The Impact of Structured Event Embeddings on Scalable Stock Forecasting Models

Janderson B. Nascimento[1,2]
janderson.nascimento@fpf.br

[1]FPF Tech
Av. Danilo Areosa 1170
Distrito Industrial
Manaus-AM, Brazil

Marco Cristo[2]
marco.cristo@icomp.ufam.edu.br

[2]Federal University of Amazonas
Institute of Computing
Av. Rodrigo Otávio 6200, Japiim
Manaus-AM, Brazil

ABSTRACT

According to the *efficient market hypothesis*, financial prices are unpredictable. However, meaningful advances have been achieved on anticipating market movements using machine learning techniques. In this work, we propose a novel method to represent the input for a stock price forecaster. The forecaster is able to predict stock prices from time series and additional information from web pages. Such information is extracted as structured events and represented in a compressed concept space. By using such representation with scalable forecasters, we reduced prediction error by about 10%, when compared to the traditional auto regressive models.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Natural Language Processing; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Stocks Forecast; Open Information Extraction; Natural Language Processing; Deep Learning

1. INTRODUCTION

It is well known that financial prices are mostly or completely unpredictable, according to the efficient market hypothesis (EMH) [7]. In particular, EMH states that share prices to always incorporate and reflect all relevant information because market agents operate in a rational fashion. However, a weak version of EMH concedes that although prices already reflect all past publicly available information, they do not instantly change to reflect new public information, which can be explored by forecasters. Also, the strong version of EMH is criticized as being overconfident on the belief of rational markets. According to critics of EMH, imperfections are empirically observed in financial markets due to a combination of overconfidence, overreaction, representative bias, information bias, and various other predictable human errors in reasoning and information processing.

Maybe as consequence of such shortcomings, many meaningful advances have been achieved in the last years on the task of anticipating market movements using machine learning techniques. Most techniques explore the trend of people following rumors spread by News published, for instance, in web pages and social media. Thus, the understanding of finance News content may provide useful information to forecast stock price movements.

To understand text content, many information representation techniques have been adopted. They range from the simple bags-of-words to complex natural language processing (NLP) strategies such as named entity recognition (e.g., determine that *Apple* is an entity in sentence $S =$ "Apple Inc has sued Samsung"), noun phrase extraction (e.g, to detect *Apple Inc* as an entity in S), and structured event extraction (e.g: to detect {*Apple Inc, sued, Samsung*} as an event in S). Thus, traditional methods to forecast stock prices based on News rely on auto-regressive predictors that incorporate News content as overlay data. In such models, prices are taken as a time series where future prices are seen as a combination of past prices and concepts represented by sets of words, phrases or structured events.

More recently, many studies on Natural Language Processing have favored the adoption of conceptually distributed representations of language events such as words or sentences. In this way, words or sentences are represented as points in concept spaces, such that conceptually related events are close in the space. For example, in *word embeddings*, each word is represented as a vector in a conceptual space M such that conceptually close words (e.g.: *uncle* and *aunt*, *lion* and *tiger*) are not expected to be distant in M.

In this work, we propose the use of structured events projected into a compressed concept space instead of traditional representations. We study the impact of our proposal on the performance of some classical methods for time-series prediction that take advantage of additional text content. We test the selected methods to forecast stock values of the S&P500 index. In particular, as predictors, we evaluate auto-regressive models and a random-forest tree ensemble as they are scalable for real work loads. From the experiments we conducted, we observed an error reduction of about 10% even using much less information to represent the web News stream. Thus, our approach makes it feasible the learning of highly scalable forecasting models.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
WebMedia'15, October 27–30, 2015, Manaus, Brazil.
© 2015 ACM. ISBN 978-1-4503-3959-9/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2820426.2820467.

2. RELATED WORK

Recent years have seen a large increase in the adoption of data extracted from the web and social networks to attempt to create better predictive models in various application fields [18].

Simple auto-regressive analysis is the classic alternative for time series prediction tasks, such as traditional stock market prediction. However, in recent years successful modern models not based on auto regression have been proposed as, for instance, the work by [15]. Shallow and deep learning techniques have been used in stock price forecasting, such as artificial neural networks, support vector machines [10] and ensembles, since that discovering recurrent, interesting, and useful patterns in time series data is a non-trivial task [8]. For a full revision of the literature, we refer the readers to the surveys published by [18] and [8].

Prior predictive models using NLP techniques were based on shallow features as input, such as bag-of-words. However, Ding *et al.* [5] achieved a substantial improvement on stock-price movement (up and down) prediction by using structured events and a multilayer feedforward neural network. Using the same dataset, Peng and Jiang [16] also adopted structured events to analyse market movement polarity with deep neural network and correlations between companies. Unlikely these works, we here adopt structured event embeddings as input, that is, structured events represented in a space learned to be conceptually sound.

3. STRUCTURED-EVENT EMBEDDINGS

Structured-event embeddings are structured events (SEs) represented in concept spaces learned such that SEs share a conceptually-sound distributed representation. In this section, we define structured events, describe how to extract them from text, and how to code them in a concept space.

3.1 Open Information Extraction

A text content can be viewed as a sequential stream of events. An event is composed of an action P, an actor O_1 that conducted the action, and an object O_2 on which the action is performed. Formally, an event is represented as:

$$E = (T, O_1, P, O_2) \quad (1)$$

where T is the timestamp, O_1 is the actor, P is the action and O_2 is the object [5]. As an example, the News fragment "Microsoft purchases Nokia's phone business" should be extracted as {*Microsoft, purchases, Nokia's phone business*}.

We translate raw text data into events using the Open Information Extraction [1] approach. This method breaks the document sentences into structured events. As OpenIE toolbox, we used Reverb [6], an extractor for verb-mediated relations. There are some alternative ways to do that, like Ollie [13] and Semantic role labeling [4], which we intend to better investigate in future work.

For this work, we extracted and processed events from datasets provided by Bloomberg and Reuters News agencies, obtaining a total of 568,820 individual events. Figure 1 illustrates the extraction of structured events from two consecutive days.

3.2 Embeddings

Embeddings were first introduced by [2]. In that work, words were represented by neuron activations of a hidden layer of a neural network built to distinguish correct sentences ("the cat sat on the bed") from random sequences of words ("bed the on sat"). As result, related words are represented by similar patterns of neuron activations.

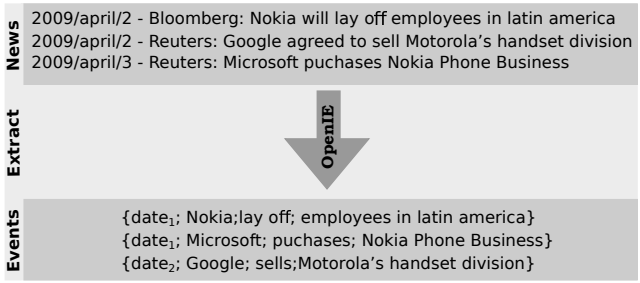

Figure 1: Extracting events from news stream. In the real extraction we conducted, we used the whole content of the documents instead of just titles, such as in this illustrative example.

Such representations capture semantic regularities of the words that allow for semantic arithmetic. For example, the operations vector('Paris') - vector('France') + vector('Italy') results in a vector very close to vector('Rome'), as vector('king') - vector('man') + vector('woman') is close to vector('queen'). Semantic arithmetic also implies that the meaning expressed by a set of words can be captured by the summation of the constituting words.

More recently, very effective ways to obtain similar representations have been introduced in literature such as the unsupervised algorithms Conceptual Bag of Words (CBOW) and Skip-gram with Negative Sampling (Skip-gram) [14]. Those algorithms learn word representations that maximize the probabilities of a word given other contextual words (CBOW) and of a word occurring in the context of a target word (Skip-gram). As CBOW and Skip-gram are unsupervised, they are able to effectively learn patterns from tens of billions of word occurrences.

Thus, to create a structured event embedding, we treat each event as a set of words, where each word is represented using the Skip-gram algorithm. More specifically, each word is represented by a feature vector in a 100-D feature space. The structured event is represented by the summation of its constituting word vectors, as illustrated in Figure 2. In this work, we use the skip-gram implementation by Mikolov *et al.* [14], referred to as *Word2Vector* and publicly available.

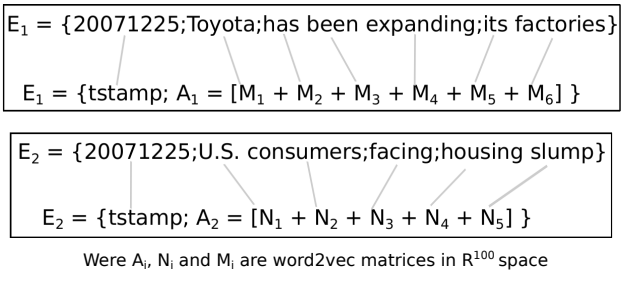

Figure 2: Structured event representation using word embeddings

We then added all vectors in same timestamp, resulting in the equation 2, which defines the resulting vector for a particular timestamp:

$$E_{time} = \sum_{i=1}^{N_{events}} \sum_{j=1}^{N_{words}} A_{ij} \quad (2)$$

Where A_ij is the vector that represents a word in a particular event.

4. PRICE REPRESENTATION AND PREDICTORS

In this work, we treat the problem of price forecasting as an auto-regression with overlay data. Thus, the target price at time t ($\hat{y}_t$) is modelled as a linear combination of prices at times $t-1$, $t-2$, ..., $t-n$ (that is, lag variables y_{t-1}, y_{t-2}, ..., y_{t-n}), where periodicity is daily and weekend days are excluded as target and lag variables. To help capture long-term temporal patterns, we also incorporate into the model quadratic and cubic time transformations (t^2 and t^3). Still regarding time, we also represented the target price using categorical temporal attributes (month and quarter) and interaction variables associated with each lag variable ($(t-1)y_{t-1}$, $(t-2)y_{t-2}$, ..., $(t-n)y_{t-n}$). The News information is represented as overlay data. More specifically, in our approach, the resulting 100-D vector embedding of timestamp t (that includes all the structured events observed at t) is also used to represent the price at t.

5. EXPERIMENTS

In this section we describe datasets, models and results used in our experiments.

5.1 Dataset

In our experiments we use the same dataset used by [5]. This dataset consists of publicly available financial news from Reuters (106,521 documents) and Bloomberg (447,145 documents), gathered from October 2006 to November 2013, and stock market prices from Standard & Poor's 500 (S&P 500) index, obtained from Yahoo Finance. This dataset is available on the web sites http://pan.baidu.com/s/1ntxCrNz and http://pan.baidu.com/s/1kTxGvKN.

This dataset was obtained by extracting on-line information directly from the HTML content of finance News pages. Due to time constraints, we sampled the whole dataset, such that only data from 2006-10-20 to 2009-12-31 is used in our experiments. This sample includes the international financial crisis that occurred in 2008. This event is particularly important, because it is often cited on empirical studies that challenge the strong version of the efficient market hypothesis. Many operations observed in that event are seen as driven by irrational fear which resulted on large losses.

5.2 Training the Model

From the news pages we extracted 1,155 structured event embeddings, each of which contains a compressed representation of the events on a particular day. These embeddings were used as non-standardized overlay data when training our proposed regression models *AR+News* and *RF+News*. In particular, the time series forecast methods used to build our models are (1) an efficient linear auto regression predictor based on gradient descent and (2) a well known tree ensemble, a Random Forest [3], with 1000 50-nodes trees. Both methods use seven lag variables ($n = 7$, cf. Section 4). The prediction task consists in determining the S&P500 daily index. The S&P 500, or the Standard & Poor's 500, is a publicly available stock market index based on the market capitalization of 500 large companies having common stocks listed on the The New York Stock Exchange (NYSE) or in National Association of Securities Dealers Automated Quotations (NASDAQ).

Note that parameters such as the number of nodes (features) and trees will be better investigated as future work. We built the models using the Weka tool with the time series plug-in [9] enabled.

We compared the methods using traditional loss metrics used in regression, that is, the Rooted Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE). RMSE and MAPE are given by Equations 3 and 4, respectively.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2} \quad (3)$$

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\frac{|y-\hat{y}|}{y} \quad (4)$$

where N is the number of instances, $\hat{y}$ is the prediction made by the forecaster, and y is the correct value.

5.3 Results

In Table 1, we compare our proposed models (AR+News and RF+News) with a plain Random Forest (RF) and a traditional linear auto-regression predictor (AR). AR and RF do not take into account News information. AR+News and RF+News include structured events embeddings as overlay data for the time series. As we can see, our methods, *AR+News* and *RF+News*, achieved a large gain over traditional AR and RF models, showing that overlay News information is useful to enhance prediction. In particular, *RF+News* achieved a gain of about 10% (RMSE) on error reduction over RF without using embeddings. It is important to note that, with embeddings, input representation is much compressed compared to traditional NLP approaches.

Table 1: Comparison among different models

Model	MAPE	RMSE
AR	1.2125	19.0963
AR + News	1.1189	17.6289
RF	0.6116	9.385
RF + News	0.5758	8.443

6. CONCLUSION

As observed, our method performed better than a simple linear auto-regression and also achieved gains over a random forest method when using structured event embeddings. But above all, the representation using structured event-vectors in concept spaces was able to achieve significant reduction on the number of features used to represent the news

streams. This way, more complex architectures, such as random forests, can be effectively used as predictors. In fact, we can also use complex models, like deep learning architectures with our compact event representation by combining Embedding methods and Open Information Extraction.
To find a more scalable and precise model, we will repeat the experiments using the entire dataset provided by [5]. We intend to study different model parameters. We also note that the pre-processing technique presented in this work can be used for reach scalability for on-line applications on other complex prediction models, such a Convolution neural networks [12], Recurrent Neural Networks [17], and Long Short Term Memory models [11], which are more able to capture the temporal dynamics observed on stock forecasting.
As an alternative, we intend to experiment with other models for pre-processing structured information from on-line news, like training a particular shallow multilayer perceptron for input features on embedding phase instead of only summing term-vectors.

Acknowledments

We would like to thank Jun-Wen Duan by kindly providing the datasets used in this work. We also would like to thank FPF Tech for allowing Janderson Nascimento to perform the research experiments in their office hours. This research was financially supported by CNPq and FAPEAM grants and fundings associated with Marco Cristo.

7. REFERENCES

[1] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Janara Christensen, Stephen Soderland, Oren Etzioni, et al. Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 52–60. Association for Computational Linguistics, 2010.

[5] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Using structured events to predict stock price movement: An empirical investigation. 2014.

[6] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[7] Eugene F Fama. The behavior of stock-market prices. *Journal of business*, pages 34–105, 1965.

[8] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, 2011.

[9] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[10] Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[13] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*, 2012.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[15] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[16] Yangtuo Peng and Hui Jiang. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. *arXiv preprint arXiv:1506.07220*, 2015.

[17] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[18] Sheng Yu and Subhash Kak. A survey of prediction using social media. *arXiv preprint arXiv:1203.1647*, 2012.