Can news headlines predict changes in stock prices

Pedro Belotti, Chun Him Cheung, Evan Fjeld, Dmitri Zadvornov
August 1, 2022

Berkeley
UNIVERSITY OF CALIFORNIA

Introduction

“Buy the rumour, sell the fact”

News has always been a key factor of investment decisions.

NLP models in finance can provide the engine to :

- Intelligent document search (reports and filings, fraud detection)
- Automate capture of earnings call, leaders presentations and central bank
- AI chatbot for investor and clients
- Build data to improve risk assessment and credit scores
- Automate internal marketing processes (customer profiling, product match)

This project will focus on how we can use NLP to create effective trading strategies.

Question

Does the sentiment of financial news stories provide useful information to help predict the stock market performance?

This question is critical to create automated trading strategies – hopefully profitable.

Our review of papers shows that although state of the art models accurately represent sentiment from news sources, using them to predict stock prices has mixed results.

The goal of this model is to quantify a momentum trading strategy. When news cycle is concentrated around a particular company, we want to know if trading on this momentum is profitable and can momentum on a single stock potentially outperform its sector.

Berkeley
UNIVERSITY OF CALIFORNIA

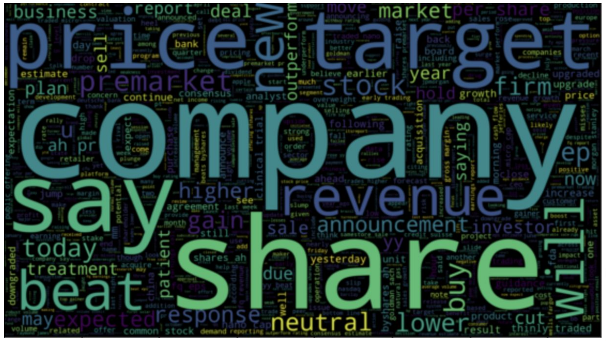
Data

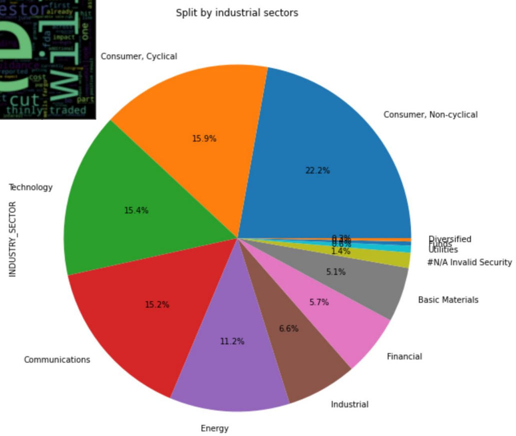
Labeled Data

- Harvested from Kaggle
- Manually labeled without strong explanation

Un-Labeled Data

- 110k headlines from May 2011 – Dec 2019
- Bloomberg

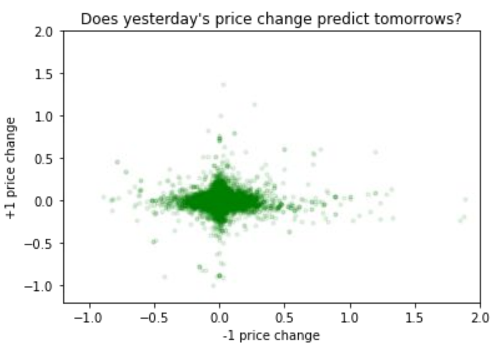




Data

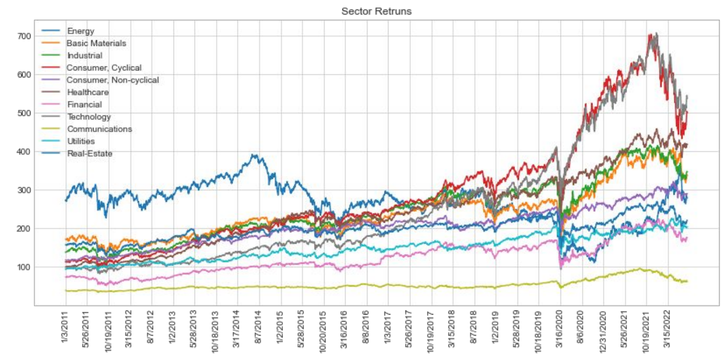
Stock Prices

- Pulled from the Yahoo! Finance API
- 4,651 tickers successfully found for 830,9231 rows of price data
- 2295 tickers were not able to be found



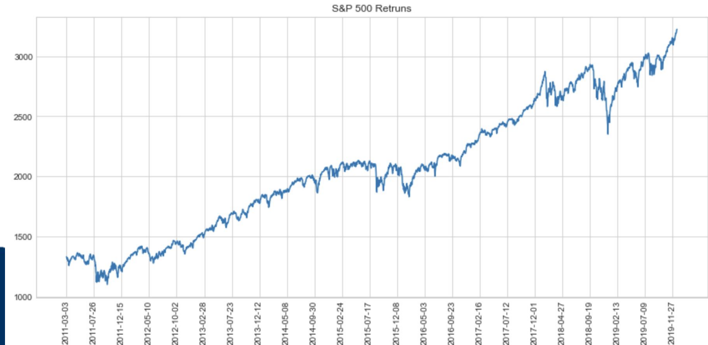
GICS sector indices

- Daily movements of GICS sector indices pulled from Bloomberg



S&P 500

- Daily prices of the S&P pulled from Yahoo! Finance to compare to our trading strategy



Berkeley
UNIVERSITY OF CALIFORNIA

Analysis Outline

Part 1: ML Model Application

Headlines Set 1

Pretrained models:
Texblob
Vader (default lexicon)
Vader (financial lexicon)

Headlines Set 2

LSTM model fine-tuned with labeled data from set 1, applied to headlines in set 2

BERT model fine-tuned with labeled data from set 1, applied to headlines in set 2

finBERT pre-trained model applied to headlines in set 2

Part 2: Financial Trading Strategy Test

Headlines + Sentiment Indicators

Stock Returns

Pre-defined trading strategy applied to each model sentiment indicator
Ex. Buy on good news, Sell on bad news

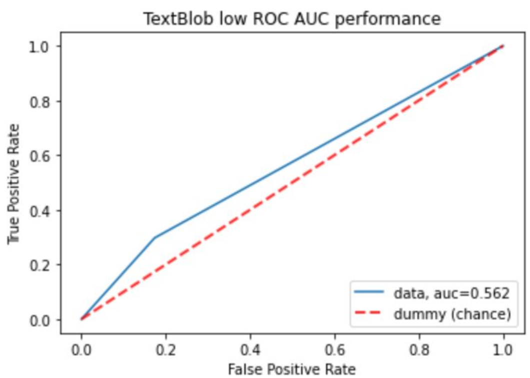
Benchmark: S&P Returns from May 2011 until Dec 2019

Success = Strategy returns > S&P returns

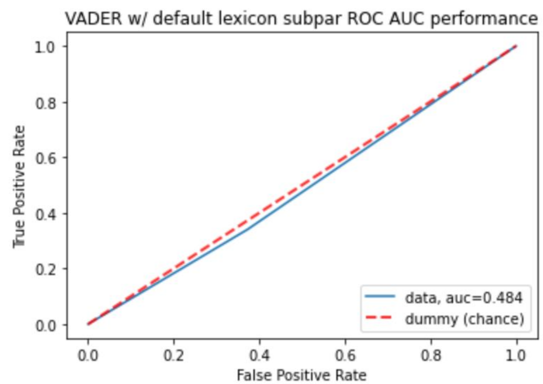
Approach - simple sentiment models

- First, utilize simpler sentiment models for baseline along with flip of a coin to beat
  - Lexicon-based TextBlob and Vader models

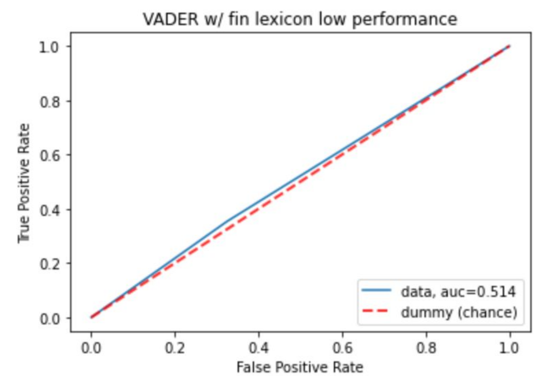
ROC AUC: 0.562



ROC AUC: 0.484



ROC AUC: 0.514



TextBlob metrics

Balanced accuracy: 0.562
F1 score: 0.418

VADER w/ default lexicon metrics

Balanced accuracy: 0.484
F1 score: 0.422

VADER w/ financial lexicon metrics

Balanced Accuracy: 0.514
F1 Score: 0.445

Approach - more advanced models

- **BERT - Bidirectional Encoder Representations from Transformers**
  - BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
  - Wiki (800M words) and BookCorpus (2,5B words)

- **finBERT**
  - based on 2019 FinBERT: Financial Sentiment Analysis with Pre-trained Language Models paper by Dogu Araci
  - BERT further trained on 50K financial records:
    - Reuters TRC2-financial news articles 2008-2010 - **46,143** examples
    - **4845** random samples from LexisNexis Financial PhraseBank5 from Malo et al. 2014

Approach - more advanced models (cont'd)

- Train LSTM, BERT models, use pre-trained finBERT model

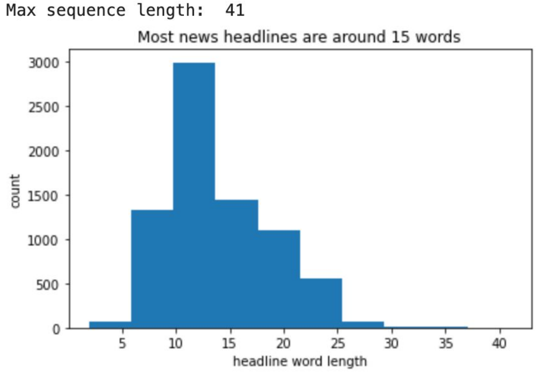
LSTM

	num_epochs	embed_dim	bidir	lstm_stack	recurrent_dropout	dropout	learning_rate	cv_balanced_accuracy	cv_f1_score	cv_roc_auc_score	cv_accuracy	cv_loss
0	5	64	True	[128]	0	0.5	0.001	0.975406	0.978860	0.975406	0.975581	0.080744
1	5	64	False	[128]	0	0.5	0.001	0.845354	0.752974	0.845354	0.830108	0.302219
2	5	64	True	[64]	0	0.5	0.001	0.974751	0.978281	0.974751	0.974921	0.082378
	5	64	False	[64]	0	0.5	0.001	0.675931	0.379854	0.675931	0.629251	0.496023
	5	32	True	[128]	0	0.5	0.001	0.974779	0.977625	0.974779	0.974262	0.088401
5	5	32	False	[128]	0	0.5	0.001	0.917966	0.929462	0.917966	0.918821	0.243887
6	5	32	True	[64]	0	0.5	0.001	0.974023	0.977196	0.974023	0.973733	0.088573
7	5	32	False	[64]	0	0.5	0.001	0.661055	0.371596	0.661055	0.617766	0.516734

BERT

num_epochs	seq_length	learning_rate	cv_balanced_accuracy	cv_f1_score	cv_roc_auc_score	cv_accuracy	cv_loss
2	128	0.00010	0.980021	0.981060	0.980021	0.978353	0.095136
2	128	0.00003	0.980379	0.981298	0.980379	0.978616	0.089413
2	64	0.00010	0.980976	0.982012	0.980976	0.979409	0.093863
2	64	0.00003	0.983214	0.983980	0.983214	0.981653	0.075998

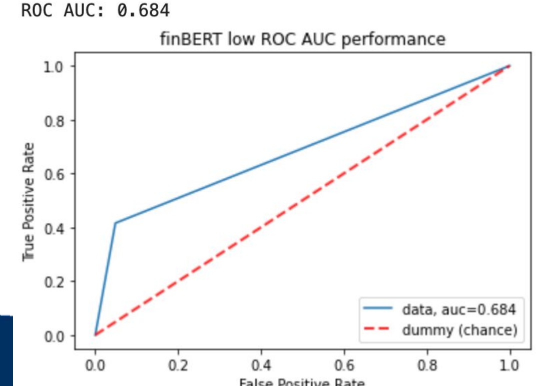
```
Max sequence length:  41
```



finBERT metrics

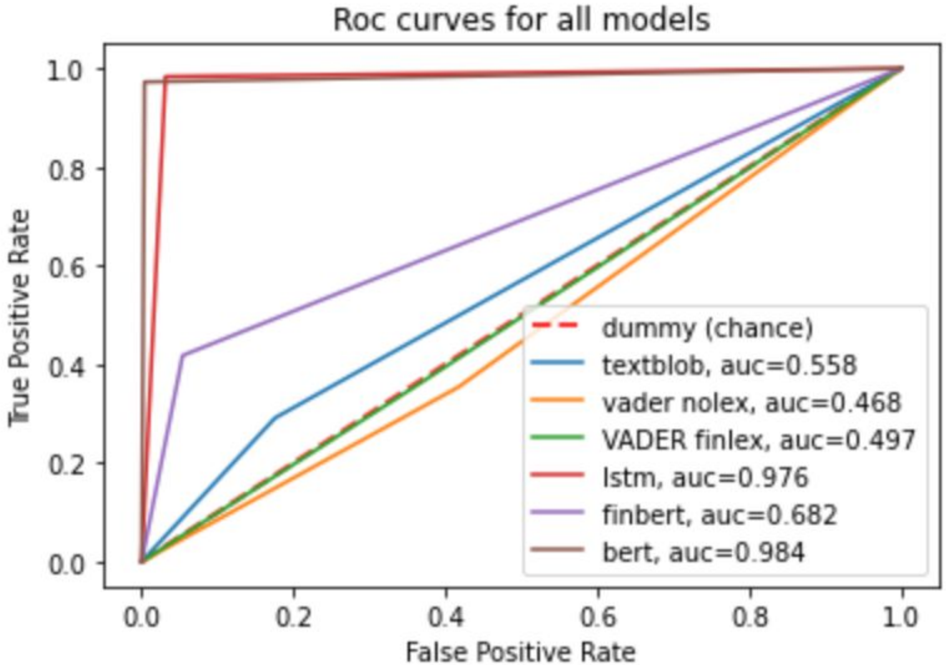
Balanced accuracy: 0.684
F1 score: 0.573

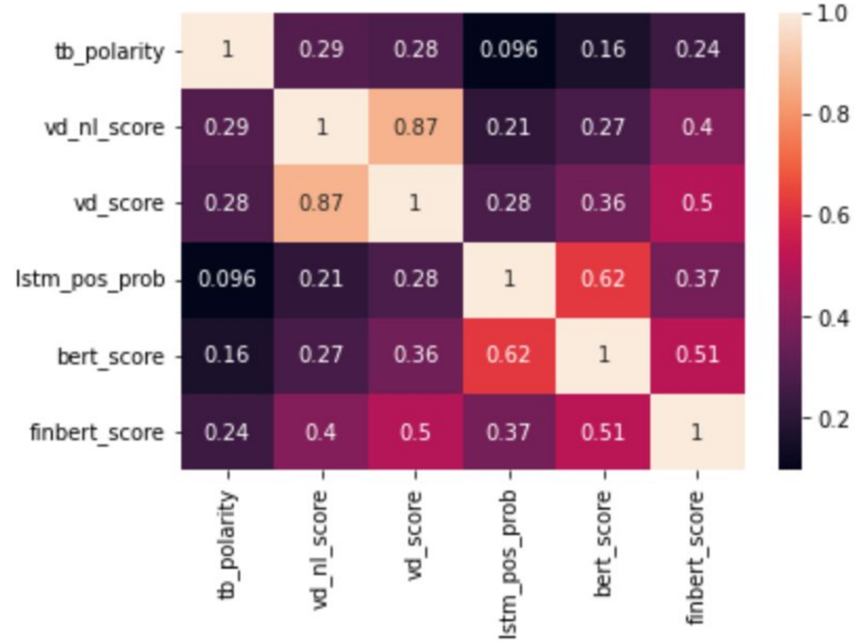
```
ROC AUC: 0.684
```



Model Comparisons

Test set performance of models





Fine-tuned BERT beat all the models, with custom trained LSTM coming in second

Berkeley
UNIVERSITY OF CALIFORNIA

Trading Strategy (1)

Results for 4 models

1. Model predicts whether headine was positive (+1), neutral (0) or negative (-1).

2. Look at the average sentiment by sector each day. If the average sentiment is above the often threshold and the number of article threshold, then buy the sector index at t+1 and calculate the return at t+2.
   a. E.g. if sentiment on FB, GOOG , MFST and TWTR were 1,1,1 and 0, average sentiment is 0.75.
   b. If sentiment threshold is 0.3 and count threshold 3, then both conditions are satisfied and trade occurs. We calculate the return as:
      i. $r = (p(t+2)/p(t+1) - 1) \times$ leverage factor – margin cost

3. Across our 6 models, we used the sentiment threshold=0 and count threshold=3.

Berkeley
UNIVERSITY OF CALIFORNIA

Trading Strategy (2)

```
Input the sentiment threshold (from 0 to 1):0
Input the count threshold (suggest from 0-5):3
Input a margin cost (IB has margin cost at 0.0383):.0383
Total number of trades are: 8837
```


```
Total return on this strategy was: 3.158197755986958
Total return on the SP 500 was: 2.3684709886384447
Total return of unlevered strategy was: 1.7244456071098757
The alpha of the levered strategy against the SP 500 was: 0.7897267673485131
```

Berkeley
UNIVERSITY OF CALIFORNIA

Trading Strategy (3)

model	Sentiment Threshold	Count Threshold	Margin	Strategy Return	Unlevered Strategy Return	SP 500 Return	Strategy Sharpe	Unlevered Sharpe	SP 500 Sharpe
bert_score	0	3	0.0383	3.896395	1.844559	2.368471	0.025446	0.034358	0.044927
tb_score	0	3	0.0383	3.370373	1.779201	2.368471	0.021945	0.031215	0.044935
vd_nl_score	0	3	0.0383	4.174579	1.908265	2.368471	0.025885	0.035108	0.045019
vd_score	0	3	0.0383	3.390759	1.781128	2.368471	0.022249	0.031549	0.045126
lstm_score	0	3	0.0383	3.649645	1.806336	2.368471	0.024186	0.033134	0.044968
finbert_score	0	3	0.0383	4.228289	1.905846	2.368471	0.026825	0.035989	0.045269

The more complicated the model, the worse it actually performs on new headlines!

- Goes back to the problem that we lack a proper labelings from our underlying data set
- None of the models can outperform the S&P 500 on a risk-adjusted basis (i.e. the Sharpe ratio of all our strategies is lower than the S&P 500

Berkeley
UNIVERSITY OF CALIFORNIA

Conclusion (1/2)

- ★ From a machine learning perspective the 2 best performing models were LSTM and BERT with AUC of 0.976 and 0.986 respectively.
  - FinBERT, which is trained on over 50k financial labels, surprisingly performed badly against our dataset.
  - This could be due to bad quality of our labelled dataset, imbalance or perhaps the specifics nuances of financial headlines.
- ★ From a financial perspective, as expected making money off the stock market is not easy.
  - Our trading strategy test showed Vader with a financial lexicon generated best returns and sharpe ratio, but none of model's trading strategies performed better than the S&P.

Berkeley
UNIVERSITY OF CALIFORNIA

Conclusion (2/2)

- From a financial perspective, as expected making money off the stock market is not easy.
  - Our trading strategy test showed Vader with a financial lexicon generated best returns and sharpe ratio, but none of model's trading strategies performed better than the S&P.
- Potential rationale:
  - Data again could be miss labeled as it was crowdsourced
  - Strategy is too simplistic, doesn't consider how much of news were already anticipated in stock prices ahead of time
  - Other variables impact stock prices and news absorption than purely the sentiment information.

Final Remarks

Better data could improve results:

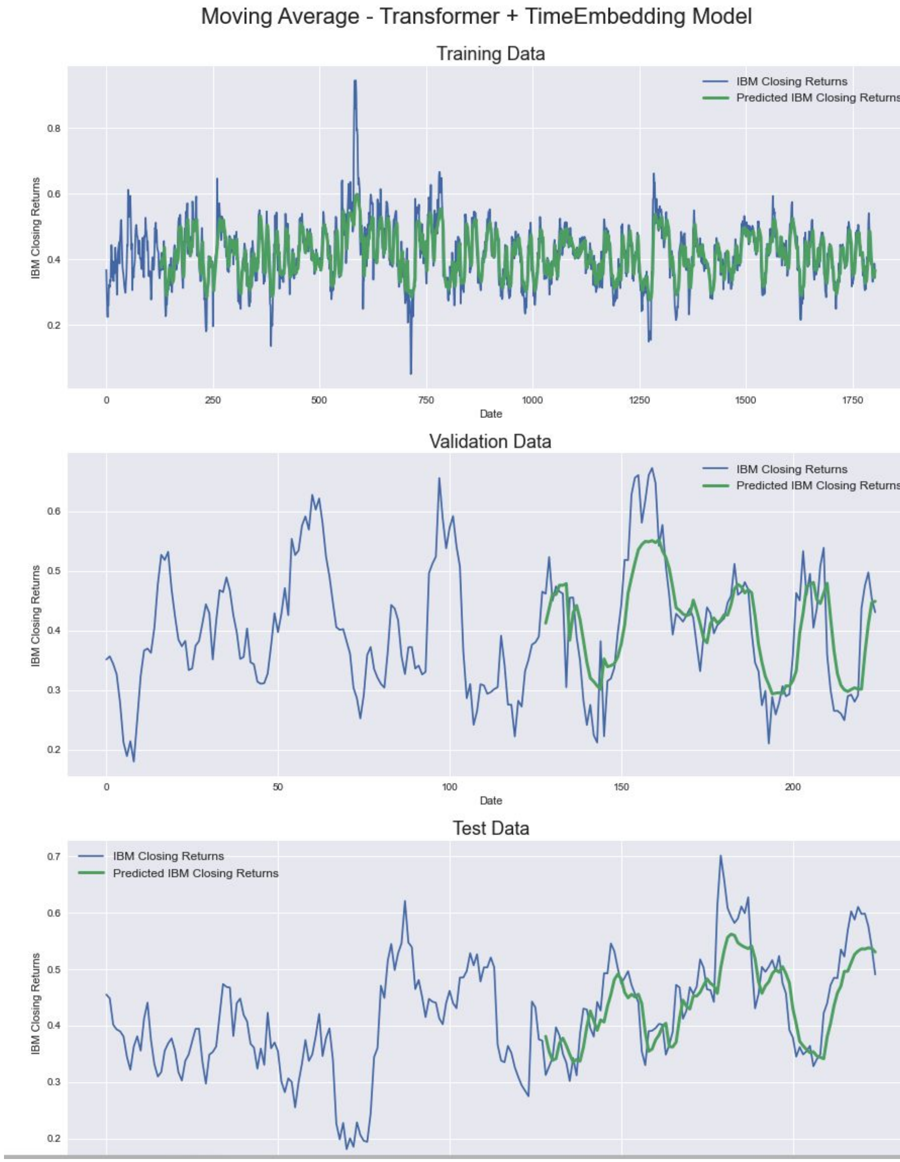
Consider using the returns themselves for a particular stock as label data to produce sentiment. This would get rid of potential labelling bias.

Modelling stock returns could improve our trading strategy:

Autoregressive models with news sentiment as feature might yield better input for our trading strategy. (we tried LSTM and Transformer applied to stock returns).

More computational power / engineering could help improve our model training.

Some models took a (VERY) long time to run on our dataset and may require more epochs to train right.



Berkeley
UNIVERSITY OF CALIFORNIA

Thank You

Contributions

Pedro Belotti: Trained fin-BERT on our labeled data providing very accurate sentiment predictions

Chun Him Cheung: primarily responsible for the trading strategy once the sentiment models were completed

Evan Fjeld: worked to pull the price data, some data exploration, and explored LSTM models to predict price

Dmitri Zadvornov: worked on the sentiment models running TextBlob, VADER, LSTM and BERT showing massive improvements on sentiment prediction using LSTM and BERT