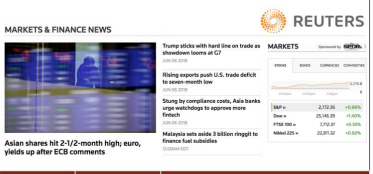
Natural Language Processing and Event-driven Stock Prediction

Project by Cheryl Ji and Jimmy Qin

CS 230 Deep Learning | Taught by Professor Andrew Ng, advised by Lucio Mwinmaarong Dery Jr.

Data Wrangling and Embedding

Original data: Reuters' news from October 2006 to November 2013 scraped by Ding et al. (2014). The raw dataset is .txt files each containing one news event with headline, main body and timestamp.



	Procedure	Affected data
Word embedding	Tokenization	• Tokenize headline into a word list • Unify each word in terms of tense, noun and case using the `en` package, and remove stopwords using `nltk`.
	Transfer learning	• Do transfer learning with the help of the `Word2Vec` model. • Train the model once.
Data wrangling	Stock market data acquisition	Download S&P500 data from Yahoo Finance.
	Data alignment	Align each event with the next trading date and the corresponding index movement direction.
	Response engineer	To capture extreme movement, we set response as 1 if the absolute change rate > 0.7%. This makes the data more balanced(see below).
	Missing values	Drop 25 empty news headlines and nan trading data.

End result: We are left with **92,063** news events that we can feed into our Bag-of-words and LSTM model. Among all the news events:
· The number of extreme changes (y=1): 42414
· The number of non-extreme changes (y=0): 49649

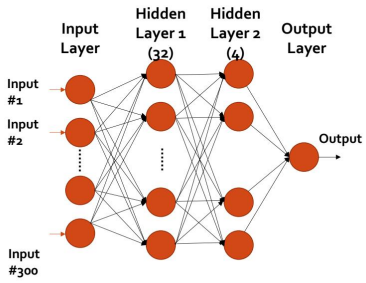
Training, Validation, and Test Sets

Training set: The first [**75869**] news titles (the first 80% of which will be used for training and the last 20% for dev and testing) to address the time-varying characteristics of data; This includes events from 2006-10-20 to 2012-10-06.
Data skewness check:
· The number of extreme changes: changes: 37509
· The number of non-extreme changes: 38360

Dev and test set: The rest of the news events are in the dev and test set. Considering the fact that some of the triggering words are influential for some specific periods of time while useless for the rest, we shuffle the last 20% of the data and allocate them equally to dev and test set.
· Number of events in the dev set: 8097
· Number of events in the test set: 8097

Bag-of-words and Neural Network

Bag-of-words
Our baseline model is "bag-of-words" using a standard 3-layer neural network, with 300-d sum/mean word as inputs.



Linear → ReLU → Linear → ReLU → Linear → Sigmoid

· **Batch-normalization**
· **Drop-out regularization**
· **Grid search**: learning rate, drop-out rate and batch size

Sum and Mean Model

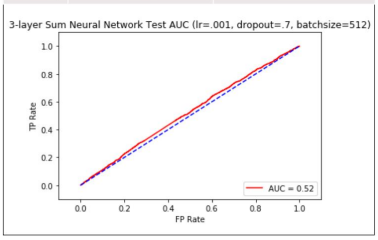
For each news title, we sum up (1) / average (2) the 300-dimensional vector of each word in that title to form a single 300-dimensional vector (see the formular below). We use this vector as input.

$$input_i = \sum_{j=1}^{|title_i|} word\ vector_{i,j} \quad (1)$$

$$input_i = \frac{1}{|title_i|} \sum_{j=1}^{|title_i|} word\ vector_{i,j} \quad (2)$$

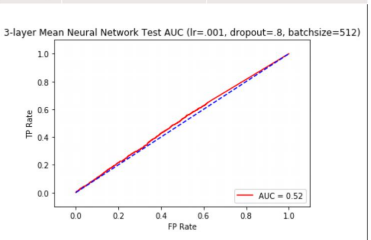
Sum Model result

	Train	Test
AUC	0.5927	0.5224
Accuracy	0.5641	0.5974



Mean Model result

	Train	Test
AUC	0.5738	0.5165
Accuracy	0.5550	0.6535

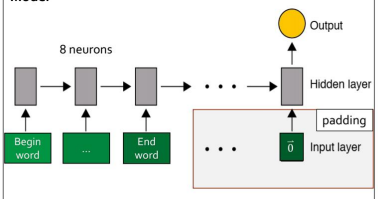


LSTM

Padding

Generally the number of layers in the sequence mode is equal to the number of words in a given news title. However, we require the structure of the inputs to be exactly the same with each other. In order to implement this model, we do padding to make sure that the number of layers is exactly 18. (The maximum length of the news title consists of 18 words.) Specifically, we add the layers of all zeros to the 300-dimensional vectors until it achieves 18 layers in total.

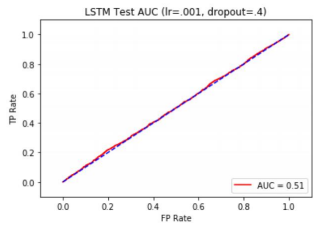
Model



Our sequence model is many-to-one, meaning that for each news title, there is a sequence of inputs and a single output predicting the up or down of the S&P 500 index. The structure as well as the mathematical detail of each hidden layer is demonstrated below.

Result

	Train	Test
AUC	0.6372	0.51356
Accuracy	0.5980	0.5934



Contextual Decomposition

To this end, we have focused on prediction accuracy. Another problem of interest is to understand the influence of each individual word, by comparing changes in predicted probability before and after masking the word from the headline using the Bag-of-words Sum Model.

Examples of News with Predictive Power from BOG

Morgan Stanley's deputy head of investment banking exits

U.S. government may not hit debt limit until October

Very predictive: |change|>10%	Predictive: 5%<|change|<10%

Though our predictive words search is not exhaustive, from the 2 examples above we can see that actions in financial news headlines, e.g. 'exit', etc., are likely to have strong influence on the volatility of S&P 500 index. So are some important named entities, e.g. 'government', 'head', etc.

However, the joint influence, e.g. 'head' and 'exit' is not significant in our test.

Conclusion

- Despite well-known randomness of stock exchange, our study tries to capture (volatile) stock movement via financial news using natural language processing. Bag-of-words and LSTM perform equally well under "Accuracy" metric (**0.65** for mean bag-of-words). Considering the nature of capital market, this result is more than satisfactory. However, LSTM model suffers more from overfitting post tuning .
- We also preliminarily explore the predictive power of individual words.
- To improve our study, we would expand our dataset by scraping data and focus on training the pretrained word2vec . We expect to get better results.