Financial News in Predicting Investment Themes

Andrew Han
Department of Computer Science
Stanford University
Stanford, CA, 94305
handrew@cs.stanford.edu

Abstract

In 2015, Eugene Fama and Kenneth French discovered that five factors explain between 71 and 94% of portfolio returns [5]. This discovery and others like it have led to the development of the "smart beta" industry and ETF products that enable investors to invest in "factors" just as they would in individual securities. Meanwhile, recent advances in distributional semantics and neural language models have enabled the possibility of computationally processing and understanding large corpuses. In my work, I propose the use of neural network models to predict relative factor returns. Using a rolling window of relevant news articles, I hope to accurately predict which among five such factors will offer the highest return for a forward-looking window. I utilize three models: a baseline two-layer neural network that uses *doc2vec*-produced document embeddings, a gated recurrent unit (GRU) model with attention, and a convolutional neural network with attention.

1 Introduction

The possibility of describing and predicting asset returns has captivated financial economists, investors, and computational scientists alike. In 2015, financial economists Eugene Fama and Kenneth French, the fathers of the efficient markets hypothesis, discovered that five factors explain between 71 and 94% of portfolio returns [5]. This discovery and others like it have led to the development of the "smart beta" industry and ETF products that enable investors to invest in "factors" just as they would in individual securities. In response, financial markets participants have recently explored the possibility of profitably "timing" the movements of these factors [14].

Meanwhile, recent advances in distributional semantics and neural language models have enabled the possibility of computationally processing and understanding large corpuses. Previous work in financial prediction by computer scientists apply classical natural language processing techniques to analyze the relationship between Securities Exchange Commission (SEC) financial reports, Twitter data, and news on prices, risk, and trading of stocks [6].

For my work, I use a variety of neural architectures to analyze financial news articles and predict relative factor returns. In particular, I make use of a simple attention mechanism to identify particular news articles that may be more relevant to forward-looking asset returns.

The task can essentially be described as a supervised document classification problem. For any instant in time, the input is a set of news articles from a backward-looking window, and labels are one or two of best performing factors from a set of five in a forward-looking window.

In some sense, it is not too dissimilar from the classic sentiment classification problems in canonical machine learning literature. On the other hand, because our task involves larger text inputs (sometimes thousands of words, depending on how many news articles there are in a given window) some sort of dimensionality reduction is needed before neural models can be feasibly applied. Ideally, such techniques will not lose too much semantic information.

Preprint. Work in progress.

2 Related Work

To this end, Le and Mikolov sought to apply the same insights and techniques of *word2vec*, which Mikolov proposed in an earlier paper, ("Distributed Representations of Words and Phrases and their Compositionality" [4]) to the task of representing variable-length documents. In the latter, either a context of words are used to predict a missing word (called a continuous bag-of-words model or CBOW) or a center word is used to predict its context (known as a skip-gram model). Every word is mapped to a unique vector, represented by a column in a matrix *W*. Those word vectors are trained using stochastic gradient descent in a neural network, with the gradient obtained via backpropagation. After the training converges, words with similar meaning are mapped to similar positions in high-dimensional vector space.

Le and Mikolov's algorithm utilize similar concepts. In *doc2vec*, paragraphs are also asked to contribute to the prediction task of the next word given many contexts sampled from the paragraph. Just as every word is mapped to a unique vector, the authors also map paragraphs to vectors, and use them to predict the next words in the sequence. The authors call this approach the Distributed Memory version of Paragraph Vector (PV-DM), analogous to CBOW in *word2vec*. The authors also propose a Distributed Bag of Words version (PV-DBOW) that is similar to *word2vec*'s skip-gram model. After training, the paragraph vector is a learned representation of the longer, variable length paragraph. The authors use the learned representation of documents to achieve state-of-the-art results on sentiment analysis and information retrieval tasks.

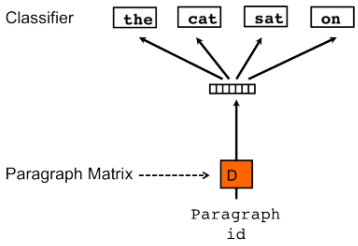

Figure 1: Distributed Bag of Words

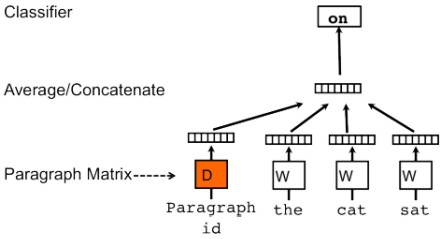

Figure 2: Distributed Memory

Also relevant to the task of learning document representations is Yang et al.'s Hierarchical Attention Networks (HANs) [13]. Building on the attention networks of Bahdanau [9] and Xu et al. [11], Yang et al. propose a hierarchical neural architecture designed to mirror the basic hierarchical structure of documents. Similarly to Le and Mikolov's *doc2vec*, the authors construct representations of documents using their natural substructures, leveraging the fact that words form sentences and sentences form a document. The core insight underlying their model is that not all parts of a document are equally relevant for answering a query and that determining the relevant sections involves modeling the interactions of the words, not just their presence in isolation. As such, their model includes two levels of attention – one at the word level and one at the sentence level, which allow the model to pay more or less attention to particular words and sentences in constructing the representation of the document.

Kim et al. [12] create a character-level convolutional language model. Instead of using pre-trained word-embeddings, the authors use a character-level convolutional encoder (consisting of a one-dimensional convolutional layer and a highway network) to generate word-level embeddings. Their work, as with Yang et al. and Le and Mikolov above, is another example in which smaller units of a task (in this example, characters) are used as building blocks in learning a representation of a larger structure (word embeddings).

3 Approach

For my task, I use two methods: first, *doc2vec*, an approach to document embeddings developed by Le and Mikolov [7] and second, embeddings learned by a convolutional encoder similar to the character-based convolutional encoder used in Kim et al.'s character-aware language models [12]. The former is attractive because the widely available *doc2vec* implementations online make using it as a component in a learning pipeline very accessible. However, such embeddings aren't trained

explicitly with respect to a supervised task, so one might surmise that training embeddings specifically for a task at hand may yield better results.

The architectures that I deploy are as follows:

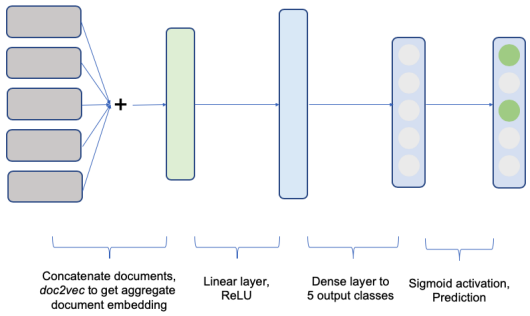

Figure 3: Baseline Feed-Forward Neural Network

- **Baseline feed-forward neural network**. The first layer performs a linear projection of an input document embedding vector (representing a single document made up of every news article from a backward-looking window) and applies a ReLu nonlinearity. The second layer performs another linear projection and either a sigmoid nonlinearity to attain the final labels.
- **Recurrent attention network**. Modeled off of Yang et al.'s work [13] in attention networks for document classification, I use a gated recurrent unit (GRU) with attention over input document vectors (which, in this model, will not be aggregated, as in the baseline network) [13]. The attention scores are multiplied with the document vectors, and the resulting tensor undergoes a linear projection to the number of classes. Finally, a sigmoid or softmax nonlinearity allows us to obtain the final label(s).
- **Convolutional document embedder, attention**. Inspired by Kim et al.'s character-level convolutional encoder, I create an embedder consisting of a single convolutional layer over the word embeddings of each word in each document, followed by a ReLU nonlinearity and a max pooling (see figure 4). These resulting document embeddings are then projected into higher dimensions and attended over (as in figure 5), before a final linear projection and sigmoid activation to obtain the final labels.

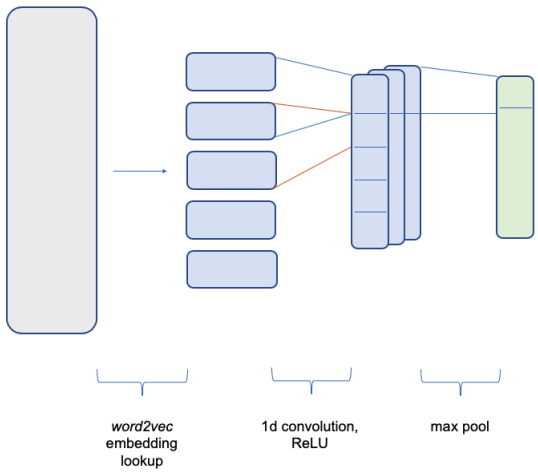

Figure 4: Convolutional Document Embedder

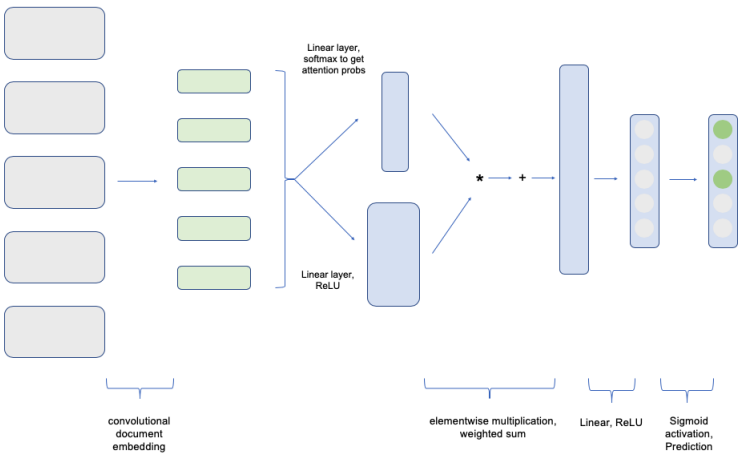

Figure 5: Attention CNN

4 Experiments

4.1 Data

Fama and French [1], as well as Applied Quantitative Research (AQR) [2], one of the world's largest quantitative hedge funds, have made data on daily factor returns public. Forward returns data will serve to make the learning task supervised. It should be noted that while Fama and French's five identified factors were

1. **Market** (the returns of a diversified market portfolio)
2. **Value** (cheaper, lower valuation companies over higher valuation ones)
3. **Size** (the returns of smaller companies over larger companies)
4. **Profitability** (also known as "quality") (more profitable companies over less)
5. **Investment** (companies that invest conservatively versus those that invest aggressively)

we have opted for a slightly different set of factors, consisting of the market, size, value, momentum (the returns of companies that have appreciated in price over those that have depreciated), and volatility (the returns of companies with smaller fluctuations in price data over those with larger fluctuations). We have opted for this alternative set for two reasons: (a) this set is closer to the ones conventionally used by financial markets participants, (the only difference being that the factors used in industry would normally swap "quality" for the market factor) and (b) because daily returns for the "quality" factor was not as readily available as those for "momentum", "volatility", and "market".

For input data, I have downloaded headlines and article metadata pertaining to "US stocks" from the New York Times API dating back to the 1950s. Using that metadata, I scraped article contents.

4.2 Evaluation

Predictions are multilabel. This is meant to avoid the problem in which, for example, the second-best prediction was in fact right (in other words, if the prediction was very close). Such an error, from an investing perspective, is mostly acceptable. To this end, a sigmoid nonlinearity in the final layer and binary cross-entropy loss is most appropriate. Regular cross-entropy loss on a softmax output would not as easily allow for backpropagation to particular classes. I define a "correct" prediction as those predictions that accurately identified at least one of the top two factors in a forward window.

The attention mechanic in the aforementioned architectures will also lend itself well to qualitative commentary on what and how the networks learn. In "peeking" at the network's activations and attention probabilities in predictions during volatile times, we can get a sense for what the network learned and which news articles it considers important.

4.3 Experimental Setup

Standard neural hyperparameters, such as learning rate, training time, and hidden dimensions were relatively fixed at $1e-4$ to $1e-5$, 10-100 epochs, and document embeddings of dimension 300, respectively. Some exploration found that varying these parameters did not affect the performance very much. The Adam optimizer was used for gradient descent.

Much more important to this particular task was the choice of forward-looking window to use. Factor returns are presented simply as a time series, so depending on the periodicity one chooses to sample the series, the size of the dataset may vary. The available data on factor returns are daily in frequency and reach back to the early 1970s. With roughly 50 years of data, if returns are sampled monthly, then there are only roughly 600 data points to work with. If sampled weekly, there would be roughly 2600. Models were trained on both a weekly and monthly basis, with one week and six month forward-looking returns.

4.4 Results

The class distribution is shown below:

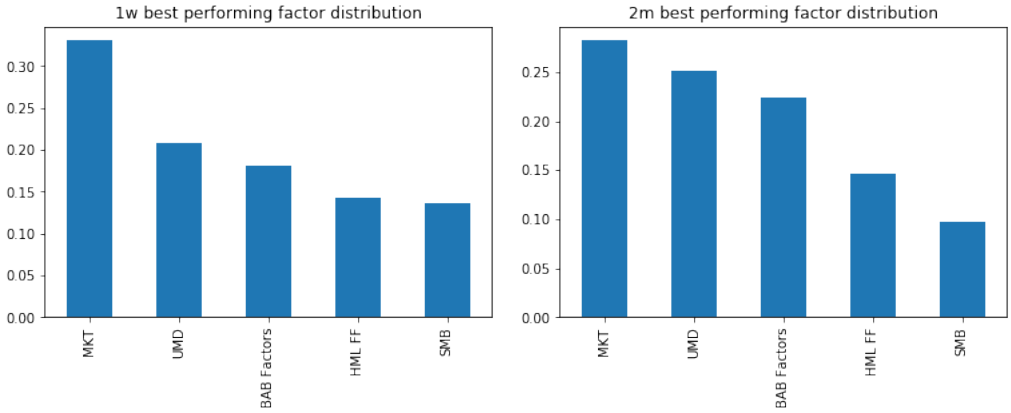

Figure 6: Best performing factors, sampled weekly and monthly

Simulations suggest that random guessing would have yielded roughly 69% accuracy. The baseline model, if fed dummy zero-tensors and the true labels, achieves up to 77% accuracy. Surprisingly, both the two layer baseline network and the attention RNN, when fed a document vector created from *doc2vec*, achieved very high validation accuracy (above 90%) within a couple dozen epochs. The attention RNN in particular learned very quickly before overfitting to the training set. Test accuracy for both the baseline network and attention RNN were over 80%.

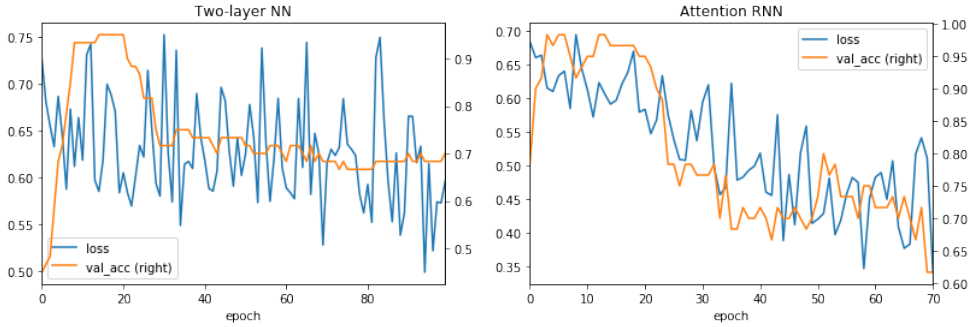

Figure 7: RNN and Baseline NN training loss and validation accuracy on monthly data

5 Analysis

5.1 Quantitative

The baseline neural network and attention network with convolutional embeddings seemed to outperform GRU with attention. This was to be expected, since the dataset was on the smaller side and the GRU, being a more complex model, was prone to overfitting. Additionally, predictions were more accurate on a longer time scale, suggesting more noise in financial markets in short-term windows.

Model	Accuracy (6m)	Accuracy (1w)
Random guessing	69%	69%
Baseline NN, doc2vec embeddings	90%	**80%**
RNN with attention, doc2vec embeddings	80%	70%
Attention network, conv. embeddings	**95%**	73%

5.2 Qualitative

One attractive feature of the attention mechanism is that it offers some insight into the features that neural networks consider important. And indeed, the document vectors that the recurrent and convolutional networks attended to showed some evidence that networks were able to differentiate between more and less relevant news articles. Consider the following two selected examples from the convolutional model with attention.

December 2007: A little less than a year before the collapse of the global economy, cracks in the mortgage bubble were beginning to show. In this case, our model accurately predicted both of the top two best-performing factors (in this case, small-cap and momentum stocks). Curiously, the most-attended article from that window was one that discussed factor investing (emphasis added, below), and the least-attended article was one discussing very technical aspects of mortgage debt and government loan programs. This might suggest that the attention mechanism learned to ignore more specific articles and attend to articles more related to assets and investing (though, with hindsight perhaps one might argue that more attention should have been paid to articles detailing aspects of the mortgage crisis).

> **Papers Study August Crisis, From First Wave to Last Ripple** ... That set off a wave of deleveraging, or selling, that in turn caused stocks to do strange things. Specifically, cheap stocks, or **value stocks**, got pummeled, and expensive stocks, or popularly shorted stocks, rose. This caused a lot of pain on the street, especially among quantitative hedge funds, or quants...

> **Shouldering the blame for subprime loan failures** ... The figures come from NeighborWorks America, a nonprofit organization created in 1978 by Congress to deliver financial aid and training to troubled urban communities. Its affiliate, the Neighborhood Housing Services of America, makes loans to home buyers of low and moderate incomes, a group that resembles the typical subprime borrower...

April 2015: Eight years after the financial crisis, the Federal Reserve began tapering the monetary stimulus it provided to the economy. Still recovering from the worst economic recession since the Great Depression, investors reacted negatively. In what has since been dubbed as a "taper tantrum," financial markets experienced a great deal of volatility in 2015. As a result, unsurprisingly, the best factors for the six months following April 2015 were "size" and "betting-against-beta" (or, small and low volatility stocks). Our model predicted that small stocks and momentum stocks would do well, though it was a toss-up between low volatility stocks and momentum stocks. Further, the most- and least-attended articles seemed to suggest careful consideration of markets-related articles, and effective screening of unrelated articles. Below, the most-attended article was an opinion piece on mutual fund management with a great deal of markets-related content, and the least-attended article had to do with shipping in Canada.

> **How Many Mutual Funds Routinely Rout the Market? Zero** The bull market in stocks turned six last Monday, and despite some rocky stretches — like last

week, when the market fell — it has generally been a very pleasant time for money managers, who have often posted good numbers.

Deep Freeze on Great Lakes Halts Cargo Shipments THUNDER BAY, Ontario — The trip to pick up a load of iron ore powder in Conneaut, Ohio, was supposed to take four days by way of the Great Lakes.

6 Conclusion

Using relatively little data, models using a convolutional embedder and *doc2vec* yielded document embeddings that, when fed to other neural architectures, seemed to be mildly predictive of relative factor returns. This suggests that there is some relationship between financial news and short-term asset movements and investment themes. Further, insight gleaned from adding an attention mechanism also suggests that, when supervised against empirical forward asset returns, neural network methods are able to differentiate between news article that are more or less relevant to financial markets.

This initial exploration seems promising. Future work may extend this inquiry along two main axes. First, it may gather and analyze more news articles (across different news outlets, time periods, etc.), in different languages across many countries (and as a result, across more equity markets), and with a wider variety of assets. Second, further exploration may make use of more complex and deeper neural models, with attention across a wider variety of subcomponents, and may model the temporal and sequential nature of news articles more explicitly than we have.

Acknowledgments

Thanks to Michael Hahn for his mentorship.

References

[1] Fama, French. *Current Research Returns*. http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_5_factors_2x3.html

[2] Applied Quantitative Research. *Betting Against Beta: Equity Factors, Daily*. https://www.aqr.com/Insights/Datasets/Betting-Against-Beta-Equity-Factors-Daily

[3] Fama, French. 1993. *Common risk factors in the returns on stocks and bonds*.

[4] Mikolov et al., 2013 *Efficient Estimation of Word Representations in Vector Space*. https://arxiv.org/abs/1301.3781

[5] Fama, French. *A Five-Factor Asset Pricing Model*. 2014.

[6] Lee et al. 2014 *On the Importance of Text Analysis for Stock Price Prediction*. https://nlp.stanford.edu/pubs/lrec2014-stock.pdf

[7] Mikolov, Le, 2014 *Distributed Representations of Sentences and Documents*. https://arxiv.org/abs/1301.3781

[8] Kim, 2014. *Convolutional Neural Networks for Sentence Classification*. https://www.aclweb.org/anthology/D14-1181

[9] Bahdanau, 2014. *Neural Machine Translation By Jointly Learning to Align and Translate*. https://arxiv.org/pdf/1409.0473.pdf

[10] Zhang, Wallace. 2015. *A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification* https://arxiv.org/pdf/1510.03820.pdf

[11] Xu et al, 2015. *Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention*. https://arxiv.org/abs/1502.03044

[12] Kim et al., 2016. *Character-Aware Neural Language Models*. https://arxiv.org/abs/1508.06615

[13] Yang et al., 2016. *Hierarchical Attention Networks for Document Classification*. https://www.cs.cmu.edu/ hovy/papers/16HLT-hierarchical-attention-networks.pdf

[14] Bender et al., 2017. *The Promises and Pitfalls of Factor Timing*.