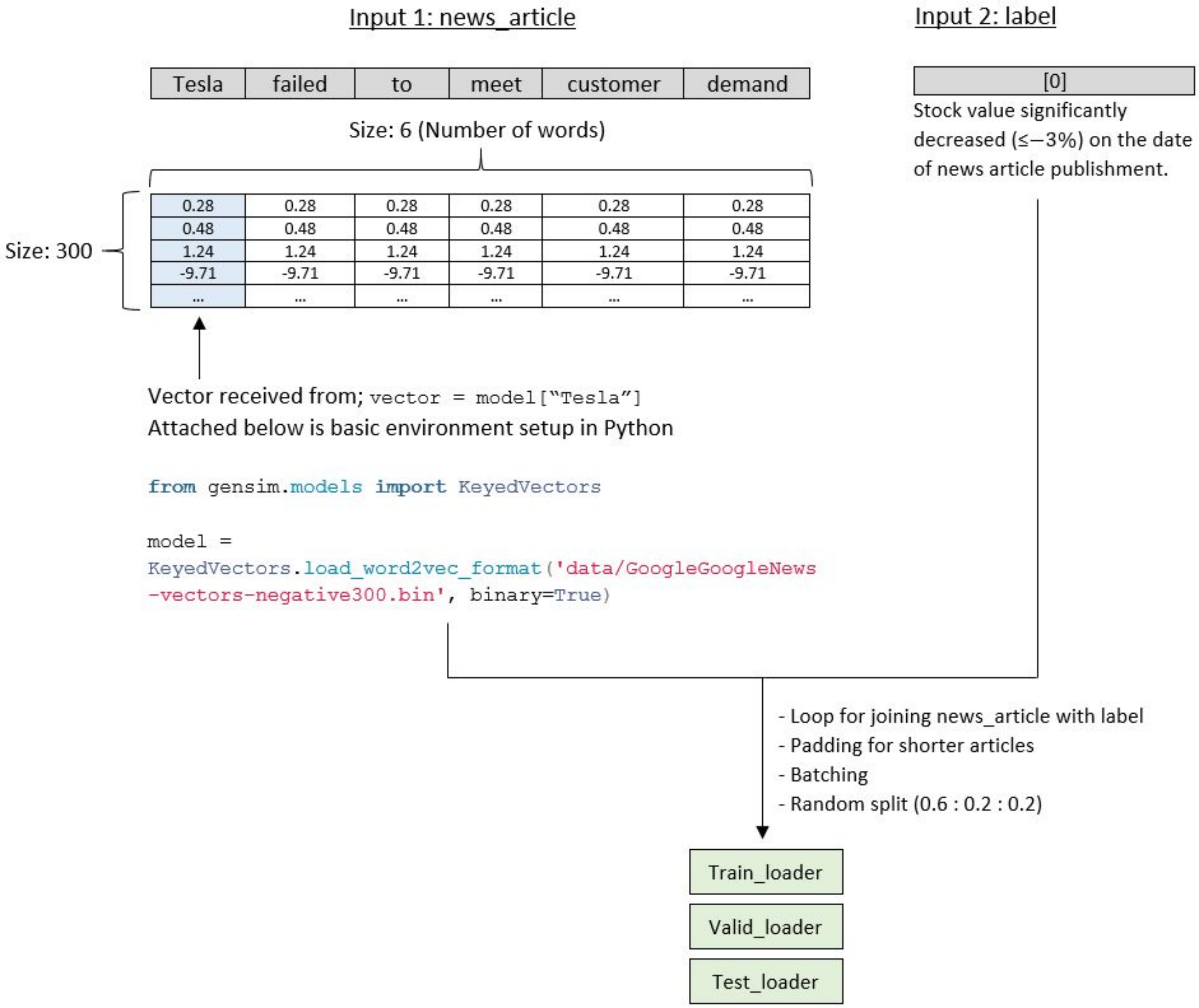
Companies to query

The top 10 companies, ranked by market capitalization, within each S&P 500 industry were chosen to simplify the problem while capturing broad coverage of the U.S. market trend [5]. This also provides an additional benefit of diverse contextual data (news articles) for model generalization.

Word to embedding conversion

Following the news article query, a word tokenizer is applied, which then undergoes GoogleNews-300 word2vec pre-trained model to convert its words to embedding of 2D size 300 x N(word) tensor, where N represents the number of words per article. This intermediate result will then be wrapped into a dataloader function as a desired input to the ML model. Note that stop words are not removed, as retaining such value might suit better for our specific needs [6].

Figure 1. Data pipeline for raw data to dataloader conversion



Tesla	failed	to	meet	customer	demand
0.28	0.28	0.28	0.28	0.28	0.28
0.48	0.48	0.48	0.48	0.48	0.48
1.24	1.24	1.24	1.24	1.24	1.24
-9.71	-9.71	-9.71	-9.71	-9.71	-9.71
...	...	...	...	...	...

Stock value significantly decreased (≤−3%) on the date of news article publishment.

Vector received from; `vector = model[“Tesla”]`
Attached below is basic environment setup in Python

```
from gensim.models import KeyedVectors

model =
KeyedVectors.load_word2vec_format('data/GoogleGoogleNews
-vectors-negative300.bin', binary=True)
```

- Loop for joining news_article with label
- Padding for shorter articles
- Batching
- Random split (0.6 : 0.2 : 0.2)

Architecture

RNN architecture

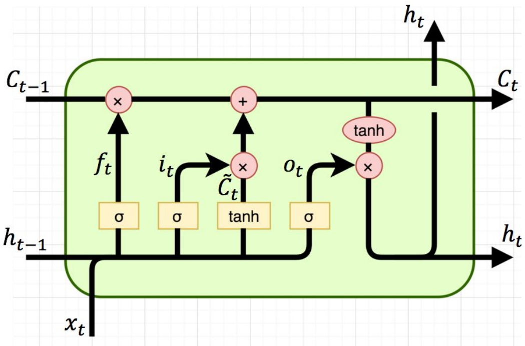
RNN is one of the most common and high-performance architectures that many machine learning scientists apply to solve NLP problems [7]. Specifically, LSTM variant is chosen as it excels in retaining data from earlier sequence [8]. Initial and potential hyperparameters are attached below [Table 2].

Table 2. LSTM parameters [9]

Parameters	Values
input_size	300 Restricted to 300, due to the usage of GoogleNews-300 vector.
***hidden_size**	32 32 selected for initial training [10].
***num_layers**	1 (default)
***bias**	False (default)
batch_first	True
***Default training hyperparameters**	Batch_size Learning_rate Epoch Loss (CrossEntropyLoss) Optimizer (Adam)

*Potential hyperparameters

Figure 2. LSTM architecture visualization



Input data

Input data is in dataloader format, composed of tensors and labels. For details, please refer to “word to embedding section” above.

Transfer learning

Transfer learning with GoogleNews-300 model is used, as it’s already trained based on news platform [11]. Moreover, the model can take advantage of the broad word coverage (3 million words, including financial domain-languages), and its marginal dataset used for training (100 billion words) [11].

Baseline Model

The baseline model involves obtaining the GoogleNews vector embedding for all the words in an article and taking an average. This averaged embedding will be passed through a simple fully-connected neural network (input size of 300) to get a prediction of the influence of this article on the company. This process will be repeated for all articles in the test set.

Ethical Considerations

Bias in the news inputs will be prominent. Each source's opinion is reflected in news based on their political standing, geographic location, religious beliefs or self-profitability. Incorporating a wide range of media sources will help minimize bias.

An issue with the model’s decisions will be that it will not consider the impact of investing in “unethical” companies on the investor’s publicity, for example, investing in a weapons manufacturer or a tobacco company.

Project Plan

Table 3. Project Plan

Meeting Time	● Tuesdays 6-8pm ● Extra meetings with upcoming deadlines, if needed
Communication with TA	● Email ● Meetings may be arranged with agreement from the TA.
Team Communication	● Messenger group chat
Document Writing	● Google Docs/ Slides for writing deliverables ● Shared Google Drive folder for storage
Code Writing	● Google Colab will be used in junction with Github for version control and tracking contribution. ● Team members will fork the project repository. ● Pull requests will be used, and others will review before merging to prevent overwriting code or introducing bugs.
Timeline & Task Division	Two internal deadlines: ● To ensure that the team can ask more technical questions, by the progress meeting, the team should have finished data collection and have started building the baseline model and RNN. ● The model training and tuning should be completed/ very close to complete on July 26th to leave time for final term test studying. Refer to the Gantt chart for more details.

Table 4. Gantt Chart

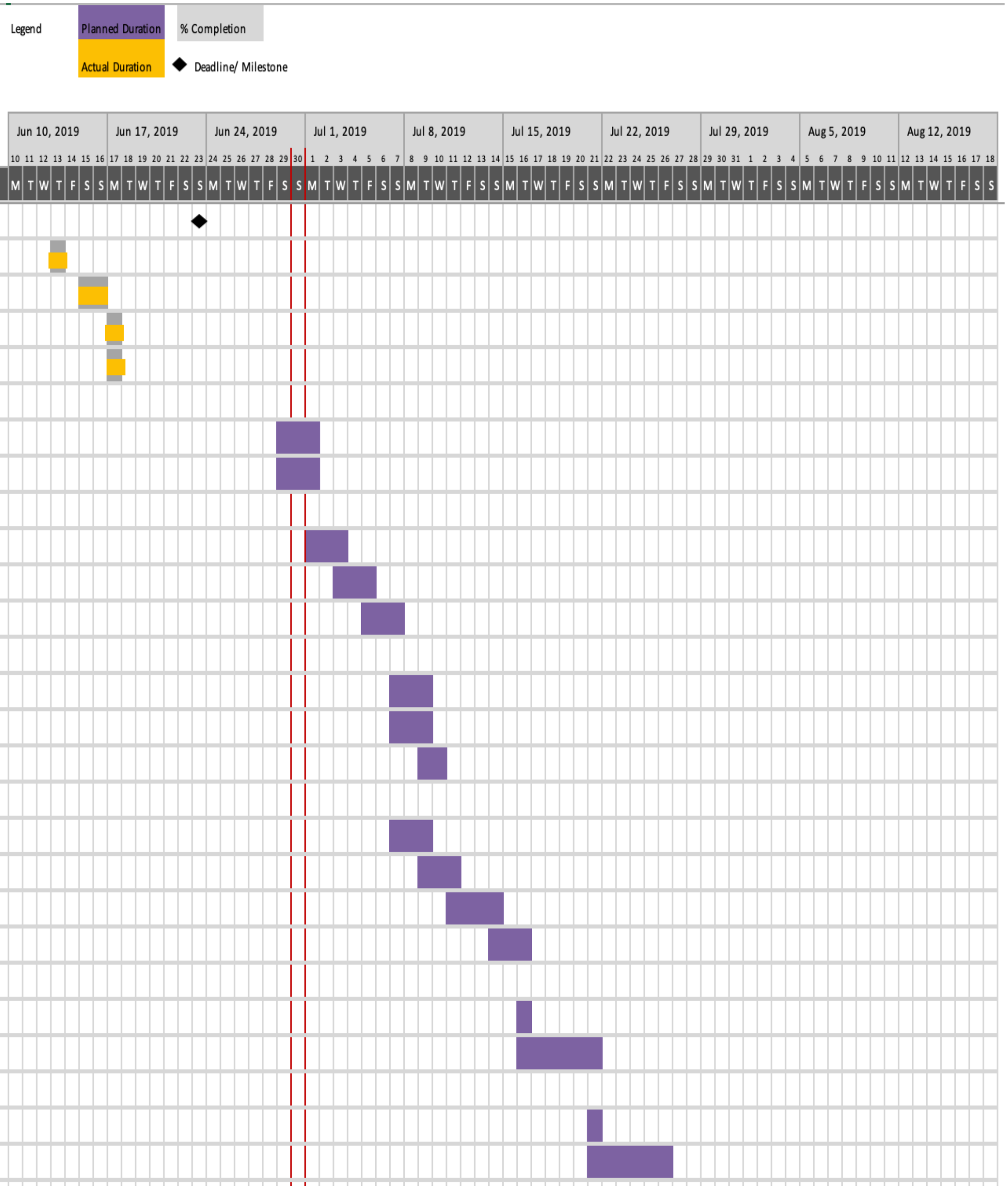
APS360 Project

Yong (Kyle) Kim, Ayse Ozdincer, Yi Xin (Gloria) Xiang

Project Start: Thu, 6-13-2019

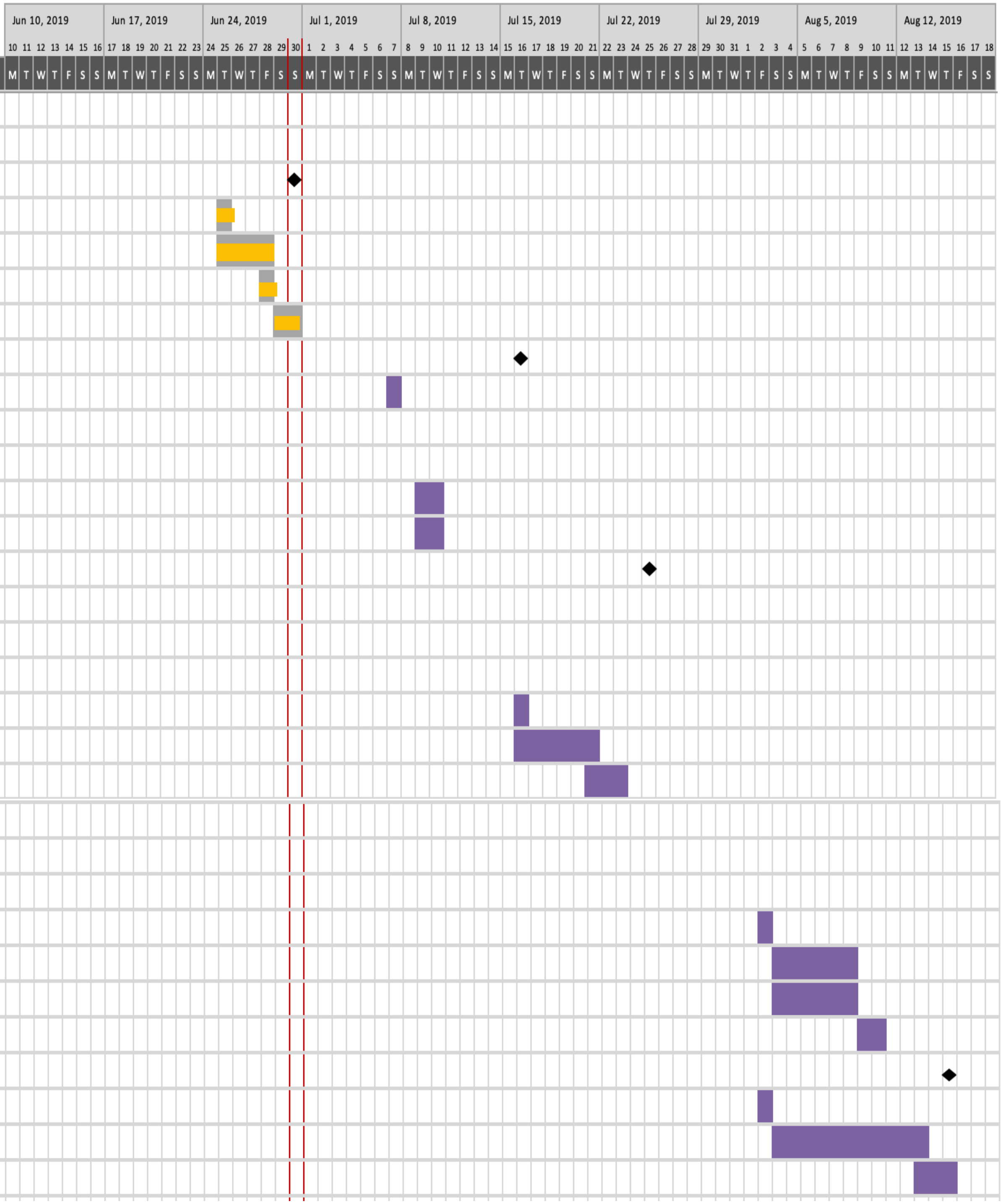
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Team Formation & Project Brainstorming				
Form team	All	100%	6-13-19	6-13-19
Generate project ideas (2 per person)	All	100%	6-15-19	6-16-19
Finalize project idea	All	100%	6-17-19	6-17-19
Send uniqueness approval email	Kyle	100%	6-17-19	6-17-19
Data Collection & Cleaning				
Determnine the critical stock deviation	All	0%	6-29-19	7-1-19
Determine the data range that news will be selected from	All	0%	6-29-19	7-1-19
Repeat the processes above until sufficient data can be gathered		0%		
Write code to gather raw news articles	Kyle	0%	7-1-19	7-3-19
Assign labels	Gloria	0%	7-3-19	7-5-19
Join/ structure data & split training, validation & test sets	Ayse	0%	7-5-19	7-7-19
Baseline Model				
Process data for baseline model (eg. obtain average embeddings for articles, etc)	Kyle	0%	7-7-19	7-9-19
Write code to build baseline model	Gloria	0%	7-7-19	7-9-19
Input test set to obtain a baseline performance	Ayse	0%	7-9-19	7-10-19
RNN Training				
Write code for building the RNN	Ayse	0%	7-7-19	7-9-19
Write training code	Kyle	0%	7-9-19	7-11-19
Debug with a small debug set	Gloria	0%	7-11-19	7-14-19
Train the network	Kyle	0%	7-14-19	7-16-19
Hyperparameter Tuning				
Decide on which hyperparameters to tune & assign each person several hyperparamet	All	0%	7-16-19	7-16-19
Tune the chosen hyperparameters & add new ones if needed	All	0%	7-16-19	7-21-19
Performance Evaluation				
Use test set to obtain accuracy	Ayse	0%	7-21-19	7-21-19
Compare with baseline model & keep training/ tuning hyperparameters if needed	All	0%	7-21-19	7-26-19



Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Course Deliverables				
Project Proposal				
Distribute proposal sections for team members	All	100%	6-25-19	6-25-19
Write individual sections (First draft)	All	100%	6-25-19	6-28-19
Group discussion on project planning	All	100%	6-28-19	6-28-19
Proofread (Final draft) & submit	All	100%	6-29-19	6-30-19
Progress Meeting				
Book the meeting with TA	Gloria	0%	7-7-19	7-7-19
Finish data collection	Refer above (Data Collection & Cleaning)			
Have started building the baseline model and RNN	Refer above (Baseline Model & RNN Training)			
Prepare technical questions to ask the TA	All	0%	7-9-19	7-10-19
Prepare progress presentation to the TA	All	0%	7-9-19	7-10-19
Progress Report				
Finish data collection	Refer above (Data Collection & Cleaning)			
Finish baseline model	Refer above (Baseline Model)			
Have at least one result from training the RNN	Refer above (RNN Training)			
Distribute report sections for team members	All	0%	7-16-19	7-16-19
Write individual sections (First draft; include updates since proposal/ progress meeting)	All	0%	7-16-19	7-21-19
Proofread (Final draft) & submit	All	0%	7-21-19	7-23-19
Project Presentation				
Finish training the RNN	Refer above (RNN Training)			
Obtain a test accuracy higher than baseline model	Refer above (Performance Evaluation)			
Divide presentation roles	All	0%	8-2-19	8-2-19
Prepare powerpoint presentation	All	0%	8-3-19	8-8-19
Prepare speech	All	0%	8-3-19	8-8-19
Finalize & practice	All	0%	8-9-19	8-11-19
Project Report				
Distribute report sections for team members	All	0%	8-2-19	8-2-19
Write individual sections (First draft)	All	0%	8-3-19	8-13-19
Proofread (Final draft) & submit	All	0%	8-13-19	8-15-19



Risk Register

Model’s predictions aren’t accurate enough:

Having a success rate lower than desired is a big risk. This is highly likely especially in the beginning stages of using the model. The following example modifications to the model will be made until the predictions are optimized:

- Changing the time period of the query (only inputting immediate news before change in market or news from the preceding month)
- Changing the query sources
- Changing critical stock deviation.

Impact of news and media bias is reflected in the model's decisions:

The opinions of more politically driven and biased sources can be reflected in the model’s decision making. In this case, the news sources would be changed or the more sources will be used to increase diversity in input.

Model takes too long to train:

Due to the complexity of the model, training time could be a large risk. The approach to this issue will be to try to simplify the model and using CUDA to speed up the computing.

Data volumes & duration conflict:

The News API allows free query for articles that are published in the past month. This might introduce issues due to the omission of other major events, like the release of financial reports, that impact stock market behaviour. Older new articles may be queried manually if needed.

Colab: https://colab.research.google.com/drive/1tcQhuO8TXYGIsXHV8C7tIm6_JtR3Kt2G

Github: https://github.com/kyle-yong-kim/News_Stock_Prediction

Reference

[1] W. Dolde *et al.*, (2002, December). *Evidence that Extreme Volatility in Stock Prices is Associated with Reported News Items*. [Online]. Available: http://dx.doi.org/10.2139/ssrn.334602

[2] J. Folger. (2019, May 12). *Automated Trading Systems: The Pros and Cons.* [Online]. Available: https://www.investopedia.com/articles/trading/11/automated-trading-systems.asp

[3] “Word sense disambiguation application in sentiment analysis of news headlines: an applied approach to FOREX market prediction”, SpringerLink, 2019. [Online]. Available: https://link-springer-com.myaccess.library.utoronto.ca/article/10.1007%2Fs10844-018-0504-9#Sec17

[4] “A Method of Measurement of The Impact of Japanese News on Stock Market”, ScienceDirect , 2018. [Online]. Available: https://www-sciencedirect-com.myaccess.library.utoronto.ca/science/article/pii/S1877050918313620.

[5] "Why do investors use the S&P 500 as a benchmark?", *Investopedia*, 2018. [Online]. Available: https://www.investopedia.com/ask/answers/041315/what-are-pros-and-cons-using-sp-500-benchmark.asp. [Accessed: 28- Jun- 2019].

[6] "Why is removing stop words not always a good idea", *Medium*, 2019. [Online]. Available: https://medium.com/@wilamelima/why-is-removing-stop-words-not-always-a-good-idea-c8d35bd77214. [Accessed: 27- Jun- 2019].

[7] J. Brownlee, "When to Use MLP, CNN, and RNN Neural Networks", *Machine Learning Mastery*, 2018. [Online]. Available: https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/. [Accessed: 28- Jun- 2019].

[8] "LSTM Networks for Sentiment Analysis — DeepLearning 0.1 documentation", *Deeplearning.net*. [Online]. Available: http://deeplearning.net/tutorial/lstm.html. [Accessed: 28- Jun- 2019].

[9] "torch.nn — PyTorch master documentation", *Pytorch.org*. [Online]. Available: https://pytorch.org/docs/stable/nn.html. [Accessed: 25- Jun- 2019].

[10] "input size and hidden size for rnn", *Piazza.com*, 2019. [Online]. Available: https://piazza.com/class/jv9syktmvhc1on?cid=112. [Accessed: 28- Jun- 2019].

[11] "word2vec", *Code.google.com*, 2013. [Online]. Available: https://code.google.com/archive/p/word2vec/. [Accessed: 27- Jun- 2019].