Predicting Major Market Events from Media Headlines and Multimedia Items

Michael Di Amore, Yaqing Wang, Yitong Xu, Varun Aggarwal
Columbia University
In association with Capital One

December 4, 2017

I. Introduction and Goals

Since the last report we were able to expand our models, work on getting feature importance for our Deep Neural Networks, and create a working business case to present to our stakeholders. As a refresher, we implemented two models one a classification model and the other a Seq2Seq time-series model. We were also previously using only the GDELT data from US business sources. Since our last report we've improved our models and come to firm conclusions about the importance of GDELT in predicting rare market events on a daily timescale.

II. Model Specification and Expanded Feature Space

We expanded the GDELT data feature space to include all business news not just US news from 2000-01-01 to 2017-10-06. We feel this gives us a more thorough representation of the GDELT events database.

We continued to expand upon our classification model referenced in the first report. This time however, we added the ability to group results by a sliding window. As our previous model was predicting events before and after they happened, we group the results with a stride of window size of 5 days. Meaning a predicted event one trading week before and one trading week after an actual event has occurred will count as one prediction for model evaluation purposes.

For our Seq2Seq model, we simplified the architecture and added a highway network before the last layer of the network. By simplifying the architecture we reduce the number of parameters in the network and the highway network adds a boost in performance. See the Seq2Seq section for more details.
Author: *Michael*

III. Classification Model Performance

We present the following results across different definitions of market events to show that our model is relatively robust to changes in market definition. Below we present the results on the validation set. This set runs from 2011-07-22 to 2014-04-28, we choose to highlight this set as it is closer to the training set (which runs from 2000-07-05 to 2011-07-21) in terms of chronology therefore it is more likely to hold similar market dynamics to the train set. It is also worth noting that this set runs approximately a full three years, which is quite some time for market dynamics to change. Our performance over this time period suggest that our model does well in these states of the market. Figure 1 titled "Metrics", shows pointwise

metrics for different definitions of a market event. These results are promising, as they reveal our models ability to learn at least some of the complicated dynamics of the market.

We also present Figure 2, a plot showing the predictions versus the actual, and their intersections. The red vertical lines in the first plot represent actual events, the blue vertical lines in the second plot represent predicted events, and the green vertical lines represent intersection between these two sets. As you can see from Figure 2, the point estimates do not tell the whole story. As our model captures the major regions of red, while also finding new patterns like the space around 2012-06.
Author: *Michael*

# of Stdev from Rolling Mean	2.0	2.5	3.0	3.5
Accuracy	89%	91%	92%	94%
F1 Score	0.42	0.37	0.26	0.13
RoC Score	0.83	0.90	0.88	0.97
Precision	0.29	0.24	0.16	0.07
Recall	0.78	0.90	0.83	1.00
Area under PR Curve	0.54	0.57	0.46	0.53

Figure 1: *Metrics of Classification Model*

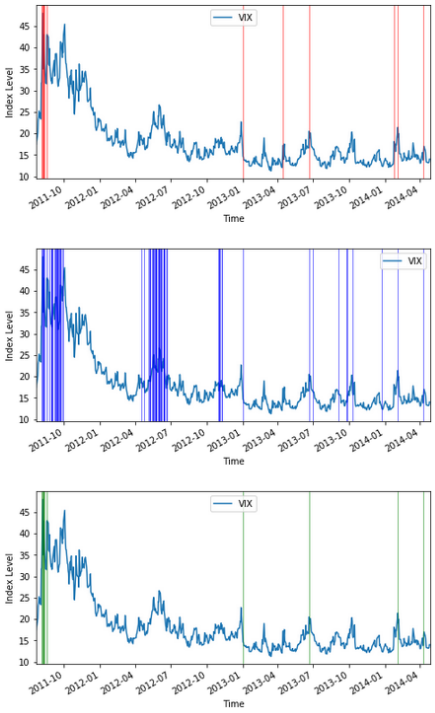

Figure 2: *Predicted Plots of Classification Model-3.0 Stdev*

IV. Seq2seq Prediction Model

Last time, we proposed a dual-staged attentive Long-Short-Term-Memory (LSTM) model (Qin, 2017) [**DualRNN**] for our security return forecasting, and further improved our model predictive performance through adversarial training (Goodfellow, 2014) [Ian14]. We proved our model is capable of modeling both local and global variants through attention mechanism given our input features are important. The encoder structures are shown in Figure 3. At each time step, the model calculated attention weights for each input feature, and re-calculated each input feature by these calculated attention weights, then serve the new inputs to the LSTM layer. Intuitively, it is similar to an image caption process. Imagine in video image captioning, we need to first identify the critical position of the picture so that we can further describe it. Here, the model

$$e_t^k = \mathbf{v}_e^\top \tanh(\mathbf{W}_e[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_e \mathbf{x}^k)$$

and

$$\alpha_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^n \exp(e_t^i)},$$

Figure 3: *Dual Attentive Encoder*

pays more attention on the features that receive higher weights, and later in training process, the signal can be efficiently backpropagated to the front layers though these differential exponential functions. Another benefit of applying this attention architecture is that we could analyze the feature importance regarding to the prediction by examining the attention weights of the encoder.

However, to predict major market events, our data are noisy and highly unbalanced. From 2000 to 2017, we ended up 3600 training data and 600 testing data. The amount of data are not sufficient to train a complicated deep learning model for prediction, e.g. the

dual-staged attentive model. We implemented original model on our dataset, the forecasting accuracy showed our model suffered from the underfitting problem. Therefore, we simplified the original model architecture by replacing the complicated decoding LSTM layers with a weighted sum of all hidden outputs from encoder, we adopted (Yang, 2016)[Yan+16] attention architecture as shown in Figure 4, and kept the encoder as it was before. Empirically, we found that adding highway network [SGS15] (2015) before the final prediction layers helped improve the model accuracy. We also added dropout for regularization.

$$u_i = \tanh(W_s h_i + b_s),$$
$$\alpha_i = \frac{\exp(u_i^\top u_s)}{\sum_i \exp(u_i^\top u_s)},$$
$$v = \sum_i \alpha_i h_i,$$

Figure 4: *(Yang, 2016) Attention mechanism*

Since DeepLIFT and Integrated gradient methods had shown their effectiveness on computing feature importance of deep learning model. We integrated the important features selected by both feature attribution methods. Specifically, 9 features ended up in our Seq2Seq model, e.g, *"vix"*, *"slsi"*, *"unemploy"*, *"wti_co"*, *"dff"*, *"m1v"*, *"m2v"*, *"AdjClose"*, *"avgtone"*. Figure 5 shows our predictive performances. The green curve is the ground truth, while the blue curve is our model results. By leveraging the importance features selected by Integrated Gradients and DeepLIFT, our model shows great capacity to capture the future trend. We believe in the future this *Seq2Seq* model could facilitate other similar research.

To further analyze the feature importance in terms of predictive performance of our *Seq2Seq* model, we apply a similar procedure as in (Ghosh, Muresan, 2017) [Deb17]. As shown in Figure 6, we visualize the attention weights of all input features, the larger the weights, the more important of the feature. Specifically, we found out that Adj Close is the most informative feature across all time steps.
Author: *Yitong*

Figure 5: *Predictive Performance on Test*

V. Integrated Gradients

Integrated Gradients (IG) is a method for extracting feature importance for Deep Neural Networks. Originally applied to convolutional networks on images, we apply it to convolutional network on 1D grids which we can think of as a time series. The authors of the Integrated Gradients paper [STY17] like to think of IG as an "Axiomatic Approach" , with two main axioms "Sensitivity" and "Implementation Invariance". The paper defines sensitivity as "if for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution." While this explanation is somewhat wordy, we can start to make sense of it. If we think of a simple input as having only two features where one is constant and the other is variable and give different predictions then the

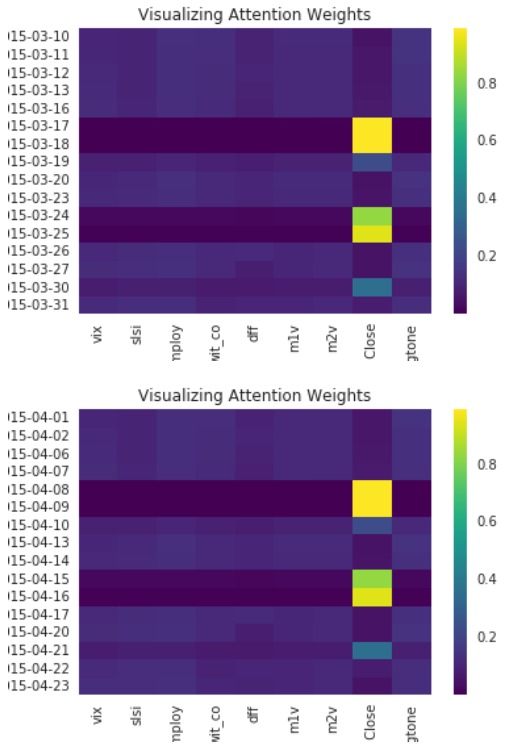

Figure 6: *Attention Visualization on Test*

feature that is variable should have a non-zero attribution. For "Implementation Invariance" the idea is that two networks should have the same attribution if they are functionally equivalent. Meaning if two networks are equal in output for all inputs, regardless of implementation, they should have the same attributions. We present some results from integrated gradients below

Figure 7 presents the feature importance for Integrated Gradients for a setting of 3.5 Standard Deviations, we've chosen this arbitrarily and though there is some variation due to initialization, the results are somewhat robust for different settings of sigma.
Author: *Michael*

VI. DeepLIFT

DeepLIFT (Deep Learning Important FeaTures) is another way for computing importance

```
Volume           0.007826
numsources       0.024539
numarticles      0.025227
avgtone          0.038073
NumMentions      0.044597
Adj Close        0.053449
GoldsteinScale   0.067063
Open             0.068683
Low              0.077508
dff              0.082976
wti_co           0.090159
High             0.098485
Close            0.108424
unemploy         0.111372
m1v              0.154115
m2v              0.185978
slsi             0.187246
vix              0.501248
dtype: float64
```

Figure 7: *Importance scores computed by Integrated Gradients*

scores to the inputs for a given output in deep neural networks. It is a rather novel and effective approach in that it could deal with both zero and discontinuous gradient problems as well as reveal dependencies while other methods would fail or miss to do so.

DeepLIFT [SGK17] adopts backpropagation algorithm to decompose the output prediction on specific inputs through the contributions of all neurons in the neural network. In order to achieve this, a 'reference' would be chosen, and the contribution score of each neuron would be assigned based on the difference between the activation and its 'reference' activation. Then through backpropagation, the difference between the output and the 'reference' output would be eventually explained by the difference between the input and the 'reference' input.

After running our model through DeepLIFT, we obtained importance scores with both positives and negatives. So we took the absolute value before taking the mean, and the sorted scores are shown in Figure 8. Based on the

results, it is obvious that real time prices and quantities of S&P 500 (i.e. Low, Close, Volume, High, Adj Close, Open) do not play any significant role in predicting major market events. However, their volatility measure 'vix' seems to be the most crucial feature. Also, nearly all GDELT attributes have the lowest importance scores compared to other financial features, which is the same case using Integrated Gradients. Therefore, it would be reasonable to conclude that GDELT dataset does not provide much information as we hoped.
Author: *Yaqing*

```
Low               0.216537
Close             0.278578
NumMentions       0.376449
Volume            0.394436
numarticles       0.419895
numsources        0.444722
GoldsteinScale    0.457009
High              0.473969
avgtone           0.525098
Adj Close         0.557617
dff               0.605750
m2v               0.635376
Open              0.735713
m1v               0.803235
wit_co            0.842305
unemploy          0.860302
slsi              0.959443
vix               1.148718
dtype: float32
```

Figure 8: *Importance scores computed by DeepLIFT*

VII. Business Case

Even though our models over-predict the number of events, the predictions correspond to an upward trend (when prediction is a positive event) and a downward trend (when prediction is a negative event). Thus, a trading strategy based on our model has the potential to generate profits. And we are going to trade with the SPDR S&P 500 ETF (SPY), which tracks S&P 500 market index. Based on our model, which predicts an event one day in advance, we have come up with a simple trading strategy:

If the model predicts a positive event for the next day, we buy SPY at the opening price on that day. Our model generally over-predicts and will continue to predict a positive event for the next few days. We hold on to the stock during this period. The moment the model predicts a negative event or a no event, we sell at the opening price. A similar strategy is employed for a negative event, where we short the ETF instead of buying it. In the case where no event is predicted, we do nothing. On each day that an event is predicted, we buy/sell only $10,000 worth of it.

Since the events we are trying to predict are very rare, we only see 5 windows of trading based on our strategy over the 33 month period under test. We end up making a profit of $1,892.83. The average profit over each trading window was 3.78%. One thing to note is that SPY is a heavily traded ETF meaning we can scale our strategy to an initial investment in the hundreds of thousands and have little slippage in trading.

Due to the rareness of the major market events,we make another attempt to allow more frequent trades. We are trying to build a model to predict the ups and downs of S&P 500. The data frames are similar as before, except that we now give labels based on the rise or drop of the market. If the price of the index goes up, we label the day with 1's and if falls down, we label it with 0's.

For the prediction model, we are adopting the similar deep neural network used for classification section with slightly less layers. If the predicted sign shows rise in the index price, we will buy SPY with the open price the next morning and sell it with the close price at the end of the day. If the predicted sign indicates fall in the index price, we will short the ETF in the opening and buy back at the end of the day. For our validation set running from 2011-07-22 to 2014-04-28, we gained approximately

40% of the original investment using this strategy. This is comparable to a long-only strategy that returns 43%. Of course we don't factor in transaction costs so the long-only model will greatly outpace us in the real world. This goes to show that predicting daily moves requires a whole different perspective and model than predicting rare events.
Author: *Varun and Yaqing*

VIII. Future Work

For future work we can explore feeding the predictions of our classification model into another neural network as a feature. The most natural route is to use our classification model as a feature into a time-series model. Other extensions include expanding the feature space, perhaps incorporating some more momentum based features or better volatility estimates into the model. Since we are predicting rare events, having a good estimate of future volatility would help the model greatly. From our experiments we can already see that VIX is a strong indicator, but there may exist other volatility measures that are more powerful. We would also like to look into more granular time scales. If we could use second-by-second data or tick data perhaps news would show to be important. It is common knowledge that markets react in someway to news, what's difficult is finding how fast they do so. It may be that on a daily time frame the news is already acted upon and therefore we are trying to measure an effect that already becomes priced into the market.
Author: *Michael*

References

[Mac97] A. Craig MacKinlay. "Event Studies in Economics and Finance". In: *Journal of Economic Literature* 35.1 (1997), pp. 13–39. ISSN: 00220515. URL: http://www.jstor.org/stable/2729691.

[JWZ11] Kissan Joseph, M. Babajide Wintoki, and Zelin Zhang. "Forecasting abnormal stock returns and trading volume using investor sentiment: Evidence from online search". In: *International Journal of Forecasting* 27.4 (2011), pp. 1116–1127. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2010.11.001. URL: http://www.sciencedirect.com/science/article/pii/S0169207011000021.

[Smi12] Geoffrey Peter Smith. "Google Internet search activity and volatility prediction in the market for foreign currency". In: *Finance Research Letters* 9.2 (2012), pp. 103–110. ISSN: 1544-6123. DOI: https://doi.org/10.1016/j.frl.2012.03.003. URL: http://www.sciencedirect.com/science/article/pii/S1544612312000189.

[Ian14] Christian Szegedy Ian J. Goodfellow Jonathon Shlens. "Explaining and Harnessing Adversarial Examples". In: *arXiv Preprint arXiv:1412.6572* (2014).

[SGS15] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. "Highway Networks". In: *CoRR* abs/1505.00387 (2015). arXiv: 1505.00387. URL: http://arxiv.org/abs/1505.00387.

[XNS15] R. Xiong, E. P. Nichols, and Y. Shen. "Deep Learning Stock Volatility with Google Domestic Trends". In: *ArXiv e-prints* (Dec. 2015). arXiv: 1512.04916 [q-fin.CP].

[Yan+16] Zichao Yang et al. "Hierarchical Attention Networks for Document Classification." In: *HLT-NAACL*. 2016, pp. 1480–1489.

[Deb17] Smaranda Muresan Debanjan Ghosh Alexander Richard Fabbri. "The Role of Conversation Context for Sarcasm Detection in Online Interactions". In: 2017. URL: arXiv:1707.06226.

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *CoRR* abs/1704.02685 (2017). arXiv: 1704.02685. URL: http://arxiv.org/abs/1704.02685.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *CoRR* abs/1703.01365 (2017). arXiv: 1703.01365. URL: http://arxiv.org/abs/1703.01365.

[Smi12] [XNS15] [Yan+16] [Mac97] [JWZ11] [**DualRNN**] [Deb17] [Ian14] [SGK17] [STY17]