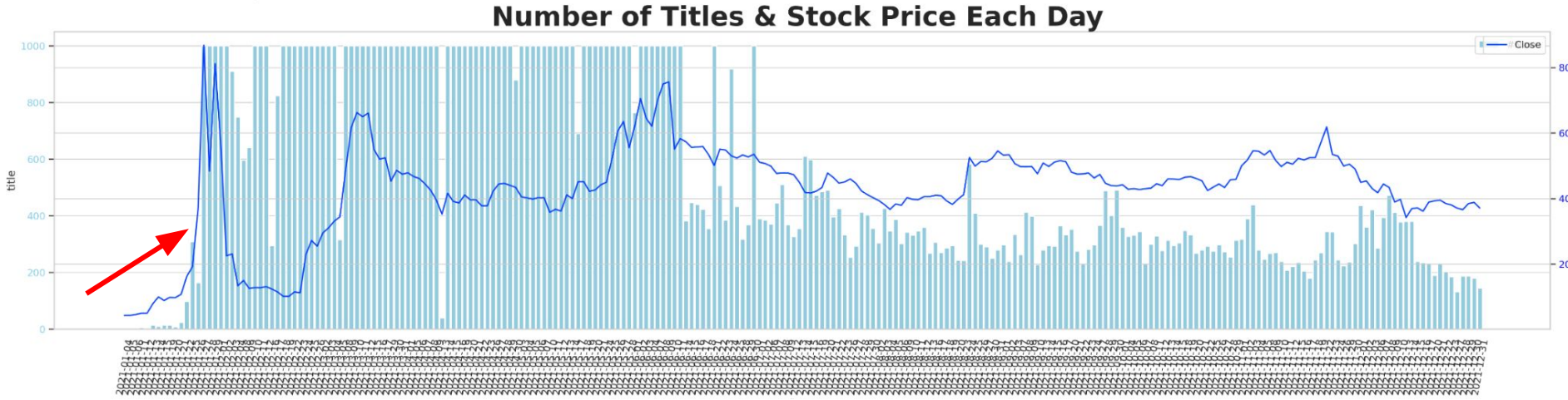
GME Stock Forecasting with NLP

Joey Xie

Background

In late January 2021, GameStop (GME) became the center of a financial phenomenon known as a 'short squeeze.' A surge of retail investors, coordinating through social media platforms like **Reddit**'s r/wallstreetbets, began buying up GameStop's stock. This drove up the stock price dramatically. It gives us a great opportunity to study if we can utilize **NLP** techniques for analyzing Reddit post titles and to enhance stock prediction.



Experiments

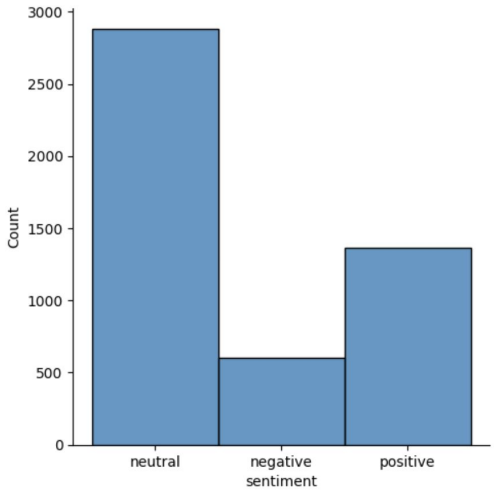
1. Fine Tune Bert-base with kaggle financial news dataset to get a language model that can be used in our NLP task
2. Experiment with three way to integrate NLP features:

- Train LSTM with stock time series + **sentiment labels**
- Train LSTM with stock time series + **TF-IDF embeddings**
- Train LSTM with stock time series + **Bert embeddings**

Fine tune Bert with Kaggle Financial news dataset

	sentiment	news
2659	neutral	The order comprises all production lines for a...
2424	neutral	CapMan said the deal 's effect on its cash flo...
3120	neutral	Simultaneously , Alma Media has purchased a 35...
3528	neutral	An Apple spokeswoman said the company declined...
2449	neutral	Exel is headquartered in Mantyharju in Finland .
...	...	...
211	positive	In addition , Kone signed a two-year maintenan...
1768	positive	Tieto offers Aktia a good foundation and the r...
952	positive	Finnish software developer Basware Oyj said on...
31	positive	The company 's net profit rose 11.4 % on the y...
4499	positive	W+Ærtsil+Æ 's solution has been selected for i...

8637 rows × 2 columns



This dataset contains financial news titles and the sentiment label, which can be used to fine tune our base Bert model to better **adapt to the financial domain** for later use. Note that the classes neutral, negative, positive are highly imbalanced so we perform **upsampling** before training.

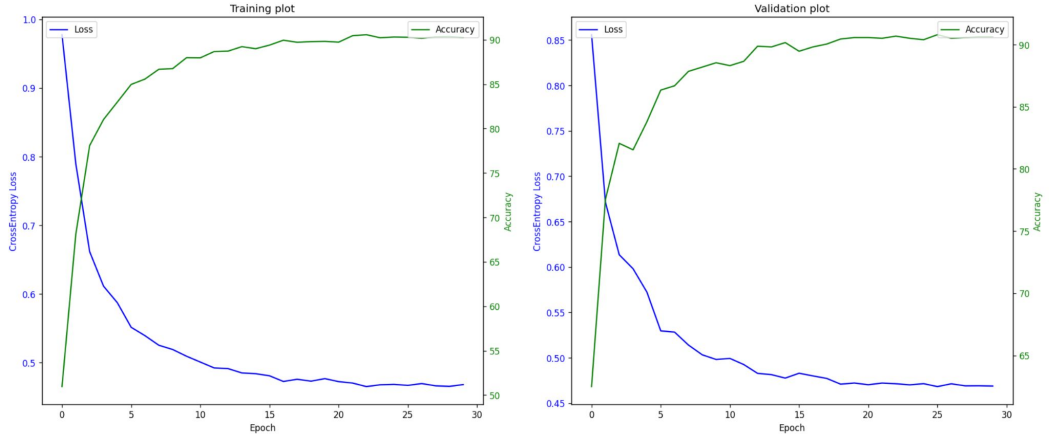
Model Training and Eval

```
from transformers import BertTokenizer, BertModel
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
bert = BertModel.from_pretrained("bert-base-uncased").to(device)
```

```
class BERT(nn.Module):
    def __init__(self, bert):
        super(BERT, self).__init__()
        self.bert = bert
        self.mlp = nn.Sequential(
            nn.Linear(768,256),
            nn.LeakyReLU(),
            nn.Linear(256,256),
            nn.LeakyReLU(),
            nn.Linear(256,3)
        )

    def forward(self, x):
        input_ids= x[0].to(device)
        token_type= x[1].to(device)
        att_mask= x[2].to(device)
        out = self.bert(input_ids.to(device),  attention_mask=att_mask.to(device), token_type_ids=token_type.to(device))

        return self.mlp(out.pooler_output)
```



We used the Bert base uncased model from hugging face, and add a **MLP** of sizes(768, 256, 3) to map the output to three classes, **neutral, positive, and negative**. The learning rate is set to 1e-5 for Bert and 1e-3 for MLP.

And from the curves we can see that the training and validation accuracy goes to **90%+.**

Then we save the model to use it in the later time series forecasting tasks.

Scrape S&P 500 stock with yahoo finance

	0
0	MMM
1	AOS
2	ABT
3	ABBV
4	ACN
...	...
499	ZBRA
500	ZBH
501	ZION
502	ZTS
503	GME

504 rows × 1 columns

We scrape all stock information (**Open, high, low, close, volume, adj close**) of 504 stocks from 2018-01-01 to today, manually including GME(kicked out of S&P 500) to prepare for training.

Some of the stock got bankrupted, so we delete them.

In our actual training process, we use only **GME** to train the model.

Ticker	MMM						AOS				...
Attribute	Open	High	Low	Close	Adj Close	Volume	Open	High	Low	Close	...
Date											
2018-01-02	235.779999	237.070007	232.809998	235.639999	187.720306	2931000	61.450001	61.610001	61.040001	61.529999	...
2018-01-03	235.070007	235.729996	233.289993	235.630005	187.712326	2193700	61.599998	61.990002	61.290001	61.900002	...
2018-01-04	237.000000	239.440002	236.470001	238.710007	190.165985	2243100	62.000000	62.400002	61.820000	62.189999	...
2018-01-05	238.649994	240.899994	237.740005	240.570007	191.647751	1835900	62.410000	63.000000	62.369999	62.990002	...
2018-01-08	239.380005	240.940002	239.179993	239.789993	191.026367	1869000	62.880001	63.400002	62.599998	63.290001	...
...	...	...	...	...	...	...	...	...	...	...	...
2024-02-06	92.739998	93.849998	92.400002	93.760002	93.760002	3723400	77.169998	78.459999	77.099998	78.220001	...
2024-02-07	94.599998	94.709999	93.500000	93.839996	93.839996	3780900	79.019997	80.410004	78.919998	79.839996	...
2024-02-08	94.150002	94.379997	92.709999	93.199997	93.199997	3964300	80.220001	80.820000	79.730003	80.250000	...
2024-02-09	93.199997	93.300003	92.389999	92.900002	92.900002	3667100	80.290001	80.800003	79.980003	80.540001	...
2024-02-12	93.120003	95.059998	93.050003	94.629997	94.629997	4089500	80.760002	81.610001	80.430000	81.279999	...

1538 rows × 2988 columns

Reddit dataset preprocessing

	date	title	score	num_comments
0	2021-01-04	You NEED to see this about GME 🚀🚀🚀🚀🚀🚀	1.0	9.0
1	2021-01-04	Short Squeeze Incoming 🚀🚀🚀🚀🚀🚀🚀	1.0	1.0
2	2021-01-05	THIS CONVINCED ME TO ALL IN 💰GME (EXTREME PUMP...	1.0	6.0
3	2021-01-05	You already know what we must do brothers and ...	1.0	4.0
4	2021-01-06	ICR conference (11th Jan)	1.0	10.0
...	...	...	...	...
1033231	2021-12-31	Hedgies are relics of past generations- they h...	1.0	0.0
1033232	2021-12-31	Snapchat on 🔥🔥🔥	1.0	0.0
1033233	2021-12-31	MOASS has ruined me this year. But I will stil...	1.0	0.0
1033234	2021-12-31	Patterns for 2022???	1.0	0.0
1033235	2021-12-31	When GWagon	1.0	0.0

1033236 rows × 4 columns

Each day sample 300 news

```
[45] df_news = df_reddit.groupby('date').apply(lambda x: x.sample(min(len(x), 300))).reset_index(drop=True)
     df_news
```

	date	title	score	num_comments
0	2021-01-04	you need to see this about gme rocket rocket...	1.0	9.0
1	2021-01-04	short squeeze incoming rocket rocket rocket...	1.0	1.0
2	2021-01-05	this convinced me to all in money_bag gme ext...	1.0	6.0
3	2021-01-05	you already know what we must do brothers and ...	1.0	4.0
4	2021-01-06	hey guys we have a free discord channel thats ...	1.0	15.0
...	...	...	...	...
170322	2021-12-31	2021 timeline	1.0	0.0
170323	2021-12-31	happy new year apes	1.0	0.0
170324	2021-12-31	hold gme moon was testing out my word per minu...	1.0	0.0
170325	2021-12-31	i got my cs letter today time for account veri...	1.0	0.0
170326	2021-12-31	short interest has fallen 5 doesnt matter 165...	1.0	0.0

170327 rows × 4 columns

We use the post titles from the Reddit dataset about the stock GME.

We first demojize the emoji in titles to text. E.g. 🚀 is converted into **rocket**, because it contains very important information about the stock. And clean the title by converting to lowercase and keep only words.

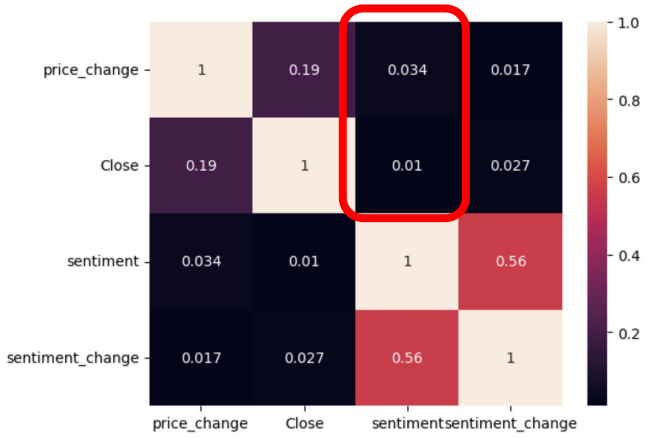
From the statistics, some days have number of titles >10k+, most of the days have around 300 titles. Therefore, we for each day **sample 300** titles, which will be utilized later in NLP.

Time Series + Bert Sentiment Analysis(Baseline)

inferencing...: 100% 199/199 [04:06<00:00, 1.06s/it]

	title	score	num_comments	Date	Open	High	Low	Close	Volume	sentiment
0	short squeeze incoming rocket rocket rocket...	1.0	1.0	2021-01-04	4.750000	4.775000	4.287500	4.3125	40090000	0
1	you need to see this about gme rocket rocket...	1.0	9.0	2021-01-04	4.750000	4.775000	4.287500	4.3125	40090000	0
2	this convinced me to all in money_bag gme ext...	1.0	6.0	2021-01-05	4.337500	4.520000	4.307500	4.3425	19846000	0
3	you already know what we must do brothers and ...	1.0	4.0	2021-01-05	4.337500	4.520000	4.307500	4.3425	19846000	0
4	icr conference 11th jan	1.0	10.0	2021-01-06	4.335000	4.745000	4.332500	4.5900	24224800	0
...	...	...	...	...	...	...	...	...	...	...

	count
negative	14163
neutral	49391
positive	1479



We used the Bert we fine tuned, and output sentiment labels on each Reddit title.

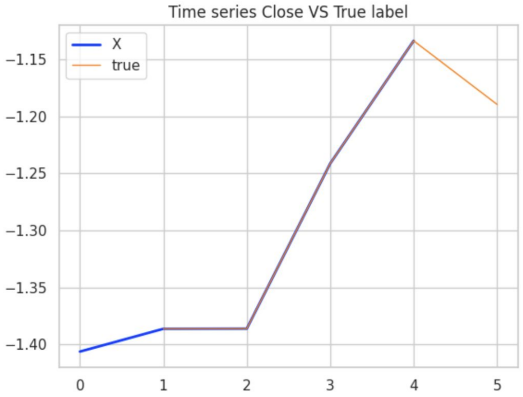
However, from the heatmap, the sentiment analysis our fine-tuned Bert gives does not give a good **correlation** with price or price change. Therefore the sentiment is **not going to be so useful**. This can be due to several reasons:

1.Our Bert is fine tuned on **kaggle financial news** headlines, which is drastically different from our **Reddit** post, and the model fails to generalize to Reddit post.

2.Our model mostly outputs sentiment as **neutral**, if we could have included only positive and negative classes, or simply using the embeddings, it would give us stronger correlation with the stock price.

But it serves as a good baseline methods, we'll train LSTM with sentiment labels first, and then explore Tf-idf embeddings, finally explore bert embeddings.

Time Series + Bert Sentiment Analysis(Baseline)



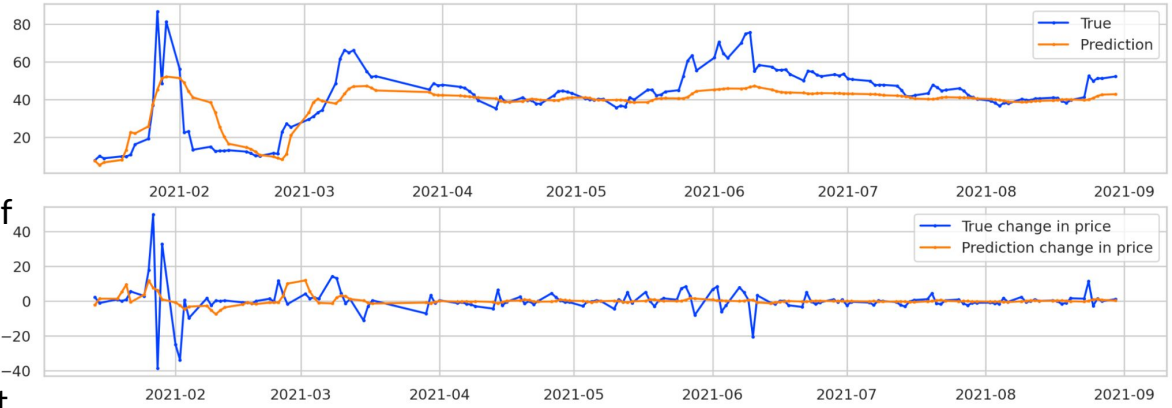
Time Series data sample is constructed in **sliding window** manner. With last 5 days as a look back window and predict tomorrow's price.

The feature dimension is 6 =5(open, high, low, close, volume)+1(average sentiment each day)

From the prediction graph, we can see the model **perform really bad,** by predicting almost horizontal line and barely follow the trend. Probably because we have a significant amount of titles being classified as neutral.

Although the Correlation Coefficient for the whole time series is high, we cannot trust that number! The deserialized Correlation Coefficient of **change in stock price** is **negative,** which means the model fail to predict the change in stock price, and even reversely predict the up and down. So we are going to explore the other two methods.

1/1/2021-8/31/2021 Prediction VS True Price and Price change(Return)



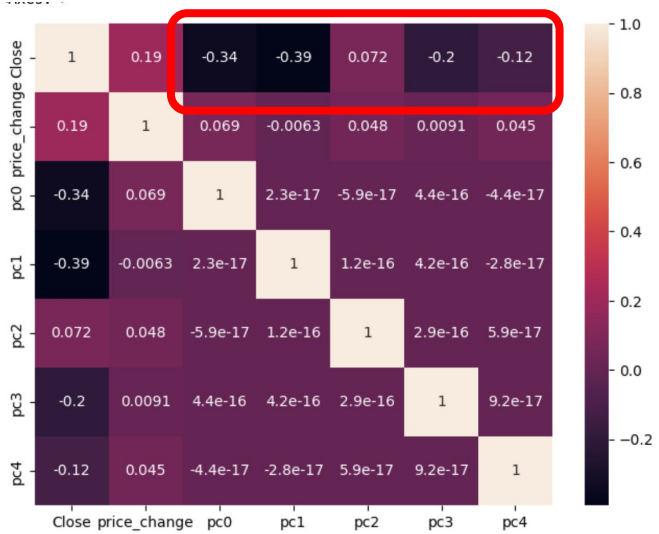
	MSE	RMSE	MAE	MAPE	Serial Corr	Pearson Corr(IC),	Spearman Corr(IC)
0	70.697731	8.408194	6.030578	0.138628	0.872062	-0.168163	-0.116123

Time Series + TF-IDF embeddings

Date	title
2021-01-04	short squeeze incoming rocket rocket rocket...
2021-01-05	this convinced me to all in money_bag gme ext...
2021-01-06	icr conference 11th jan. hey guys we have a fr...
2021-01-11	ryan cohen not being paid. buckle your seatbel...
2021-01-12	questions from a potential investor about game...
...	...
2021-12-27	ryan cohen on twitter. rocket rocket we had...
2021-12-28	right clicking all the nfts. throw the book at...
2021-12-29	so banks charge overdraft fees when you dont h...
2021-12-30	gamestop is doing something much bigger than j...
2021-12-31	happy new year apes painting by me spot the ea...

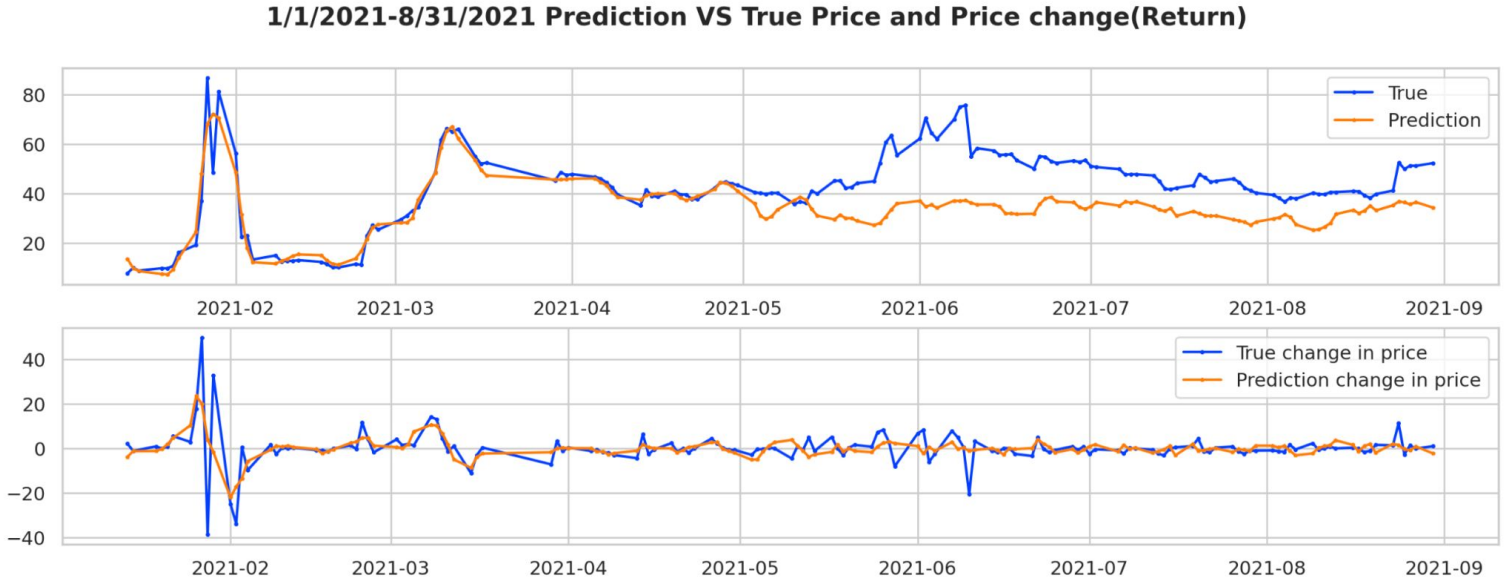
240 rows × 1 columns

In this section, we construct the dataset by combining titles in each day as one document, so that the entire dataset is a giant corpus, and each day has a document. The we use **TFIDF Vectorizer** to turn each day's document into embeddings, and perform **PCA** to reduce 19k dimension vectors to 200 dimensions while keeping a explained variance ratio 91%+.



We can see from the heatmap that, the first few **principal components correlate strongly**, although negatively with the **stock closing price**, which indicates we can try to utilize them as predictors of the stock price.

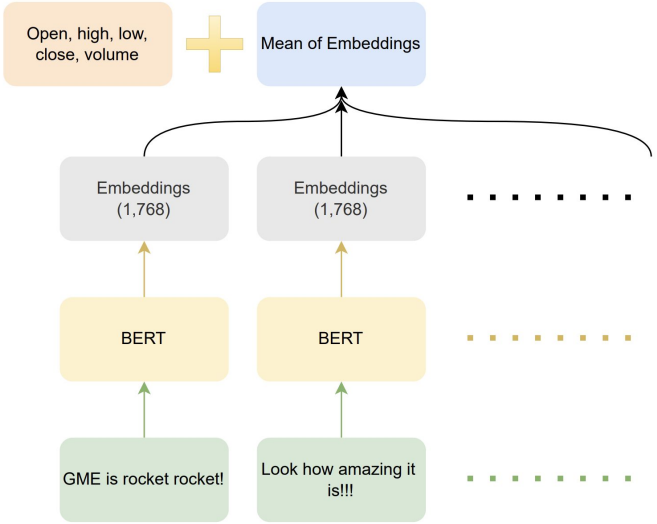
Time Series + TF-IDF embeddings



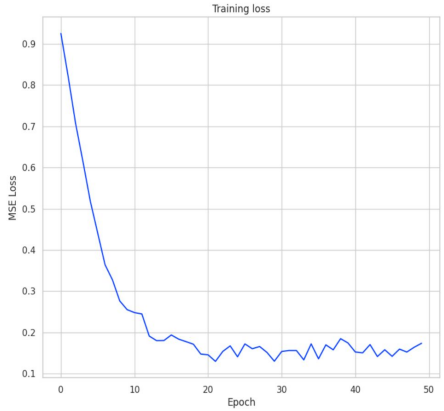
	MSE	RMSE	MAE	MAPE	Serial Corr	Pearson Corr(IC),	Spearman Corr(IC)
0	239.582047	15.478438	14.070969	0.417722	0.633061	0.106105	0.082686

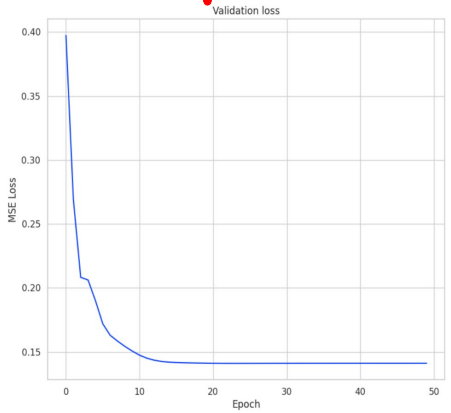
From the predictions plot, we can see that the predicted price is far off from the true price on the test data(2021-06 to 2021-09), as we can see from the large RMSE. But the predicted **change in price** can more or less follow the true change in price, which is a big **improvement** than using solely sentiment labels. Also we can see this from the metrics that, the correlation coefficient of stock returns is 10% which is a decent IC number in quantitative trading.

Time Series + Bert embeddings



Date	Close	High	Low	Open	Volume	dim0	...	dim765	dim766	dim767
2021-01-04	4.312500	4.775000	4.287500	4.750000	40090000.0	-0.839527	...	0.602982	-0.601221	0.744309
2021-01-05	4.342500	4.520000	4.307500	4.337500	19846000.0	-0.645647	...	0.098632	-0.463653	0.738352
2021-01-06	4.590000	4.745000	4.332500	4.335000	24224800.0	-0.628741	...	0.663287	-0.467869	0.636216
2021-01-11	4.985000	5.162500	4.752500	4.852500	59632000.0	-0.802747	...	0.025344	-0.559318	0.834198
2021-01-12	4.987500	5.100000	4.830000	4.990000	28242800.0	-0.769426	...	0.258986	-0.452421	0.703244
...	...	...	...	...	...	...	...	...	...	...

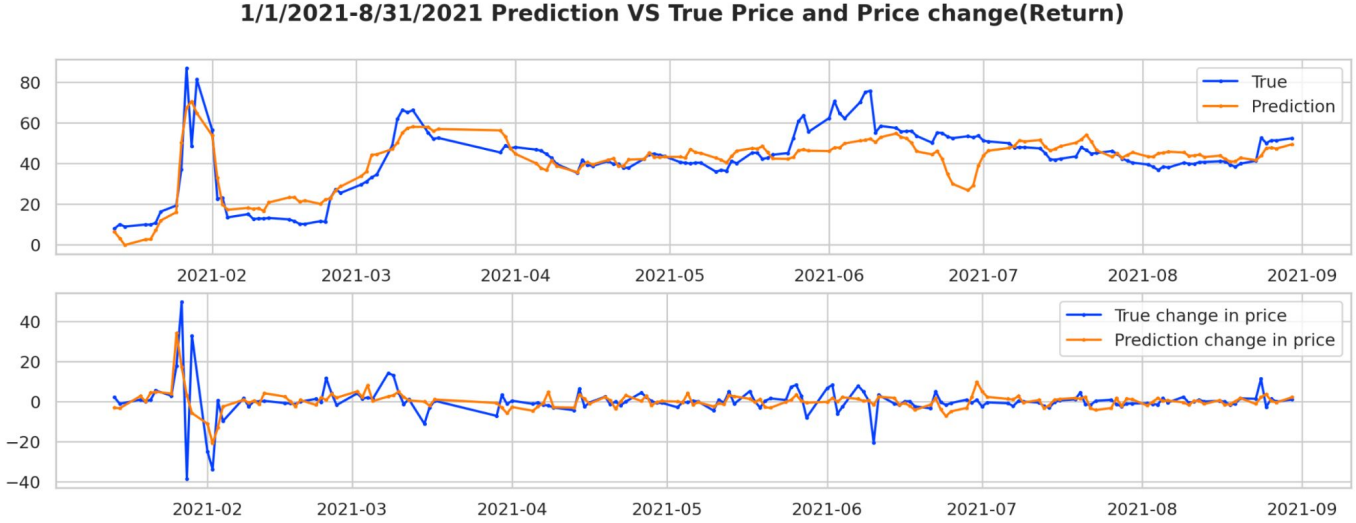




Using our fine-tuned Bert, we can get the **embeddings of each titles**(pooler_output of [CLS] token), which is a summarization of information in each title of shape (1, 768).

Each day, we take the **average** of the 300 titles' embeddings, and get a single **768** dimension vector, and **we concatenate this vector with our time series each day**, and get a 5+768=773 dimension time series.

Time Series + Bert embeddings



	MSE	RMSE	MAE	MAPE	Serial Corr	Pearson Corr(IC),	Spearman Corr(IC)
0	80.43058	8.96831	6.416645	0.156642	0.212488	0.20713	0.233539

We can see the model does a much better job at forecasting, and the Correlation Coefficient of our predicted change in price with the true price change reach 20%, which is very high in quantitative trading. Normally a multi-factor model of 1000+ technical indicators can get 18%+ correlation coefficient and make a million dollars. Our NLP model is performing very outstanding for this stock.

Summarization

	MSE	RMSE	MAE	MAPE	Serial Corr	Pearson Corr(IC),	Spearman Corr(IC)
Baseline_sentiment	73.073166	8.548284	6.207523	0.143593	0.873002	-0.177247	-0.097482
TF-IDF	239.582047	15.478438	14.070969	0.417722	0.633061	0.106105	0.082686
Bert_embeddings	80.430580	8.968310	6.416645	0.156642	0.212488	0.207130	0.233539

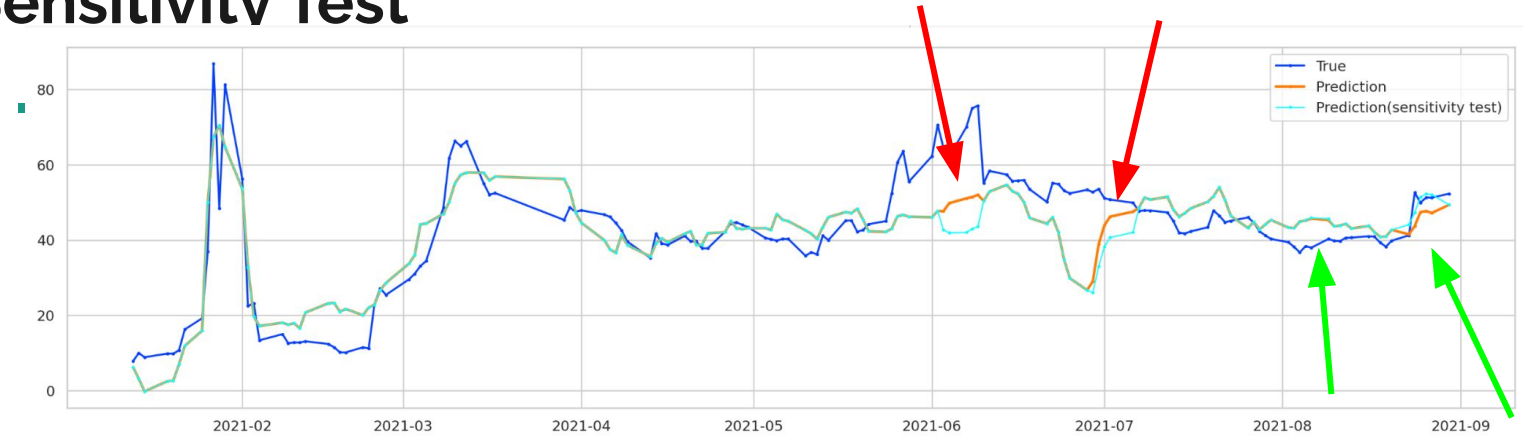
As a summarization of model performance,

Our **baseline model with sentiment labels** although has the lowest RMSE, perform the worst job at predicting stock returns. Its prediction negatively correlates with the true change in price.

Our **TF-IDF model**, perform a decent job predicting stock return, but the RMSE is really large, the stock price predicted is very off by scale.

Our **Bert embedding model,** perform a outstanding job predicting stock return, and have a chance of creating solid profit for stocks that are affected by sentiment and news.

Sensitivity Test



I put bert embeddings of Feb-03(the day when the stock fall drastically) to **Jun-02 and Jun 28**. And we can see from the blue curve that the model can now predict a **lower price** than before(orange)

I changed **Aug-02 and Aug-20**'s bert embeddings with Jun-07(the day when the stock stand on new peak) bert embeddings. And we can see from the blue curve that the model only slightly overpredict on Aug-02, and **overpredict** on Aug-20

It says that our model is not uniformly sensitive, and here are some **strategies** to improve it:

Fine tune our bert model using **Reddit** post dataset to better capture the language used there.

Train with **all S&P 500** stocks with corresponding news to better learn the relationship between reddit post and stock price

We can give **most voted titles** a higher weight in the dataset, because of its influence.

Summarization

- **Summarize:** Our best performed model is **LSTM+bert embeddings**, with a outstanding correlation coefficient of change in price, but our sensitivity is not very uniform and stable.
- **Discuss:** The study of GameStop short squeeze gives a very **promising** direction of including **sentiment factors** into traditional multi-factor forecasting model in quantitative trading to enhance the predictive power of traditional models. However, it's crucial to navigate the data collection process carefully to ensure compliance with copyright and privacy laws.
- **Propose:** The future research can involve how to represent text better align with time series representations, probably using **Large Language Model** and **multimodal** techniques, **contrastive learning** to better differentiate between relevant and irrelevant information within textual data and noise in stock price. Such advancements could lead to more sophisticated models that better capture the nuanced relationship between market sentiment and stock price movements, offering a richer understanding of market dynamics and improving the predictive capabilities of quantitative trading models.