News Article Prediction Using Deep Learning

Research conducted in CS230: Deep Learning at Stanford University [450 Serra Mall, Stanford, CA]
Joshua Griffin [jrtg@stanford.edu] Kayiita Johnson [kayiita@stanford.edu] Levi Lian [levilian@stanford.edu]

Abstract

News media companies and news readers can benefit substantially from revenue prediction based on article content. We found this to be a novel area of research. We partnered with DeepNews.ai to source 45,000 articles from the Dallas Morning Star newspaper. Using a one-million-word dictionary, we converted texts to their Fasttext word context embeddings and compared the ability of a feed-forward deep learning architecture and a uni-directional LSTM architecture to resolve the relationship between articles and article revenue. We find that our LSTM network performs best with 34 percent accuracy when compared to our FF-DNN at 30 percent. We suggest better data screening and preprocessing, and network changes to improve algorithm performance.

Introduction

News media has undergone a major transformation in the last two decades. Previously, physical newspapers and magazines were delivered to the doorsteps of readers who paid a monthly subscription for a given service. Readers interested in national news were forced to choose between one or two major news companies. Additionally, local news was limited to distribution in its specific locale, making it challenging for readers to stay abreast of the current events in other areas. These factors led to a fractured market in which there were a large number of news outlets reporting the same or very similar stories for their respective markets.

Today, modern telecommunications services have led to the consolidation and elimination of much of that redundancy. More eyes are on fewer articles and the ability of readers to access content across news companies has broken the subscription model in many cases. In the process, these services have shifted marketing dollars from a dispersed collection of cheaper advertisements to a consolidated collection of more expensive ones. Additionally, online search, retail, and social networking platforms have risen and are capturing a disproportionate amount of ad revenue that was previously shared amount traditional news and media outlets. This has led to an increased reliance of news companies on generating reader traffic to their sites.

In this environment, news agencies, now more than ever, must publish stories that readers want to read; but investment in even a single story can run weeks or longer and the return is not clear. If news media companies are better able to provide the stories the public wants and needs, they will draw larger audiences and experience better financial performance while the public becomes better informed and engaged. To help bring news companies closer to realizing this outcome, we have partnered with DeepNews.ai.

Related Work

Predicting the revenue generated on a news article web page based on the text of the article is a novel problem in deep learning which involves understanding sentence context, article context, and revenue drivers. Although we were not able to draw directly from previous work on the problem, we were able to find research into article sentiment classification, popularity prediction, and stock price prediction based on article title and body text.

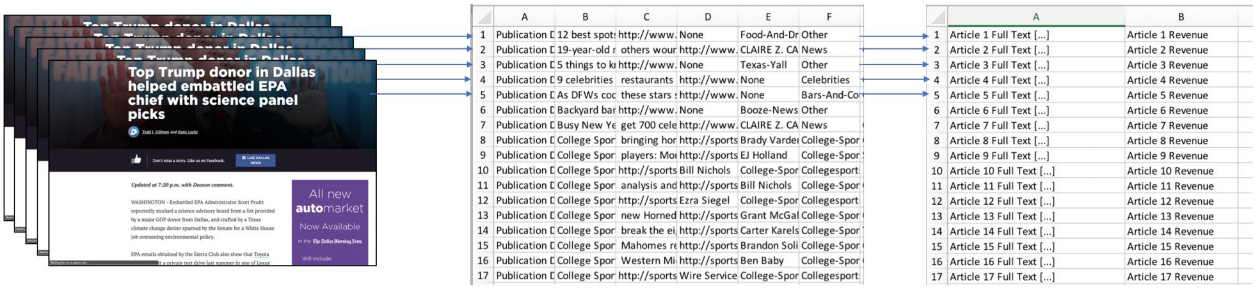
Sentiment classification predicts an abstract feature like level of positivity based on input string data like a movie review. Sentiment classification is an applicable area of study because it shares a many-to-one input structure with our target problem and can be used to gain abstract understandings from string data. Work by Xu et al. proposes an LSTM based architecture to capture sentiment information, especially useful for long strings of text. The method groups LSTM memory into long and short groups with long memory groups capturing overarching semantic features and short memory groups capturing specific semantic features.

Popularity prediction predicts the popularity of an article based upon article features, primarily text and title. Research by Stokowiec et al. addresses the problem by using a bidirectional LSTM with the late fusion approach introduced by Wang on the article titles. The bidirectional LSTM enables the algorithm to recognize context in both directions while, similar to Xu et al., the late fusion approach enables the algorithm to distinguish between inter and intra sentence relationships. Finally, adding word vector embeddings as outlined by Penningtong et al. improves the context recognition by 15 percent when compared against benchmark algorithms. These results are particularly interesting because the title of a given article can be easily changed.

Stock market prediction predicts stock market movements based on analysis of input string data like a set of news articles. Stock market prediction can build on sentiment classification to understand context and then draw connections between the two to generate price predictions of stocks. However, we noted that in the literature, sentiment classification is not a prerequisite for generating networks with relatively strong predictive power. We see this in research by Poulos who proposes a 12-layer GRU, taking character-level news headlines as features. After implementing masking to handle varying input lengths and dropout to each GRU hidden unit, he was able to achieve near state of the art accuracy. Taking this approach one step further, Pinheiro et al. compared the performance of a set of LSTM, CNN, and RNN networks. While multiple models featured layers of context and sentiment analysis, the character-level LSTM performed very competitively. Finally, Yoshihara was also able to realize strong results with an RNN operating on full news article text in a bag-of-words format. Yoshihara's work shows that it is not necessary to distill sentence context to generate strong predictions.

Dataset and Features

Our goal is to help bring clarity to news reporting investment decisions by using deep learning to reveal the link between content created and revenue generated. We have been granted access to 45,000 articles from The Dallas Morning News, in addition to header data on each article. This is not public data, but has been granted to Deepnews.ai and then given to our team. The article content was given to us in a series of text files, and the header data was provided in a large CSV file. Each text file corresponded to an entry in said spreadsheet, which included data such as article title, author, date published, section, impressions and various types of revenue. See the figure below for an example section of the CSV file and a screenshot of a Dallas Morning News article.



Top Trump donor in Dallas helped embattled EPA chief with science panel picks

Todd J. Gillman and Katie Leslie

Don't miss a story. Like us on Facebook.

Updated at 7:20 p.m. with Deason comment.

WASHINGTON – Embattled EPA Administrator Scott Pruitt reportedly stocked a science advisory board from a list provided by a major GOP donor from Dallas, and crafted by a Texas climate change denier spurned by the Senate for a White House job overseeing environmental policy.

EPA emails obtained by the Sierra Club also show that Toyota

t a private test drive last summer in one of Lexus'

	A	B	C	D	E	F
1	Publication D	12 best spot:	http://www.	None	Food-And-Dr	Other
2	Publication D	19-year-old r	others wour	http://www.	CLAIRE Z. CA	News
3	Publication D	5 things to k	http://www.	None	Texas-Yall	Other
4	Publication D	9 celebrities	restaurants	http://www.	None	Celebrities
5	Publication D	As DFWs coc	these stars :	http://www.	None	Bars-And-Co
6	Publication D	Backyard bar	http://www.	None	Booze-News	Other
7	Publication D	Busy New Ye	get 700 cele	http://www.	CLAIRE Z. CA	News
8	Publication D	College Spor	bringing hor	http://sports	Brady Varde	College-Spor
9	Publication D	College Spor	players: Mo	http://sports	EJ Holland	College-Spor
10	Publication D	College Spor	http://sports	Bill Nichols	College-Spor	Collegesport
11	Publication D	College Spor	analysis and	http://sports	Bill Nichols	College-Spor
12	Publication D	College Spor	http://sports	Ezra Siegel	College-Spor	Collegesport
13	Publication D	College Spor	new Horned	http://sports	Grant McGal	College-Spor
14	Publication D	College Spor	break the ei	http://sports	Carter Karels	College-Spor
15	Publication D	College Spor	Mahomes re	http://sports	Brandon Soli	College-Spor
16	Publication D	College Spor	Western Mi	http://sports	Ben Baby	College-Spor
17	Publication D	College Spor	http://sports	Wire Service	College-Spor	Collegesport

	A	B
1	Article 1 Full Text [...]	Article 1 Revenue
2	Article 2 Full Text [...]	Article 2 Revenue
3	Article 3 Full Text [...]	Article 3 Revenue
4	Article 4 Full Text [...]	Article 4 Revenue
5	Article 5 Full Text [...]	Article 5 Revenue
6	Article 6 Full Text [...]	Article 6 Revenue
7	Article 7 Full Text [...]	Article 7 Revenue
8	Article 8 Full Text [...]	Article 8 Revenue
9	Article 9 Full Text [...]	Article 9 Revenue
10	Article 10 Full Text [...]	Article 10 Revenue
11	Article 11 Full Text [...]	Article 11 Revenue
12	Article 12 Full Text [...]	Article 12 Revenue
13	Article 13 Full Text [...]	Article 13 Revenue
14	Article 14 Full Text [...]	Article 14 Revenue
15	Article 15 Full Text [...]	Article 15 Revenue
16	Article 16 Full Text [...]	Article 16 Revenue
17	Article 17 Full Text [...]	Article 17 Revenue

Given the amount of data we have, we used an 80/10/10 split, with 80 percent of our data used for training, 10 percent used for development, and the final 10 percent used for testing. This breaks down to 36,000 articles for training and 4,500 articles each for development and testing. To analyze the article texts, we used a 400,000-word dictionary and 50-dimensional GloVe word embeddings. Preprocessing of the data was required. By sourcing our articles from a single newspaper, we automatically correct for revenue biases that can arise based on the news source itself. This data set will form the basis of our model, which we hope DeepNews.ai will be able to apply to their overall set of millions of articles in the future.

The first step of our preprocessing was to determine the correct format the data had to be in to be the model's input. Based on the models we decided on, we needed a single CSV file with two columns - the first column being the article content, and the second being the corresponding revenue class. We decided our output would be a rating of the revenue potential of a given article, 1-5. If an article had an output of 5, it meant it had high revenue potential, and conversely for an article with an output of 1, it means it had low revenue potential. We decided against having an exact revenue number as the output since it would not adjust for inflation and would almost always be inaccurate. We set the categories as: 1 - $0-2; 2 - $2-8; 3 - $8-15; 4 - $15-40; 5 - $40+.

Once we did this, we then filtered the CSV file for revenue numbers that were two weeks since publishing, so we could establish a standard for the revenue numbers, as well as a minimum article length to avoid their 404 error page which displays a very short article. After doing that, we created code that would open up the corresponding articles we were interested in from the CSV, turn it into a comma-delimited word string with no contractions, punctuation, etc., and add it to a matrix. At the end, we combined the X and Y into the "article_dataset.txt" file.

Methods

Model Selection

Our specific problem blends ideas from each of the aforementioned areas of research. Results from sentiment analyses indicate that it is possible to extract contextual, intuitive information from sentences and articles while results from popularity prediction research validate the concept of exploring the link between contextual information and content popularity. Lastly, stock analyses indicate that highly informative results can be garnered from text data with little preprocessing and straightforward neural network architectures. This may be true in part because a deep level of sentiment understanding may not matter as much to traders who care primarily about factual data, but it may matter more to news readers who may have a strong preference for reporting conducted with a certain tone or point of view. Additionally, negative news may attract more readers than positive news, or vice versa. Thus, for our study, it is important to test multiple approaches.

To help probe the degree to which contextual understanding impacts article revenue prediction accuracy, it is necessary to test two fundamentally different architectures. For deep understanding of sentence and document level context, we found the literature to consistently favor bidirectional LSTMs with high-dimensional word embedding vectors further improving performance. Thus, we adopt this architecture for our first test. Next, we draw upon inspiration from the work of Poulos and others who probed the minimum level of algorithm needed to achieve benchmark competitive results. Thus, we adopt a feed forward, deep neural network with a bag-of-words approach to string data processing for our second model.

Data Cleaning & Preprocessing

To date we have achieved two major milestones. First, have completed preprocessing of the data. The preprocessing step involves exporting the data from its native CSV file location into the Python workspace. From there, we structured the data such that each input vector element contained one word of the article text. The output for each corresponding article is the revenue generated over a normalized period. Once we correctly structured an article's data in Python, we stored it in a new file which we continuously added to. We did this until all of the articles had been correctly formatted into a single file.

The second milestone involved running our first iteration network on the Amazon cloud-based GPU. We first established, configured, and tested our connection to the server. One challenge we had was determining how to approach the problem. From the literature, we knew that a RNN would be well suited to our application; however, we had not covered the material in class and could not get early access. As such, we chose to implement a simpler model first. We based our model on the feed forward neural network and found an appropriate template. After customizing and validating on a small number of examples locally, we ran our algorithm on the full training set, using the cloud-based GPU.

Model Tuning

In order to extract higher level features in DNNs and longer sentences in LSTMs, it is beneficial to train deeper networks. To address the vanishing and exploding gradient problem, we employ batch normalization, Xavier initialization, and ReLU hidden layer activation functions. To increase algorithm learning efficiency, we use Adam optimization as described by Kingma et al., which combines stochastic gradient descent and RMS propagation. We use the recommended default hyper-parameter values of $\alpha = 0.001$, $\beta1 = 0.9$, $\beta2 = 0.999$, and $\varepsilon = 10^{-8}$. We tune α to its final value of 0.001 by iterating alpha as shown in the adjacent graph.

Approach

The problem we are addressing is one with many layers of complexity. Considerable effort has gone into related areas of study like sentiment classification and stock market prediction. As a result, a myriad of architectures and approaches exist. Our approach is to compare two distinct and foundational ones. The first model architecture is a feed-forward, deep neural network. We want to investigate how simpler models perform at predicting revenue. As we saw from our research, simple models have actually performed quite well at similar tasks. The second model is a bidirectional LSTM with GloVe word embeddings. With the second model we want to explore how increasing complexity affects the prediction accuracy. We include word embeddings to enable the algorithms to better discern word and sentence context. With the bi-directional LSTM, we draw inspiration from Tang et al. to enable our algorithm to contextualize information in not just from the text before the current timestep, but in the text after it as well with the goal of improving the algorithm's sentiment capture. Intuitively, given that individuals respond to bias, voice, and perspective in news articles, we would expect the algorithms that are better able to capture those features to perform better.

Experiments & Results

Model 1: Feed-forward deep neural net + Fasttext

For the feed-forward deep neural network, our primary metric is categorical accuracy, calculated as "how often predictions have maximum in the same spot as true values". After running 100 iterations on 25530 texts as the training set, our accuracy on 319 goes from the initial value of 0.3258312 to 0.3324351679245285.

Model 2: LSTM + Fasttext

We built a two-layer LSTM and used zero-padding to make the length smaller than or equal to the longest length of the text. Each LSTM layer consisted of 128 hidden nodes. We used 300-dimensional GloVe word embedding vectors to generate a final three-dimensional vector of size 2 x max_length x 300. We used mini-batches of size 32 to speed training. We employed dropout with a probability to 0.5 to prevent overfitting and as was utilized in the literature. Because our model is essentially a classification problem, we decided to use a softmax activation function, which generates a 0/1 with differing probabilities to get back a batch of 5-dimensional vectors. We used Adam optimization as well. We calculated our loss function as follows:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\frac{1}{N}\sum_{i}^{N}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Conclusion & Future Work

In conclusion, it is clear that it is possible to predict the class of potential revenue based on article content. We used two models to attempt this, feed forward DNN and LSTM RNN, and we saw that there was better performance from LSTM, which we expected given its propensity to be successful with word processing problems such as this one. LSTM works better as it is able to retain memory of what has passed through, which allows for an understanding of how words connect and the value they have intrinsically, which is difficult to ascertain for the feed-forward DNN.

Though the accuracy was not as high as in previous class assignments, we expect overall accuracy to be low because we assume the article title and content itself only explain a fraction of the reasons a reader may choose to read a given article. So long as the prediction accuracy is greater than random chance, we are capturing some of the relationship.

There are several next steps we would pursue if we had more resources. One, we would try to increase the accuracy of this model for Dallas Morning News. We would create a 2-layer LSTM network with 500, 100, and 50 hidden units and 50 time-steps. We will analyze the results to determine what additional steps like regularization, data augmentation, or cost function tuning will be needed to improve our results. We would continue to work with our corporate sponsors and TAs to troubleshoot and gain insight into implementation issues.

Two, we would use transfer learning to port this model to outlets like the New York Times or the Wall Street Journal. Since we only had articles from Dallas Morning News, we could not accurately use this model to predict other news outlet revenues. Of course, after the model is ported we would have to train it further based on the specific outlet, but this allows for a starting point.

Three, we would want to investigate the addition of more sentiment analysis capabilities based on our previous research. There is so much more that can be done to analyze the influence of emotions on the article's impressions and revenues, and perhaps there are other subtle elements that can contribute to this model's accuracy.

Four, it would be interesting to include more data points beyond the content, such as the section of the newspaper, the title, and other key pieces of data. Pulling in this key information could dramatically boost the accuracy, although we still hesitate to include the authors, as this could weight the results unnecessarily towards election cycle pieces.

Contributions

Lian and Johnson are in charge of data preprocessing, word embeddings, CBOW inputs, AWS setup, the baseline algorithm and project management. Griffin is involved in tensorflow, pytorch and keras tutorial learning, code maintenance, project milestone writeup, and media outreach.

References

- Xu J, Chen D, Qiu X, and Huang X: Cached long short-term memory neural networks for document-level sentiment classification. Published as long paper of Empirical Methods in Natural Language Processing (EMNLP), 2016.
- Xiao Ding, Yue Zhang, Ting Liu, Junwen Duan: Deep Learning for Event-Driven Stock Prediction. IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence, 2015.
- Sepp Hochreiter and Jurgen Schmidhuber: Long Short-Term Memory. Published in Journal Neural Computation, 1997.
- Jeffrey Pennington, Richard Socher, Christopher D. Manning: GloVe: Global Vectors for Word Representation. In the proceedings of the Conference on EMNLP, 2014.
- Jason Poulos: Predicting Stock Market Movement with Deep RNNs, 2015.
- Arnau Ramisa: Multimodal News Article Analysis, 2016.
- Leonardo Dos Santos Pinheiro and Mark Dras: Stock Market Prediction with Deep Learning: A Character-based Neural Language Model for Event-based Trading, 2017.
- Wojciech Stokowiec, Tomasz Trzcinski, Krzysztof Wolk, Krzysztof Marasek and Przemyslaw Rokita: Shallow reading with Deep Learning: Predicting popularity of online content using only its title, 2017.
- Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu: Effective LSTMs for target-dependent sentiment classification, 2016.
- Tian Wang and Kyunghyun Cho: Larger-Context Language Modelling with Recurrent Neural Network, 2016.
- Akira Yoshihara, Kazuki Fujikawa, Kazuhiro Seki, and Kuniaki Uehara: Predicting the Trend of the Stock Market by Recurrent Deep Neural Networks, 2014.