Deep Language Classification for Relabeling of Financial News and its application in Stock Price Forecasting

Giulio Trichilo
École Polytechnique Fédérale de Lausanne
In association with The Jožef Stefan Institute
giulio.trichilo@gmail.com

Miha Torkar
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
miha.torkar@ijs.si

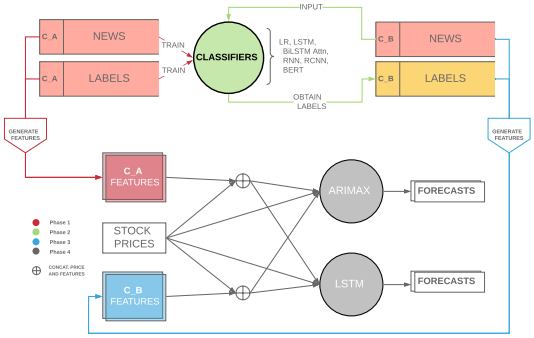

Figure 1: Workflow of the employed methodology.

ABSTRACT

This paper aims at assessing the performance of the transfer learning task consisting of training set of classifiers on high frequency financial news data for 74 publicly traded companies, with domain specific labels. This source of data is provided by the Jožef Stefan Institute and is used exclusively for the purposes of this research. The trained classifiers are then used to attribute labels to an unlabelled source of high frequency aggregated news, *Event-Registry*. The aim is for the relabelled data to be used in the generation of exogenous features for use in time series forecasting of the companies' prices. It is found that using a fine-tuned BERT [1] model yields the most semantically coherent labels, and the features generated from the newly labelled data prove to yield the highest accuracy forecasts on held out price data.

Keywords

Deep Learning, NLP, Language Model, Finance, BERT, Stocks, Forecast

1. INTRODUCTION

In recent years both natural language processing and algorithmic feature and signal based trading have been subject to an increasing level of automation, with statistical methods arguably being at the core of both. While the methods developed in both fields are still largely disjoint, in both these fields there is an attempt at modelling sequential data generating processes, be it natural language or price signals. Furthermore, empirical evidence strongly suggests that the dissemination of news regarding a financial entity such as a publicly traded company will in some way or another affect how active market participants will react.

This paper begins by characterizing the data from both corpora: the domain-specific labelled corpus and the *EventRegistry* corpus will hereby be referred to as C_A and C_B respectively. This data is initially used for the training of word2vec [7], doc2vec [5] and fasttext [2] to generate word embeddings to be used downstream in a set of deep language classifiers: LSTM, BiLSTM with Attention [6], CNN [3], and RCNN [4]; a logistic regression model serves as a baseline. Furthermore, BERT is employed, however as is standard practice, pretrained word vectors constitute the model's initial state which is then fine tuned on the C_A. The trained classifiers are then used to attribute labels from C_A to C_B. While there is no standard metric for the evaluation of the pertinence of the attributed labels, this is approached through employment of semantic similarity metrics and t-SNE in an attempt to extrapolate a relationship between semantic and projected spatial clustering.

The time series forecast setting consists of two separate subtasks performed twice, once on the data from C_A where the true labels are available, hence the feature vector construction is not subject to potential mislabelling in terms of semantic incoherence, then again on the relabelled C_B corpus with BERT labels as basis for feature construction.

Each set of features is used firstly as a series of exogenous regressors in an ARIMAX setting, this primarily in order to gauge coefficient significance and quality of the forecast with respect to the baseline of no exogenous regressors. Concurrently, each set of features is used as inputs to a two layer LSTM followed by a feed-forward net, which allows forecasting of both the stock price and the news series, however it is to be kept in mind that the explicit exogeneity relationship which characterizes ARIMAX is not maintained in the LSTM setting.

2. DESCRIPTION OF THE CORPORA

Corpus C_A consists of 3M timed news headlines between Jan 1st 2006 and Dec 31st 2018. For each headline, the associated label as well as the company in question are given. There are 53 labels however the distribution of labels over the news entries is heterogeneous, resulting in an imbalanced dataset. Therefore a balanced subset of the 120,000 entries per label from the 20 most frequent labels (yielding 2.4M entries) was selected. Of these, the train, validation, test split was selected as 75/15/10. This data split is used in the training, validation and testing of all classifiers examined. The selected subset of data contains news entries from 74

	mean	min	25%	50%	75%	max
C_A	4389	20	1298	2493	5550	37800
C_B	13489	43	2029	5522	14815	236856

Table 1: Distribution of news counts for the 74 companies.

publicly traded companies.

On the other hand, C_B consists of roughly 1M timed news headlines from Jan 1st 2014 to May 31st 2018, exclusively for the 74 companies examined. This is the corpus on which labelling is to be performed. In Table 1 summary statistics for C_B and for the subset of the C_A consisting only of the 74 companies examined is presented, both at their unadulterated frequency. C_A and C_B have a mean news headline length of 11.05, 11.36, with standard deviation of 4.52 and 9.34, respectively. Their empirical distributions are approximately χ^2-distributed.

Finally, a second corpus from *EventRegistry* consisting of 50 *dmoz* labels was made available, however this corpus contains no associated company information. From this corpus, only those headlines whose class belongs to a subset of 20 top level categories, chosen by hand due to similarity with C_A's labels, has been kept. This data subset (hereby corpus C_{B2}) is not used in modeling and plays only a very minor role in the evaluation of the relabelling performance in section 3.3.

3. CLASSIFICATION AND LABELLING

All classifiers are trained on C_A according to the chosen split. This section begins by outlining the methods used for word embedding generation used in all classifiers except BERT, then covers overall model performance. In order to assess the ability for a given classifier to attribute semantically consistent labels to C_B corpus, cosine similarity between the label vector and its neighborhood, defined here as the subset of the 30 words with the highest empirical probability of occurring for each label, according to each classifier, is computed.

3.1 Generation of Word Embeddings

In standard literature, in order to perform text classification the elements of a labelled corpus $C = \{(c, D)\}$, where (c, D) is a class-document pair, the elements $w \in D$, where $D \subset V$ and V is the vocabulary, must be mapped to a vector space, typically $\mathbb{R}^n$, where $n = \{|V|, \mathbf{d}\}$ depending on whether a count based model is used or whether one aims to represent each word as a (typically dense) $\mathbf{d}$-dimensional vector.

In general, a neural embeddings model aims at finding

$$\hat{\theta}, \hat{\mathbf{E}} = \underset{\theta, \mathbf{E}}{\operatorname{argmin}}\ L$$

Where $\mathbf{E} \in \mathbb{R}^{|V| \times \mathbf{d}}$ is the embeddings matrix which can be then passed on to downstream tasks such as text classification, and L is a loss function over the corpus, the context for each word in the corpus, given the embeddings matrix, and all other trainable parameters θ.

In this paper, word2vec, doc2vec and fasttext[1], using Contextual Bag of Words, Distributed Memory (DM), and Bag of Tricks respectively, are used to obtain three separate embeddings matrices given the training corpus. The embeddings are chosen to have $\mathbf{d} = 300$. All models were trained for 20 epochs, with a minimum count of 4, and a context window of size 7. All other parameterizations are as in [7], [5], [2], respectively.

3.2 Classifier performance on C_A

In this section performance of the LSTM, BiLSTM with Attention, CNN, and RCNN, and BERT, is analyzed. Results of training a logistic regression model serve as a basis for comparison.

3.2.1 Logistic Regression

In order to gauge classifier performance all generated word embeddings are used in training a logistic (softmax) regression classifier, as this is taken to be the simplest model trainable on the data[2]. This classifier aims at maximizing

$$P(c \mid D) = \text{softmax}\left(W_c \sum_{w_i \in D} \text{embed}_{\mathbf{E}}(w_i)\right)$$

where the summation term yields the embedding for the document[3]. The classifier is trained on all three sets of word embeddings, with 72% average class accuracy for fasttext, 69% for word2vec and 68% for doc2vec. The labels attributed to misclassified samples for each class are generally evenly distributed amongst the other 19 classes.

3.2.2 Deep Word Embedding Classifiers

LSTM, BiLSTM with Attention, CNN, and RCNN are the four deep classifiers tested. As fasttext embeddings have yielded the highest accuracy on C_A, these will be the embeddings used for these models. This choice does not in general guarantee classifier optimiality, however it gives grounds for standardized comparison. In order to further enforce this, all LSTM-based models were trained with the following common hyperparameters:

|V|	**d**	LSTM_out	batch_size	epochs
263,088	300	256	64	5

For all LSTM-Based models the initial hidden and cell states were set as $(h_0, c_0) = (\mathbf{0}, \mathbf{0})$. For the CNN the following hyperparameters were given. The model was trained for 5 epochs with the same embeddings as the previous cases. Furthermore, one channel was used in input and eight in output. Kernel sizes were 2,3,4, the stride was set to 2 for all layers and the vertical padding to 1.

All models were trained using Cross Entropy as the loss function and ADAM as the optimizer, with a learning rate $\eta = 10^{-3}$, no weight decay, and numerical stability parameter $\varepsilon = 10^{-8}$.

[1]No subword information was used as no significant accuracy was gained in subsequent use of the embeddings.

[2]Logistic Regression with Bag of Words as input, trained on a subset of data exclusively from the year 2017, yields an average class accuracy of 70%

[3]Obtaining the document embedding from the word embeddings is not a trivial problem, however addition is sufficient for the purposes of this classifier.

3.2.3 BERT

BERT leverages masked language modeling and the enocoder from the transformer archictecture in order to learn contextually coherent word representations. Unlike the previous cases BERT is initialized with its own pre-trained embeddings; all hyperparameters are kept as in BERT-Base as specified in [1]. The model was trained for 5 epochs.

Given that BERT uses wordpiece for tokenization, the size of its pretrained vocabulary is not indicative of the true dimensionality of vocabulary space. The model was adapted for classification trained using Cross Entropy as the loss function and ADAMW as the optimizer, with a learning rate of $\eta = 10^{-3}$. Furthermore a scheduler with a linear warmup is implemented, with 100 warmup steps.

For all models, a weighted average of precision and recall, along with the F1 scores of the best and worst scoring classes are given in Table 2.

3.3 Evaluation of Labelling on C_B

In order to attempt at quantifying the pertinence of the domain-specific labels attributed to C_B, the cosine similarity between the label and and the 30 most frequent words attributed to it (net of english stopwords and special characters), constituting a threshold on the empirical distribution of words for each label, is computed for all classifiers; then, the empirical similarity quartiles are computed for said classes, and the maximum over all classes for each quartile is reported[4]. In order to have some idea of how this compares to labelled data, this is repeated both for C_A and for C_{B2}. The results are reported in Table 3[5].

In accordance with intuition, those labels with worse test performance across models have a less relevant set of top words associated to them. It is interesting to note how the similarity between BERT's attribution of C_A's labels on C_B is in all cases higher, and the standard deviation lower, than is the case with C_{B2}. It is to be noted that these are not fair grounds for comparison as the corpora are different, however this does point to BERT's ability to capture semantic similarity in a more 'natural' manner than the other models.

4. FEATURE GENERATION FROM NEWS

Feature vectors are constructed by taking the relevant news events for each company for all trading days between Jan 1st 2014 to Dec 31st 2017. For each trading day, for each company, the count of the events for each category is assigned as the elements of the feature vectors (20 dimensional). The labels are the original ones for C_A, and C_A's BERT-attributed labels for C_B. The price series data used is the daily close price adjusted for dividends. The following operations were performed in order to assure consistency in the construction of feature vectors, for each company:

- For each day, obtain the feature vectors as described above for three time intervals: Pre-Hours (00:00-09:30), During Trading Hours (09:30-16:00) and After Hours (16:00-24:00). Any day over the entire year (365 days) where no events happen is attributed a zero vector.[6]
- Given the adjusted close price is being used, the assumption is made that today's close will be affected by news from today's pre-trading hours, today during trading, as well as yesteday's after hours. Therefore yesterdy's after hours vector is added to today's pre-trading hours vector and to today's trading hours vector.
- Given that the trading days a year are 252, feature vectors indexed at a non trading day are made to contribute to the next trading day (ex: the resulting feature vectors for a weekend are added with next monday's).

This construction assures the removal of any look-ahead bias (we are only interested in the scenario where the news affects the price, and not when the news event manifests itself as a reaction to a change in the stock price), however this construction does assume that news on a given day takes at most one trading day to incorporate into price.

5. FORECASTING USING NEWS

In this section the predictive performance for feature vectors generated from both C_A and the C_B with BERT-attributed C_A labels will be evaluated. The training period is the first three trading years: Jan 1st 2014 - Dec 31st 2016, and the held out period is the last 52 weeks.

5.1 Features as exogenous variables

An ARIMAX model is initially employed to test for significance of the categories of the events. In this setting, each dimension of the feature vectors constitutes a univariate time series. It is therefore these 20 exogenous series which are used as regressors in the ARIMAX setting.[7] For each price series the optimal order, ARIMA(p, d, q), is computed based on SBIC, and the inferred order is maintained when including the respective exogenous variables[8]. It is found that 3.68 ± 2.07 categories are statistically significant in predicting the price for C_A, and 1.78 ± 1.55 for C_B.

5.2 Features as inputs in LSTM

A unidirectional two-layer LSTM network is employed in order to gauge performance of price as well as news forecasting. The inputs to the networks are, for each time step, the 10 previous observations for both the close price and the 20 news series. Minmax scaling is used in order to render the input space more isotropic and promote gradient stability; all variables are then rescaled after training.

In Table 4 error metrics are computed for the holdout period from Jan 1st 2016 to Dec 31st 2017 (the final year of data). The Diebold-Mariano test is computed pairwise for each forecast: C_A, C_B, and the vanilla ARIMAX and LSTM

[4]The maximum is taken as the relabelled dataset is in all cases unbalanced.

[5]In addition, t-SNE is used to project label and neighborhood into $\mathbb{R}^2$; observable clusters are, expectedly, less well defined on the relabelled C_B than the clusters identifiable when projecting C_A.

[6]This yields 22K, 40K, 49K events for C_A, and 218K, 270K, 367K events for C_B, for the respective brackets.

[7]The training period must for some stocks be lengthened to compute coefficient significance (guarantee exogenous nonsigularity).

[8]Inferred order directly including exogenous series would sometimes yield $p = q = 0,\ d = 1$; this is never the case on just the series.

Model	Embed	Wavg. Precision	Wavg. Recall	Best Class	F1	Worst Class	F1
LR	fasttext	72%	70%	Exploartion	1.00	Insider-Trading	0.35
LR	word2vec	72%	69%	Exploration	1.00	Insider-Trading	0.35
LR	doc2vec	73%	68%	Credit	1.00	Insider-Trading	0.28
LSTM	fasttext	74%	74%	Credit	1.00	Labor-Issues	0.46
BiLSTM	fasttext	71%	68%	Investor-Relations	0.85	Marketing	0.40
CNN	fasttext	75%	74%	Credit	1.00	Analyst-Ratings	0.41
RCNN	fasttext	75%	73%	Exploration	0.99	Insider-Trading	0.44
BERT	BERT	79%	78%	Legal	1.00	Stock-Prices	0.52

Table 2: Model Performance on C_A's Test Set.

	LR	CNN	RCNN	LSTM	BiLSTM	BERT	C_A	C_{B2}
mean	0.142	0.103	0.118	0.119	0.148	0.388	0.507	0.281
std	0.282	0.274	0.285	0.274	0.272	0.294	0.381	0.334
min	-0.219	-0.213	-0.213	-0.213	-0.213	-0.107	-0.054	-0.150
25%	-0.021	0.009	-0.021	-0.017	0.067	0.176	0.365	0.143
50%	0.165	0.126	0.155	0.107	0.145	0.395	0.579	0.282
75%	0.316	0.269	0.294	0.301	0.279	0.610	0.762	0.435
max	0.666	0.667	0.668	0.666	0.745	0.802	1.000	1.000

Table 3: Maximum cosine similarity quartiles across all classes for all models on C_B. The last two columns act as a baseline showing similarity scores for the two labelled corpora.

	mae	rmse	minmax	D.M.		
ARMIAX						
NONE	4.916	5.898	0.063	-	24	25
$\boldsymbol{C_A}$	4.399	5.614	0.055	35	-	44
$\boldsymbol{C_B}$	4.376	5.482	0.061	31	47	-
LSTM						
NONE	7.158	8.033	0.103	-	20	31
$\boldsymbol{C_A}$	1.553	1.930	0.018	22	-	41
$\boldsymbol{C_B}$	1.001	1.348	0.015	50	54	-

Table 4: Median forecast error metrics across all stock prices and forecast disparity counts between models (number of stocks for which a given forecast prevailed).

forecasts respectively.[9]

It is found that when no news is used the model is more likely to learn a degenerate prediction (a constant) than when news is used as input. However, forecasts using news are for all nondegerate cases more volatile than those without. Since this behavior appears to be pseudo-deterministic, degenerate predictions were left in when calculating error metrics and performing the DM test.

6. CONCLUSIONS

In the present work it has been shown that BERT is able to perform the classification task with the highest accuracy out of all models, as well as yield the most semantically consistent labels on the previously unseen corpus C_B. Furthermore, it has been shown that utilizing features generated from news for forecasting stock prices for the given sample of companies over the selected interval yields significantly better predictions than not using news for ARIMAX. The LSTM network however seems to predict prices with much higher accuracy in all nondegenerate cases, with news features from C_B yielding the set of predictions with lowest median error across all measures, indirectly pointing to BERT's efficacy in relabelling. In terms of news forecasts with this model however, it is with C_A's data that news series forecasts are on average more reliable.

[9]While the test does assume the loss differential to be covariance stationary, which isn't often the case for ARIMAX, plotting all three sets forecasts for this model class seems to empirically validate the vedrict of the test statistic (in cases when $DM \sim \mathcal{N}(0,1) \gtrless \pm 1.96$).

7. ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 675044.

The first author is grateful to Dunja Mladenic and the E3 department for the opportunity for a summer at the Jožef Stefan Institute.

8. REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2018.

[2] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *EACL*, 2016.

[3] Y. Kim. Convolutional neural networks for sentence classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 08 2014.

[4] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, 2015.

[5] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *ArXiv*, abs/1405.4053, 2014.

[6] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. *ArXiv*, abs/1703.03130, 2017.

[7] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.