Relevance-Adjusted Sentiment for Market News

Shrivu Shankar
The University of Texas at Austin
shrivu.shankar@utexas.edu

Abstract

Media sentiment has been an important tool for investing over the last few decades. This paper addresses the issue of sentiment relevance when trying to measure attitudes with respect to a specific company. We analyze various methods for computing embeddings and sentiment using both classical and deep learning techniques to produce a more viable relevance-adjusted metric. Experimental results show intuitive changes in sentiment for the same news but different companies, interpretable company embeddings, as well as correlations with a company's stock price. Source code is available at https://github.com/sshh12/Relevance-Adjusted-Sentiment-For-Market-News.

1 Introduction

Numerous financial news websites allow users view news related to a specific stock symbol (a unique trading ID for companies and other tradable securities). While large, media-popular companies may have numerous headlines for any given day, related less popular companies will have fewer headlines. For example, a headline about new electric car regulations may only be associated with Tesla Inc. despite the large impact of the regulations on other less popular electric car companies in the US. This is likely due to the limitations of manual symbol labeling and the use of simple string-matching algorithms. When computing a metric like sentiment, this would lead to a not-so-useful volatile metric for less newsworthy companies because of the sparseness of input data. Ideally, by matching an article's higher-level topics with pertinent company domains, a system could label an article with relevant symbols even if those symbols were not directly mentioned in the article.

In this work, higher-level topics and domains are modeled using learned embeddings. These vector representations of articles and companies can be used to compute relevance and relevance weighted sentiment can be used as a metric for investors. The use of vector-based embeddings further allows for relevance to arbitrarily be calculated in a one-shot fashion (e.g. cosine similarity on article and company embeddings) without additional model training for each company. This allows for greater scale in terms of the number of articles and companies that can be analyzed than other techniques mentioned in section 2.

How the outputs of the relevance-adjusted sentiment are used will depend on one's intuitions and use case, so this paper will generally split evaluation into three categories: accuracy (how well a method achieves a subtask), price correlation (how well it correlates with a stock price), and trade signaling (profit from a naive trading strategy based solely on the metric).

2 Related Work

Significant work has been done in the field of NLP and stock market prediction. A Google search (as of March 2020) for "predicting stock prices based on news articles" yields 163 million results and 1,728 repositories on GitHub. Many of these projects rely on bag-of-words style methods which may lack the higher-level encodings needed for relevance and sentiment tasks (16) (6) (15) and those that use more sophisticated encodings tend to lack relevant sentiment (sentiment for an article calculated and adjusted for a target company) (8) (10) (9) (3).

Ming et al. (12) use sparse matrix factorization to find price and news correlations between companies and articles. This method accounts for relevance but does not allow for the querying of specific article-symbol relevances and sentiment (only symbol-symbol). This method may be better at

predicting price movements (price data is built into model), but it lacks topic-based embeddings which could be more interpretable for human investors. For example, Apple Inc.'s stock may correlate with a pharmaceutical company's despite them being in very different domains.

Peng and Jiang (13) use a graph of stocks with edges weighted by stock price correlation to determine relevance. As before, this may lead to a better metric for predicting price, but it loses some intuitions of news relevance.

Hu et al. (4) create Hybrid Attention Networks which use News-level Attention to identify relevant articles. Although relevance is included, their method only provides it for a fixed target (i.e. their model can only be trained for one sector or company at a time). This technique may fail for less media-popular companies for lack of labeled company-specific news.

Corro and Hoffart (1) similarly use deep learning-based embeddings which include adjustments for relevance. However, their method uses manually determined categories (ex. "business", "tech", "world") to model article topics. This lacks the specificity needed to achieve more precise article-symbol relevance (ex. a learned US airlines category) that could also be learned from headlines.

3 Dataset

The dataset is composed of around 220 companies. Some (36%) were manually chosen for popularity and diversity and the rest (64%) were randomly sampled from a large list of symbols. These companies were then used to query for related financial news on MarketWatch (95%), Reuters (4%), and Benzinga (1%). Via web scraping, 120k articles were collected with published dates, headlines, content, and reported symbol labels. Due to significant spareness in articles collected farther in the past, the dataset was trimmed to only articles published in the last two years. All headlines and content were stripped of non-alphanumeric characters, HTML tags, links, and repeated irrelevant text (e.g. "Subscribe for more...", "Don't miss:...").

For evaluation, daily symbol price data was collected from AlphaVantage and company metadata was scrapped from MarketWatch.

4 Article Embeddings

To produce the embedding (1) of an article (e_{A_i}), four methods are tested using both headlines and article content:

Pre-trained BERT (2): A pre-trained BERT-Large uncased model with a maximum sequence length of 512.

Fine-tuned BERT: A BERT-Base uncased model with a maximum sequence length of 256, fine-tuned [1] on the classification task of pairing symbols to articles.

Doc2Vec (7): A freshly trained Doc2Vec model [2] set to unigrams/bigrams, an embedding size of 128, and a minimum token count of 2.

Bag-of-Words: An embedding made from the top-2048 most frequent uncased tokens (unigrams and bigrams included).

BERT was chosen for its state-of-the-art performance on several NLP tasks, Doc2Vec for its specialization in document embeddings, and BoW as a baseline.

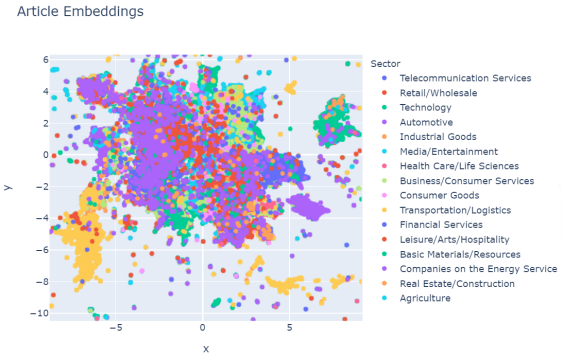

Figure 1: Article content embeddings (Reduced using UMAP (11)) using Bag-of-Words and colored by sector. The plot shows a relationship between article embeddings and their corresponding company sectors.

5 Company Embeddings

The embedding of a company (e_{C_j}, 2) is found by training a fully-connected neural network to maximize the binary accuracy of predicting the relevance r ($r_{i,j} = 1$ if article i has the label company j else 0) such that $r_{i,j} = e_{A_i} \cdot e_{C_j}$ and e_{A_i} is fixed. The number of hidden layers and other hyperparameters are varied to improve accuracy on a held out validation set.

6 Sentiment

Sentiment is tested using four techniques on both articles and headlines:

[1] https://github.com/google-research/bert
[2] https://radimrehurek.com/gensim/

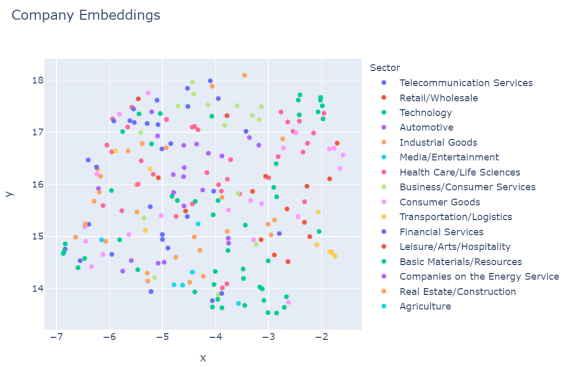

Figure 2: Company 1024-dim embeddings using BERT article embeddings and colored by sector. The plot shows a learned representation of company sectors with similar company vectors having similar sector labels.

VADER (5): A rules-based sentiment metric with a custom lexicon for financial news.
TextBlob [3]**:** A python library for determining sentence polarity.
GloVe (14): AllenNLP's pretrained sentiment model based on GloVe embeddings. To standardize the output predictions, a log-based transformation was applied to undo the final softmax layer that came with the model.
Google Cloud [4]**:** A Natural Language API for computing document sentiment.

For several articles, the relevance-adjusted sentiment is calculated by the average sentiment of each article weighted by their relevance to a target company.

7 Embedding Results

Nearly all methods of article embedding (the exception being BoW Headlines) produced vectors that when reduced, were visually consistent with their sector labels. Transportation, technology, materials, finance, etc. all had easily identifiable clusters in the embedding space. Small mixed clusters often occurred due to repetitive headline sub-sequences that were not sanitized (ex. "Top Stocks Today:...").

In the task of computing company embeddings and r (1), Bag-Of-Words had the highest accuracy classification accuracy and fine-tuned BERT did slightly better than pre-trained BERT. Content-based article embeddings yielded a significantly higher accuracy than headline-based ones. The dimensions of the the output company embedding had little impact.

[3] https://github.com/sloria/TextBlob
[4] https://cloud.google.com/natural-language

Company-Article Pair Accuracy

Method	Accuracy
Pre-Trained BERT	0.848
Fine-Tuned BERT	0.861
Doc2Vec	0.828
Bag-Of-Words	**0.921**
Headline Embeddings	0.789
Content Embeddings	**0.921**
Company Emb. 2048-dim	0.921
Company Emb. 1024-dim	**0.921**
Company Emb. 512-dim	0.921
Company Emb. 64-dim	0.917

Table 1: The maximum validation accuracy (in determining r) achieved using the given method. BoW-based and content-based methods achieve the highest accuracy.

8 Correlation Results

For a subset of stocks (AAPL, BA, BHP, DIS, GE, NFLX, TSLA, UAL), the daily relevance-adjusted sentiment score was calculated for each day over a one year time span [5]. Permutations of these scores were then created using 5, 7, 10, and 30-day moving averages. Correlations were then calculated between each of these scores and a company's stock price (2).

Every method better correlated with a company's stock price (rather than daily price changes) and trading volume. Past and current price changes had higher correlations than future ones. While content-based methods tended to perform better for past and current price changes, headline-based ones had a higher correlation with future price changes. TextBlob sentiment-based methods were scored highest and there was little difference between article embedding methods.

To compare price correlation (the Pearson correlation of daily price changes) and company similarity (the dot product between embeddings), each were calculated for every pair of companies in the dataset. This produced a weak positive correlation of 0.154, indicating a small relationship between stock price movements and domain similarity.

9 Trading Results

For a subset of stocks, each method of relevance-adjusted sentiment was tested as a signal (if $x > 0$

[5] Ideally this would be longer but it was limited by the dataset and computational power available.

Pearson Correlations

Method	Price	Today	Yesterday to Today	Today to Tomorrow	Volume
Pre-Trained BERT	0.825	0.258	0.222	0.192	0.715
Fine-Tuned BERT	0.821	0.249	0.234	**0.206**	0.692
Doc2Vec	**0.833**	0.242	0.212	0.194	0.665
Bag-Of-Words	0.826	**0.283**	**0.253**	0.199	**0.729**
VADER	0.606	0.172	**0.253**	0.137	0.432
TextBlob	**0.833**	**0.283**	0.234	**0.206**	**0.729**
GloVe	0.582	0.246	0.198	0.199	0.571
Google Cloud	0.688	0.194	0.183	0.181	0.659
Headline Embeddings	0.826	0.249	0.234	**0.206**	0.692
Content Embeddings	**0.833**	**0.283**	**0.253**	0.199	**0.729**
Headline Sentiment	0.688	0.246	**0.253**	**0.206**	0.571
Content Sentiment	**0.833**	**0.283**	0.198	0.181	**0.729**
Company Emb. 2048-dim	0.826	0.277	0.218	0.194	**0.729**
Company Emb. 1024-dim	**0.833**	0.264	**0.253**	**0.206**	0.715
Company Emb. 512-dim	0.814	0.264	0.241	0.199	0.694
Company Emb. 64-dim	0.824	**0.283**	0.196	0.192	0.727

Table 2: The maximum Pearson correlation achieved between the relevant daily sentiment of using the given method and a company's close price, open-close price change, previous day price change, future price change, and volume. Content-based methods generally performed better and higher correlations were found with past and current price changes as opposed to future ones.

BUY else SELL) to a trading simulator [6]. The maximum returns are shown 3 from January 2nd, 2019 to April 9th, 2020. The results include simulations that inversed the signal (if $-x > 0$ BUY else SELL).

Relevance-adjusted (as opposed to not adjusted) sentiment achieved higher maximum returns for every stock. Headline-based sentiment and BoW-based article embeddings most often had the highest returns. The best sentiment algorithm varied significantly.

Some abnormally high returns were the result of the algorithm trading on sentiment before the event that caused the sentiment actually occurred. Since the dataset does not include the time the articles were published nor the intraday price data, the trading simulation could use 2pm's sentiment to buy the stock at 9am. This cheat was observed in both BA and NIO's best performing strategies. Future work should account for this by including data for intraday events and price movements.

10 Discussion

All of the article embedding methods exceeded expectations in how well they corresponded to sectors. While they could have been aligned more with sentiment, market cap, success, etc. their alignment with sectors/industries shows a strong relationship between the words used in an article and the sector of the company it relates to. This makes sense as the article will likely employ domain-specific words that the model can leverage. A common error case (error here defined as similar embeddings for non-similar articles) came from the different sources of articles used. Similar headline and content sub-sequences used by a source would override domain similarity, making same-source articles clustered together over same-domain articles. Since 95% of the dataset came from the same source, this did not significantly impact results.

Although BERT has outperformed the other document embedding models in several common NLP tasks, BoW yielded better results. This is likely due to company-article pairing being a simpler task as the use of a single word (encoded by BoW but not necessarily BERT) like the company's name can give away the answer.

Intuitively, content-based methods tended to do better than headline-only ones since content embeddings include more information about the articles. In the minority of articles that describe a collection of unrelated stocks (or related in ways other than sector/industry, ex. "Top Movers Today"), the embedding will often only represent the most men-

[6] https://www.backtrader.com/

Approx. One Year Returns (as %)

Strategy Basis	AAPL	TSLA	NFLX	DIS	GE	BHP	PFE	BA	NIO	UAL
Long (None)	84	77	38	-7	-9	-16	-20	-54	-55	-67
Random	26	30	6	-6	-5	2	-14	-51	-24	-42
Pre-Trained BERT	**180**	239	90	60	**108**	42	**52**	77	335	70
Fine-Tuned BERT	169	249	95	52	79	43	45	79	294	70
Doc2Vec	175	263	**97**	54	84	48	46	82	290	**95**
Bag-Of-Words	156	**326**	93	**72**	**108**	**58**	45	**86**	**400**	70
VADER	**180**	203	77	**72**	67	45	23	62	**400**	70
TextBlob	116	286	95	40	**108**	**58**	23	82	286	63
GloVe	118	206	**97**	60	80	48	**52**	39	108	6
Google Cloud	116	**326**	80	48	84	39	25	**86**	242	**95**
Headline Embeddings	**180**	**326**	**97**	60	84	45	**52**	82	357	**95**
Content Embeddings	161	286	95	**72**	**108**	**58**	46	**86**	**400**	70
Headline Sentiment	**180**	**326**	**97**	**72**	**108**	**58**	36	**86**	235	**95**
Content Sentiment	127	286	90	60	80	48	**52**	82	**400**	70
VADER (Not Adjusted)	95	159	54	**36**	20	**1**	0	20	49	15
TextBlob (Not Adjusted)	**118**	132	43	34	21	**1**	3	15	**86**	16
GloVe (Not Adjusted)	112	222	46	28	27	**1**	**4**	**26**	84	**21**
Google Cloud (Not Adjusted)	93	**169**	**68**	34	**35**	**1**	0	20	84	12

Table 3: The max %-returns achieved using a variation of each method. Long represents the returns of just holding the stock. Methods that did not take into account relevance to unlabeled articles are marked as "Not Adjusted". Headline and BoW-based methods had the highest maximum returns. Some suspiciously high values are the result of inconsistencies between the dataset and the back-testing method.

tioned stock. This would confuse the model if the article itself is labeled with the symbol of the less mentioned stock and is likely the cause of some of the error in the embedding results.

Rules-based sentiment analysis also outperformed GloVe-based sentiment and the Google Cloud API in terms of price correlation. This does not mean that VADER and TextBlob are better at classifying sentiment, but rather that when combined with the other embedding methods, they produced the highest correlations with stock price.

While one should be skeptical of the high return results because of intraday errors (9), the data is still insightful. Adjusted-sentiment systematically outperformed regular (not adjusted) sentiment due to a richer trading signal that could include every article published in a given day. As in the other sections, BoW most often yielded the best results.

References

[1] Luciano Del Corro and Johannes Hoffart. 2020. Unsupervised extraction of market moving events with neural attention.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[3] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1415–1425, Doha, Qatar. Association for Computational Linguistics.

[4] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2017. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction.

[5] C.J. Hutto and Eric Gilbert. 2015. Vader: A parsimonious rule-based model for sentiment analysis of social media text.

[6] Kalyani Joshi, Bharathi N, and Jyothi Rao. 2016. Stock trend prediction using news sentiment analysis. *International Journal of Computer Science and Information Technology*, 8:67–76.

[7] Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

[8] Mark Levene. 2018. Market trend prediction using sentiment analysis: lessons learned and paths forward.

[9] Huicheng Liu. 2018. Leveraging financial news for stock trend prediction with attention-based recurrent neural network.

[10] Huina Mao, Scott Counts, and Johan Bollen. 2011. Predicting financial markets: Comparing survey, news, twitter and search engine data.

[11] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction.

[12] F. Ming, F. Wong, Z. Liu, and M. Chiang. 2014. Stock market prediction from wsj: Text mining via sparse matrix factorization. In *2014 IEEE International Conference on Data Mining*, pages 430–439.

[13] Yangtuo Peng and Hui Jiang. 2016. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 374–379, San Diego, California. Association for Computational Linguistics.

[14] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[15] Dev Shah, Haruna Isah, and Farhana H. Zulkernine. 2018. Predicting the effects of news sentiments on the stock market. *CoRR*, abs/1812.04199.

[16] Marc Velay and Fabrice Daniel. 2018. Using NLP on news headlines to predict index trends. *CoRR*, abs/1806.09533.