Realised Volatility Forecasting: Machine Learning via Financial Word Embedding

Eghbal Rahimikia, Stefan Zohren, and Ser-Huang Poon*

December 12, 2022

Abstract

We develop *FinText*, a financial word embedding covering around 15 years of news from 2000 to 2015 with added emphasis on financial news. We show that the best-performing model reaches substantially higher accuracies compared with general-purpose word embeddings in our gold-standard financial benchmark. In contrast to well-known econometric models, incorporating this word embedding in a simple machine learning model improves volatility forecasting performance in a sample period from 27 July 2007 to 27 January 2022. We extend our analysis by measuring the importance of n-grams, discovering the primary volatility movers in stock-related and general news stories, and presenting the primarily responsible classes of n-grams for realised volatility forecasting.

Keywords: Realised Volatility Forecasting, Machine Learning, Natural Language Processing, Word Embedding, Explainable AI, Big Data.
JEL: C22, C45, C51, C53, C55, C58

*Eghbal Rahimikia (corresponding author) (eghbal.rahimikia@manchester.ac.uk) is at the University of Manchester, Alliance Manchester Business School, Stefan Zohren (stefan.zohren@eng.ox.ac.uk) is at the Oxford-Man Institute of Quantitative Finance, University of Oxford and Ser-Huang Poon (ser-huang.poon@manchester.ac.uk) is at the University of Manchester, Alliance Manchester Business School. We are grateful to the participants and discussants of the 2021 FMA Conference on Derivatives and Volatility, 2022 British Accounting and Finance Association (BAFA) Annual Conference, Economics of Financial Technology Conference, 12th Financial Markets and Corporate Governance Conference, 2022 FMA European Conference, 14th Annual SoFiE Conference, Advances in Data Science Conference, Greater China Area Finance Conference, 2022 FMA Annual Meeting, 3rd Frontiers of Factor Investing Conference, London-Oxford-Warwick Mathematical Finance Workshop, and 2022 Cardiff Fintech Conference. All models are run on the computational shared facility of the University of Manchester. We must express our sincere appreciation to the IT services of the University of Manchester for their constant and continued support and for providing the computational infrastructures for this study. Last but not least, Our sincere thanks are due to the accounting and finance division at Alliance Manchester Business School for their financial support.

1 Introduction

Many studies have attributed news as a major contributor to volatility (Engle and Ng, 1993; Engle and Martins, 2020; Conrad and Engle, 2021). In recent years, researchers have shown an increased interest in using natural language processing (NLP) and machine learning (ML) methods to extract relevant information from textual data such as news. So far, however, despite dramatic successes in other fields, these new techniques have attracted very little attention from the finance and economic scholarly communities (Gentzkow et al., 2019).

This paper explores the use of news in realised volatility (RV) forecasting using a state-of-the-art word embedding approach. Instead of using pre-trained Google's and Facebook's word embeddings, we develop *FinText*[1], a purpose-built financial word embedding for financial textual analysis, based on the Dow Jones Newswires Text News Feed, which covers different news services around the world with added emphasis on financial news. Unlike the Loughran-McDonald (LM) dictionary approach (Loughran and McDonald, 2011) that solely relies on predefined sets of words for extracting sentiment, our approach extracts a substantial amount of information from a big financial textual dataset without using any manually predefined resources and model assumptions. Moreover, to analyse the effect of different n-grams on volatility forecasts and discover the volatility movers, we use Explainable AI (XAI) to make the forecasting performance evaluation more transparent and understandable.

Most RV forecasting studies use historical RV as the primary source to predict the next-day volatility using a linear model. Heterogeneous autoregressive (HAR) models, which are simple yet effective linear models for RV forecasting, were first introduced by Corsi (2009). The further development of the HAR-family of models continued with HAR-J (HAR with jumps) and CHAR (continuous HAR) of Corsi and Reno (2009), SHAR (semivariance-HAR) of Patton and Sheppard (2015), and HARQ model of Bollerslev et al. (2016). The study of Rahimikia and Poon (2020a) provides a valuable comparison of the HAR-family of models and shows that the CHAR model is the best-performing model among all. It also supplements the CHAR model with limit order book (LOB) data and sentiment variables extracted from financial news. The resulting CHARx model shows that news and LOB data provide statistically significant improvement in RV forecasts. Although the work above has successfully demonstrated that adding more information from news data improves the RV forecasting performance, it has certain limitations in terms of just using sentiment and linear regression for forecasting.

During the last decade, there has been a growing number of publications focusing on the theory and application of ML in financial studies. Recent evidence suggests that this group of models can outperform traditional financial models in portfolio optimisation (Ban et al., 2018), LOB models for short-term price predictions (Zhang et al., 2018; Sirignano and Cont, 2019; Zhang and Zohren, 2021), momentum strategies (Lim et al., 2019; Poh et al., 2021; Wood et al., 2021), estimation of stochastic discount factor (Chen et al., 2020), equity premium prediction using newspaper articles (Adämmer and Schüssler, 2020), measuring asset risk premiums (Gu et al., 2020), image processing for return prediction (Jiang et al., 2020), classifying venture capitalists (Bubna et al., 2020), designing trading strategies (Zhang et al., 2020), latent factor modelling (Gu et al., 2021), hedge fund return prediction (Wu et al., 2020), bond return prediction (Bianchi et al., 2021), and return forecasting using news

[1] *FinText* word embeddings are available for download from FinText.ai.

photos (Obaid and Pukthuanthong, 2021), to name a few. In the context of RV forecasting, Rahimikia and Poon (2020b) comprehensively examined the performance of ML models using big datasets such as LOB and news stories. They show that LOB data has strong forecasting power compared to the HAR-family of models, and adding news sentiment variables to the dataset only improves the forecasting power marginally. However, this study remains narrow in focus dealing only with sentiment extracted from the LM dictionary. The two principal limitations of the LM dictionary are that it does not consider language complexities and it is developed based on only financial statements. Except few studies focusing on statistical and ML models for sentiment extraction to predict asset returns (Ke et al., 2019), textual factor analysis (Cong et al., 2019), topic modelling (Bybee et al., 2020), designing a sentiment-scoring model to capture the sentiment in economic news articles (Shapiro et al., 2020), and developing word embedding for analysing the role of corporate culture during COVID-19 pandemic (Li et al., 2020); so far, there has been little focus on more advanced NLP models for financial forecasting. Much of the current trend on NLP focuses on word embedding (Mikolov et al., 2013), a more sophisticated word representation that paved the way for modern textual-oriented ML models.

As a major review of textual analysis in accounting and finance, Loughran and McDonald (2016) warns that these more complex ML models potentially add more noise than signal. We believe the signal-to-noise of ML models is reasonable if they can simultaneously improve performance and generate readable and acceptable knowledge in finance by XAI or other similar approaches. Therefore, we set out to investigate the usefulness of a more advanced NLP structure for RV forecasting. Part of the aim of this study is to develop a financial word embedding, named *FinText*, and compare it with publicly available general word embeddings by well-known general-purpose benchmarks and, more specifically, our introduced gold-standard financial benchmark. Another objective of this research is to determine whether a word embedding inside a simple ML structure, solely trained on news data, is powerful enough for RV forecasting. Finally, as another important objective, this study shines new light on these debates using XAI methods to interrogate the models.

We show that our financial word embedding is more sensitive to financial context compared with general word embeddings. The proposed gold-standard financial benchmark containing 2660 unique analogies demonstrates that the best-performing *FinText* model reaches around 14 and 55 times better accuracies than Google's and Facebook's word embeddings, respectively. Using 23 NASDAQ stocks from 27 July 2007 to 27 January 2022 and by using just previous day stock-related news headlines, our proposed word embedding performs better, especially for high volatility days, although it is developed by a substantially smaller textual corpus. From the temporal forecasting performance results, we argue that because word embeddings are trained by a specific corpus covering a limited time horizon, there is a decrease in performance over time, especially during the COVID-19 outbreak, when new terminologies appear in the news.

We also extend this framework by replacing stock-related news with entirely district general news covering major economic, financial, political, and geopolitical news stories. The obtained results show that not only the forecasting power switches from high volatility to normal volatility days but also Google's and Facebook's word embeddings show a better forecasting performance than *FinText*. This finding is partly explained by the size and mixture of the corpora because both contain substantially

more extensive and diverse textual data inside. The findings from ensemble models show the importance of the information content of both financial numbers and textual news simultaneously for volatility forecasting. Thus, textual news stories can not be considered a replacement for numerical financial news, but they can potentially improve the performance of established econometric models in RV forecasting. Last but not least, we measure the importance of n-grams and extend this approach to discover the primary volatility movers in stock-related and general news headlines using XAI methods. Regarding stock-related news, n-gram classes like analyst opinions, company events, numbers, and announcements are identified as volatility movers. For general news, this changes to n-gram classes like person names, places, and legal entities. Discovering such clear and in-depth information about the groups of n-grams responsible for the behaviour of an asset pricing model is not feasible in the classical dictionary-based approaches.

This paper is structured into eight sections. Section 2 deals with the theory of word embedding, Word2Vec, and FastText algorithms. Section 3 introduces our word embedding called *FinText*. News preprocessing steps are covered in Subsection 3.1 and evaluation and representation of this proposed word embedding in Subsection 3.2. Section 4 gives a brief review of RV forecasting followed by related models in Subsection 4.1. The proposed simple textual-based model for RV forecasting is introduced in Subsection 4.2, and XAI methods in Subsection 4.3. Next, Section 5 presents the findings of the research, focusing on the stock-related news in Subsection 5.1, general hot news in Subsection 5.2, and ensemble models in Subsection 5.3. Section 6 presents the XAI results by comparing the XAI and LM dictionary in Subsection 6.1 and discovering volatility movers in Subsection 6.2. Section 7 looks at robustness checks and finally, Section 8 concludes with a discussion.

2 Word Embedding

Word embedding is one of the most important recent developments in NLP, where a word representation is in the form of a real-valued vector such that words that are closer in the vector space are expected to be similar in meaning. Before the development of word embeddings, each token in a one-hot encoding was defined by a binary vector of zeroes except the index for that token in the vocabulary list. The vectors of any two tokens are orthogonal to each other. Hence, word embedding is a better representation of semantics. Specifically, a word embedding (V) is a $M \times N$ matrix, where M is the dimension size, and N is the number of unique tokens in the vocabulary list W. Each token is represented by a vector of M values. In the literature, M is usually about 300. Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2017) are two of the most efficient algorithms for training word embeddings. Subsection 2.1 and Subsection 2.2 briefly review these two simple and efficient algorithms.

2.1 Word2Vec

Mikolov et al. (2013) proposed supervised learning models with Skip-gram and continuous bag-of-words (CBOW) log likelihoods for the Word2Vec algorithm as shown below:

$$L_{skip-gram} = \frac{1}{T}\sum_{t=1}^{T}\Big[\sum_{j=-k\backslash 0}^{k} log\ p(w_{t+j}|w_t)\Big], \tag{1a}$$

$$L_{CBOW} = \frac{1}{T}\sum_{t=1}^{T}\Big[\sum_{j=-k\backslash 0}^{k} log\ p(w_t|w_{t+j})\Big], \tag{1b}$$

where T is the total number of tokens in the sequence $X = \{t_1, t_2, ..., t_T\}$, k is the window size around the chosen token w_t, and $p(w_{t+j}|w_t)$ and $p(w_t|w_{t+j})$ are the probability of correct predictions of Skip-gram and CBOW models, respectively. In the Skip-gram, the input (middle) token is used to predict the context (surrounding tokens), whereas the context (surrounding) tokens are used to predict the middle token in CBOW. It is generally agreed that the faster CBOW is suited for training larger datasets, while the Skip-gram is more efficient for training smaller datasets. Both models aim to maximise the aggregate predictive probability in Equation (1a) and Equation (1b) based on a simple neural network architecture.

For both Skip-gram and CBOW, the softmax operation for calculating the conditional probability is defined as follows:

$$p_{skip-gram}(w_c|w_t) = \frac{exp(u_{w_c}^T u_{w_t})}{\sum_{l=1}^{N} exp(u_l^T u_{w_t})}, \tag{2a}$$

$$p_{CBOW}(w_t|w_c) = \frac{exp(u_{w_t}^T \bar{u}_{w_c})}{\sum_{l=1}^{N} exp(u_l^T \bar{u}_{w_c})}, \tag{2b}$$

where w_c and w_t are context and target tokens, u_{w_c} and u_{w_t} are the trainable vector of context token w_c and target token w_c, and N is the number of tokens in the vocabulary. For the CBOW model in Equation (2b), as there is more than one context token, the average of the context token vectors, $\bar{u}_{w_c}$, is used. Both models are trained using stochastic gradient descent.

When there are a large number of tokens in the dictionary (N), Equation (2a) and Equation (2b) are computationally expensive. In this case, hierarchical softmax (Morin and Bengio, 2005) can be used instead. Another alternative is the negative sampling method with a binary logistic regression (Mikolov et al., 2013). In this method, the training set is the pair of target and context tokens, and K tokens are randomly chosen from a specific distribution. The output is one for the first token pair and zeroes for all other pairs. Mikolov et al. (2013) found that K ranges from 5 to 20 for large training sets and ranges from 2 to 5 for small training sets.

2.2 FastText

FastText is an extension of Word2Vec. For example, take 'profit' as a token and set n-gram equal to 3 (i.e. $n = 3$); the corresponding token vector is defined as <pr, pro, rof, ofi, fit, it>, where < and >

indicate the start and the end of the token vector. The original token <profit> is also added to this list. More formally, for token w , $u_w = \sum_{i=0}^{n} u_i$ where u_w is the vector of token w, n is the number of n-gram of this token, and u_i is the vector of each sub-token. In this enhanced representation, each token consists of a bag of n-gram characters. This algorithm is computationally intensive, but compared with the Word2Vec algorithm, it is more powerful for learning rare and out-of-vocabulary (OOV) tokens and morphologically rich languages (MRL) (Bojanowski et al., 2017).

3 Financial Word Embedding: *FinText*

At the time of writing, several well-known pre-trained word embeddings are available. These include Mikolov et al. (2018) three-million-unique-tokens Word2Vec algorithm trained using the Google news dataset with about 100 billion words, and Joulin et al. (2016) one-million-unique-tokens FastText algorithm developed by Facebook and trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset. It is arguable whether these general word embeddings are accurate for finance. Some words will have a very different meaning when used in a specific financial context, e.g. apple as fruit and Apple as the technology company. To address this concern, we train a new word embedding, called *FinText*, using the Dow Jones Newswires Text News Feed database from 1 January 2000 to 14 September 2015. While specialising in financial news, this big textual dataset is among the best textual databases covering finance, business, and political news services worldwide. Subsection 3.1 describes the database, the preprocessing steps and the properties of the word embedding developed from it, and Subsection 3.2 compares our *FinText* with those from Google and Facebook mentioned above.

3.1 News Data, Preprocessing Steps and Model Properties

This study uses all types of news (viz. financial, political, weather, etc.) from Dow Jones Newswires Text News Feed from January 1, 2000, to September 14, 2015. All duplicate news stories and stories without headline and body are removed. Extensive text preprocessing of news stories is required to eliminate redundant characters, sentences, and structures. Table A1 in the Appendix presents a brief review of the cleaning rules applied. Each rule is defined by a regular expression and may contain different variations. For brevity, only one variation is shown in this table. The text cleaning procedures fall into five main categories: 1) Primary, 2) Begins with, 3) Ends with, 4) General, and 5) Final checks. 'Primary' extracts the body of news from the extensible markup language (XML), removing XML-encoding characters (XMLENCOD), converting XML to text (parsing), converting uppercase to lowercase letters, and removing tables. 'Begins with' and 'Ends with' remove, respectively, parts begin and end with the specified structures. 'General' caters for patterns that may appear in any part of the news stories. Finally, 'Final checks' removes links, emails, phone numbers, short news (lower than 25 characters), and the leading and trailing space(s). These five sets of rules are applied to news headlines and bodies separately. Due to the importance of numbers in accounting and finance, all numbers are kept in our textual database. This plays a key role in keeping the sentences intact when incorporating this word embedding in more complex models.[1] Figure 1 shows the total number

[1] As an example, removing numbers changes 'Over 540,000 apps wiped from Apple App Store in Q3 reaching lowest number in 7 years.' headline to 'Over apps wiped from Apple App Store in Q reaching lowest number in years.' This

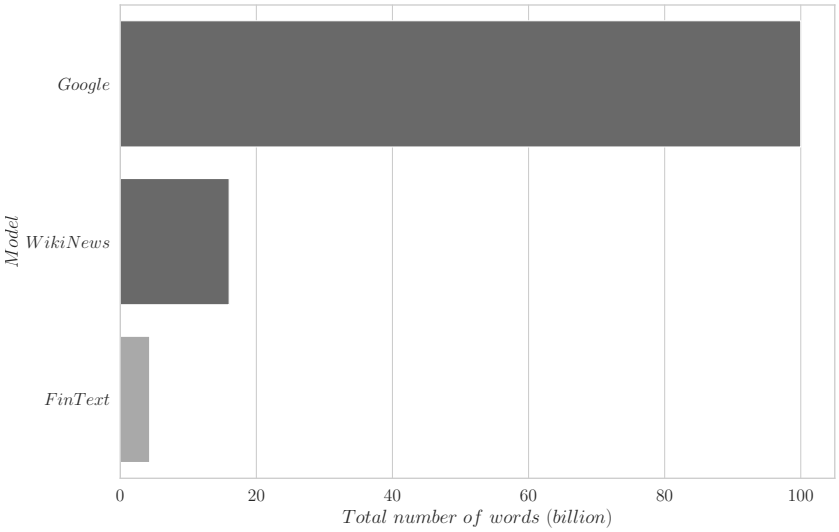


Figure 1: Number of words in corpus

Notes: This figure shows the total number of words (billion) in the corpus used for training Google Word2Vec Mikolov et al. (2018), Facebook WikiNews Joulin et al. (2016) and *FinText* word embeddings. Google Word2Vec, Facebook WikiNews, and *FinText* corpora contain 100 billion, 16 billion, and 4.32 billion words, respectively.

of words (billion) in the corpus used for training Google Word2Vec Mikolov et al. (2018), Facebook WikiNews Joulin et al. (2016) and *FinText* word embeddings. Google Word2Vec, Facebook WikiNews, and *FinText* corpora contain 100 billion, 16 billion, and 4.32 billion words, respectively. Quite clearly, *FinText* used a substantially smaller corpus compared with the other major well-known pre-trained word embeddings. Regarding Google Word2Vec, the data time span is not clearly defined, but 2013 is the approximate end year. Also, the data timespan for Facebook WikiNews is from 2007 to 2017.

After cleaning the dataset, tokenisation breaks the headlines and news bodies into sentences and words. Common bigram (two-word) phrases are detected and replaced with their bigram form. All tokens with less than five times of occurrences are ignored, the proposed bigram scoring function in Mikolov et al. (2013) is used with ten as the threshold value, and the maximum vocabulary size is set to 30 million to keep memory usage under control. Finally, the '_' character is used for glueing pairs of tokens together. For example, if 'financial' and 'statement' appears consecutively exceeding a threshold number, they are replaced by 'financial_statement' as a new token. Altogether, *FinText* consists of 2,733,035 unique tokens. Following the preprocessing steps, Word2Vec and FastText algorithms are applied with window size, minimum count[1], negative sampling[2], and the number of iterations[3] all set equal to 5. The initial learning rate (alpha) is 0.025, the minimum learning rate is 0.0001, and the exponent for negative sampling distribution is 0.75. Also, the dimension of word embeddings is 300. All these parameter values are the proposed ones by their developers.

substantial change in meaning is harmful when the model is not just focusing on single tokens like the dictionary approach but considering the order of tokens as an essential source of information.

[1]The model ignores tokens with lower frequency than this value.

[2]Number of added noisy tokens matched with each chosen token.

[3]Number of epochs over the corpus.

Table 1: Word embedding comparison (Google analogy)

Section	Word2Vec[a]			FastText		
	FinText[b] (CBOW)[c]	**FinText** (skip-gram)	**Google** (skip-gram)	**WikiNews** (skip-gram)	**FinText** (skip-gram)	**FinText** (CBOW)
capital-common-countries	77.27	85.50	83.60	**100**	85.93	47.40
capital-world	63.60	75.87	82.72	**98.78**	71.06	35.79
currency	22.49	36.69	**39.84**	25.00	32.54	10.65
city-in-state	19.93	60.48	74.64	**81.41**	58.20	15.83
family	63.46	70.51	90.06	**98.69**	58.97	59.62
gram1-adjective-to-adverb	27.47	33.00	32.27	70.46	50.59	**79.45**
gram2-opposite	33.33	32.50	50.53	**73.91**	50.83	71.67
gram3-comparative	77.65	75.04	91.89	**97.15**	77.06	87.39
gram4-superlative	61.67	55.00	88.03	**98.68**	62.14	90.71
gram5-present-participle	62.30	61.24	79.77	**97.53**	70.63	76.06
gram6-nationality-adjective	88.11	93.23	97.07	**99.12**	94.05	79.05
gram7-past-tense	42.02	39.92	66.53	**87.25**	37.98	31.09
gram8-plural	59.23	62.46	85.58	**98.69**	70.92	79.54
gram9-plural-verbs	53.26	54.53	68.95	**97.38**	61.59	79.17
overall	53.65	62.86	77.08	**91.44**	65.00	55.74

[a] For learning word embeddings from textual datasets, **Word2Vec** is developed by Mikolov et al. (2013) and **FastText**, as an extension to Word2Vec algorithm, is developed by Bojanowski et al. (2017). [b] Developed word embedding on Dow Jones Newswires Text News Feed database (**FinText**); Publicly available word embedding trained on a part of Google news dataset (**Google**); Publicly available word embedding trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (Mikolov et al., 2018) (**WikiNews**). [c] The continuous bag of words (CBOW) and Skip-gram are the proposed supervised learning models for learning distributed representations of tokens in Mikolov et al. (2013).

3.2 Evaluation & Representation

The results for Skip-gram and CBOW models are reported for Word2Vec and FastText algorithms. These results are compared with pre-trained word embeddings from Google[1] (Word2Vec algorithm) and Facebook[2] (FastText algorithm).

3.2.1 General-Purpose Benchmarks

Table 1 compares all choices of word embedding based on the Google analogy benchmark. *FinText* is our financial word embedding, 'Google' is the word embedding trained on Google news dataset, and 'WikiNews' is Facebook's word embedding trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset. Each section in the Google analogy benchmark contains a group of analogies. For example, under the 'capital-common-countries' section, the word embedding is challenged with the questions like 'London to England is like Paris to?'. The accuracy of each word embedding is reported for each section and all sections (overall).

From Table 1, it is apparent that, except for 'currency' and 'gram1-adjective-to-adverb', Facebook's WikiNews has the highest predictive accuracy. The overall evaluation scores confirm this finding. The individual and overall scores for *FinText* suggest Skip-gram works better than CBOW. For this general-purpose benchmark, our financial word embedding is outperformed by Google under Word2Vec and outperformed by WikiNews under FastText. Table 3 presents the predictive accuracy

[1] Word2Vec algorithms are downloadable from https://code.google.com/archive/p/word2vec/

[2] FastText algorithms are downloadable from https://fasttext.cc/

Table 2: Word embedding comparison (Gold-standard collections)

Benchmark	Word2Vec[a]			FastText		
	FinText[b] (CBOW)[c]	**FinText** (skip-gram)	**Google** (skip-gram)	**WikiNews** (skip-gram)	**FinText** (skip-gram)	**FinText** (CBOW)
WordSim-353[d] (relatedness)	0.3821	0.4993	**0.6096**	0.6018	0.4425	0.1677
WordSim-353 (similarity)	0.6126	0.6436	**0.7407**	0.6713	0.6393	0.4722
Simlex	0.2657	0.2650	0.3638	**0.3985**	0.2772	0.2574

[a] For learning word embeddings from textual datasets, **Word2Vec** is developed by Mikolov et al. (2013) and **FastText**, as an extension to Word2Vec algorithm, is developed by Bojanowski et al. (2017).
[b] Developed word embedding on Dow Jones Newswires Text News Feed database (**FinText**); Publicly available word embedding trained on a part of Google news dataset (**Google**); Publicly available word embedding trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (Mikolov et al., 2018) (**WikiNews**). [c] The continuous bag of words (CBOW) and Skip-gram are the proposed supervised learning models for learning distributed representations of tokens in Mikolov et al. (2013).
[d] WordSim-353 (Agirre et al., 2009) is a gold-standard collection for measuring word relatedness and similarity, and Simlex (Hill et al., 2015) is another gold-standard collection tending to focus on similarity rather than relatedness or association.

based on the gold-standard collections, viz. WordSim-353 (Agirre et al., 2009) for measuring word relatedness and similarity, and Simlex (Hill et al., 2015) that focuses on similarity. All collections contain human-assigned judgements about the relatedness and similarity of word pairs. Performance is measured by Spearman's rank correlation coefficient. It is apparent from this table that Google's Word2Vec outperformed under WordSim-353, WikiNews outperformed under FastText; both outperformed *FinText* in their respective categories. As in the previous table, for *FinText*, Skip-gram is marginally better than CBOW.

3.2.2 Financial-Purpose Tasks; *examples*

The general-purpose benchmark evidence reviewed in Subsubsection 3.2.1 seems to suggest that although our developed word embedding, *FinText*, cannot reach the general-purpose predictive accuracy of well-known pre-trained word embeddings, it still shows relatively fair performance in general tasks. In order to help familiarise readers with financial terminology and its critical importance in developing financial language models, a few financial examples are formulated here.

First, for each word embedding, principal component analysis (PCA) is applied to the 300-dimensional vectors. Figure 2 presents the 2D visualisation of word embeddings. The tokens are chosen from groups of technology companies ('microsoft', 'ibm', 'google', and 'adobe'), financial services and investment banks ('barclays', 'citi', 'ubs', and 'hsbc'), and retail businesses ('tesco' and 'walmart'). *Dimension* 1 (x-axis) and *Dimension* 2 (y-axis) show the first and second obtained dimensions. Word2Vec is shown in the top row, and FastText is shown in the bottom row. Figure 2 shows that only *FinText* clusters all groups correctly, and in line with the benchmarks reviewed in Subsubsection 3.2.1, Word2Vec produces generally better results than FastText.

Next, we challenged all word embeddings to produce the top three tokens that are most similar to

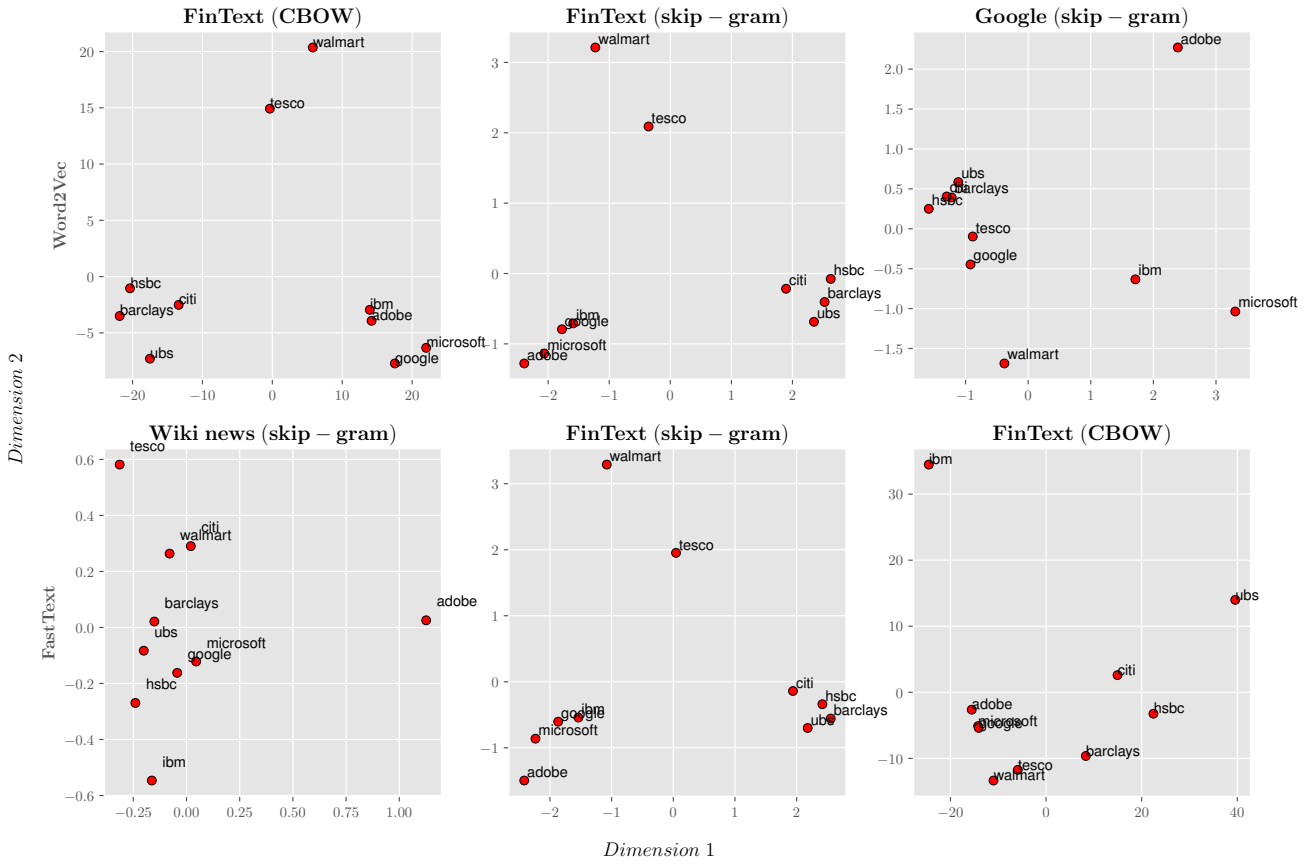


Figure 2: 2D visualisation of word embeddings

Notes: This figure shows the 2D visualisation of word embeddings. For each word embedding, principal component analysis (PCA) is applied to 300-dimensional vectors. The chosen tokens are 'microsoft', 'ibm', 'google', and 'adobe' (technology companies), 'barclays', 'citi', 'ubs', and 'hsbc' (financial services and investment banking companies), and 'tesco' and 'walmart' (retail companies). *Dimension* 1 (x-axis) and *Dimension* 2 (y-axis) show the first and second obtained dimensions. Word2Vec and FastText algorithms are shown in the first and second rows. *FinText* is the trained word embedding on Dow Jones Newswires Text News Feed database, Google is a publicly available developed word embedding trained on a part of Google news dataset, and WikiNews is another publicly available developed word embedding trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset. The continuous bag of words (CBOW) and Skip-gram are the proposed supervised learning models for learning distributed representations of tokens in Mikolov et al. (2013).

Table 3: Financial analogy examples

Analogy	Word embedding		
	Google	**WikiNews**	**FinText**[a]
debit:credit :: positive:X	positive	negative	negative
bullish:bearish :: rise:X	rises	rises	fall
apple:iphone :: microsoft:X	windows_xp	iphone	windows
us:uk :: djia:X	NONE[b]	NONE	ftse_100
microsoft:msft :: amazon:X	aapl	hmv	amzn
bid:ask :: buy:X	tell	ask-	sell
creditor:lend :: debtor:X	lends	lends	borrow
rent:short_term :: lease:X	NONE	NONE	long_term
growth_stock:overvalued :: value_stock:X	NONE	NONE	undervalued
us:uk :: nyse:X	nasdaq	hsbc	lse
call_option:put_option :: buy:X	NONE	NONE	sell

[a] *FinText* is the financial word embedding developed using the Dow Jones Newswires Text News Feed database, Word2Vec algorithm and Skip-gram model. [b] Not in the vocabulary list.

'morningstar'[1]. The results were as follows: This token is not among the training tokens of Google. WikiNews's answers are 'daystar', 'blazingstar', and 'evenin'. Answers from *FinText* (word2vec/skip-gram) are 'researcher_morningstar', 'tracker_morningstar', and 'lipper'[2]. When asked to find the unmatched token in a group of tokens such as ['usdgbp', 'euraud', 'usdcad'], a collection of exchange rates mnemonics, the results were as follows: Google and WikiNews could not find these tokens, while *FinText* (word2vec/skip-gram) produces the sensible answer, 'euraud'.

Word embeddings are expected to solve word analogies such as king:man :: woman:queen.[3] Table 3 lists some challenges we posed and the answers produced by the group of word embeddings considered here. Looking at Table 3, it is obvious that our financial word embedding is more sensitive to financial contexts and able to capture very subtle financial relationships. Although these examples have successfully demonstrated that *FinText* is substantially better in dealing with tasks based on financial jargon, these examples do not provide a gold-standard financial benchmark for a solid comparison of word embeddings.

3.2.3 Introducing Gold-Standard Financial Benchmark

Subsubsection 3.2.2 showed that *FinText* has superior performance in financial-purpose tasks. Here, by introducing the first gold-standard financial benchmark, we extend this to a financial language framework for effectively comparing word embeddings.[4] Following the Google analogy benchmark structure in Subsubsection 3.2.1, Table 4 consists of seven financial analogy sections developed by Bureau van Dijk's Orbis database. The first five sections cover publicly listed US companies, the sixth section mixes US and UK publicly listed companies, and the last section mixes US, UK, China, and Japan publicly listed companies. The chosen key elements from company information are as follows: 'Ticker' is the ticker identifier, 'name' is the full name of the company, 'city' is the headquarters

[1] Morningstar is an American financial services firm was founded in 1984.
[2] Thomson Reuters Lipper is an American financial services firm was founded in 1973.
[3] ':' means 'is to' and '::' means 'as'.
[4] Gold-standard financial benchmark is available for download from FinText.ai.

Table 4: Gold-standard financial benchmark

Section	Word2Vec[a]			FastText		
	FinText[b] (CBOW)[c]	**FinText** (skip-gram)	**Google** (skip-gram)	**WikiNews** (skip-gram)	**FinText** (skip-gram)	**FinText** (CBOW)
Ticker to City (US)	14.74	**23.68**	0.26	0.00	15.00	1.05
Name to Ticker (US)	38.55	**43.29**	0.13	0.00	34.61	19.08
Name to Incorporation year (US)	28.16	**30.79**	0.09	0.88	26.40	14.65
Name to Exchange (US)	**28.55**	24.47	4.08	0.72	19.87	10.99
Name to State (US)	**24.16**	20.26	3.26	0.63	16.16	8.79
Name to Country (US & UK)	**21.93**	18.25	2.72	0.53	13.73	7.32
Name to Country (US, UK, China, & Japan)	**19.59**	16.28	2.33	0.45	12.52	6.28
Overall	25.10	**25.29**	1.84	0.46	19.75	9.74

[a] For learning word embeddings from textual datasets, **Word2Vec** is developed by Mikolov et al. (2013) and **FastText**, as an extension to Word2Vec algorithm, is developed by Bojanowski et al. (2017).
[b] Developed word embedding on Dow Jones Newswires Text News Feed database (**FinText**); Publicly available word embedding trained on a part of Google news dataset with about 100 billion words (**Google**); Publicly available word embedding trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (Mikolov et al., 2018) (**WikiNews**). [c] The continuous bag of words (CBOW) and Skip-gram are the proposed supervised learning models for learning distributed representations of tokens in Mikolov et al. (2013).

location, 'exchange' is the stock exchange the company is traded on, 'country' is the country the headquarters is located in, 'state' (for US companies) is the state the headquarters is located in, and finally 'incorporation year' is the incorporation year of the company. For the first five sections, the top 20 companies are chosen regarding the company size in 'very large companies' group. Following the previous criterion, for the sixth section (US & UK mixture) and seventh section (US, UK, China, & Japan mixture), 10 and 5 companies are chosen respectively from each country and mixed to make the final list. The permutation of chosen companies in each section generates 380 unique analogies for each group and 2660 analogies in total. An answer is correct when it is among the top five answers.[1] The accuracy of each word embedding is reported for each section and all sections (overall).

There are some important considerations we carefully monitored in the development of this gold-standard financial benchmark. First, these key elements from company information are chosen because they don't change over time. Second, for key elements in each section, just unigrams are kept, and the rest is removed from the list. As discussed in Subsection 3.1, although *FinText* covers bigrams, Google Word2Vec and Facebook WikiNews are not generally appropriate for n-grams with a size larger than one. Finally, regarding market capitalisation, the UK, China, and Japan are among the countries with the largest stock markets in the world. Therefore, we expanded our benchmark using information from these countries for the last two sections.

Looking at Table 4, it is undeniable that *FinText* has substantially higher performance compared with all other well-known word embeddings. Facebook WikiNews accuracy is lower than 0.5% for all sections, with an overall accuracy of 0.46%. For Google Word2Vec, the overall accuracy is 1.84%. Among different variations of *FinText* word embedding, the Word2Vec algorithm and Skip-gram model show slightly better performance than the CBOW model. Overall, the best performing *FinText* model shows around 14 and 55 times better accuracies compared with Google Word2Vec and Facebook WikiNews, respectively. This benchmark shows that even though these two well-known word embeddings are developed by incorporating a substantially larger corpus, their performance in financial

[1] We found five as a fair value for this benchmark. Smaller values make the benchmark more strict, and larger values make it more lenient.

tasks is poor. Even though reaching this superior performance in the financial benchmark is an important achievement, in Section 4, we will extend this by introducing a novel structure for using word embedding in an empirical asset pricing framework.

4 News-based model for Realised Volatility Forecasting

The previous section has illustrated that the *FinText* word embeddings developed in this paper are more sensitive to financial relationships and reach a great accuracy in our proposed gold-standard financial benchmark. Here, we aim to use these embeddings in the context of volatility forecasting by introducing an empirical asset pricing framework. Engle and Ng (1993) and Engle and Martins (2020) already showed that news is a potential contributor to volatility. Therefore, we use word embeddings as a part of a simple ML model to see if word embeddings are useful in forecasting realised volatility or not. Among various possible frameworks, we found this framework efficient, fast, and transparent for empirical asset pricing. However, researchers can extend this by introducing new frameworks in future. In particular, Subsection 4.1 gives a brief review of RV forecasting, Subsection 4.2 presents our model, and Subsection 4.3 introduces XAI.

4.1 Realised Volatility Forecasting

Assume an asset price P_t follows the stochastic process below:

$$dlog(P_t) = \mu_t dt + \sigma_t dw_t, \tag{3}$$

where μ_t is the drift, w_t is the standard Brownian motion, and σ_t is the volatility process (càdlàg function). RV, defined below, is used as a proxy for the unobserved integrated variance, $IV_t = \int_{t-1}^{t} \sigma_s^2 ds$:

$$RV_t \equiv \sum_{i=1}^{M} r_{t,i}^2, \tag{4}$$

where $M = \frac{1}{\delta}$ is the sampling frequency and $r_{t,i} \equiv log(P_{t-1+i\delta}) - log(P_{t-1+(i-1)\delta})$.

To date, the HAR-family of models is the most popular group of econometric models for forecasting RV. All HAR models follow the general specification below:

$$RV_{t+1} = f(\overline{RV}_{t-i}, J_t, \overline{BPV}_{t-i}, RV_t^{+/-}, \overline{RQ}_{t-i}), \tag{5}$$

where RV_{t+1} is the forecasted RV, $\overline{RV}_{t-i}$ is the average RV of the last i days, J_t is the jump component[1], $\overline{BPV}_{t-i}$ of day t is the average Bi-Power Variation (BPV) of the last i days, $RV_t^{+/-}$ is the

[1] $J_t = Max[RV_t - BPV_t, 0]$ is the jump at time t, where $BPV_t = \frac{1}{\mu_1^2} \sum_{i=1}^{M-1} |r_{t,i}||r_{t,i+1}|$. M is the maximum value of sampling frequency, $r_{t,i}$ is the return at day t and sampling frequency i, and $\mu_1 = \sqrt{2/\pi}$ (Corsi and Reno, 2009).

Table 5: RV descriptive statistics (from 27 July 2007 to 27 January 2022)

Ticker[a]	Min	Max	1st quantile	Median	3rd quantile	Mean	STD	Kurtosis	Skewness
AAPL	0.102	229.420	0.899	1.733	3.680	4.623	12.596	111.012	9.124
MSFT	0.067	216.181	0.829	1.449	2.814	3.237	8.125	194.004	11.275
INTC	0.030	318.697	1.103	1.873	3.577	4.299	11.628	294.963	13.982
CMCSA	0.004	237.387	0.910	1.632	3.320	3.821	9.697	192.169	11.462
QCOM	0.122	373.543	1.024	1.975	4.129	5.073	15.380	200.609	12.100
CSCO	0.047	343.946	0.886	1.561	3.028	4.115	13.160	212.453	12.258
EBAY	0.205	252.608	1.319	2.271	4.356	5.082	12.592	142.684	10.009
GILD	0.064	259.489	1.167	1.892	3.379	4.304	12.930	182.820	12.063
TXN	0.177	287.897	1.047	1.905	3.748	4.014	9.820	311.666	14.242
AMZN	0.065	547.030	1.305	2.336	4.808	6.200	19.359	242.205	12.735
SBUX	0.052	265.094	0.864	1.594	3.423	4.201	11.237	161.435	10.626
NVDA	0.159	1104.351	2.282	4.358	9.084	9.756	30.117	586.612	20.058
MU	0.292	484.388	3.570	6.246	11.912	12.818	25.734	89.141	7.960
AMAT	0.292	531.579	1.783	3.028	5.712	6.005	14.632	532.194	18.338
NTAP	0.119	462.821	1.503	2.587	5.154	6.289	18.008	201.510	11.934
ADBE	0.119	569.720	1.099	2.020	3.908	4.947	15.003	588.095	18.867
XLNX	0.229	265.374	1.296	2.363	4.787	5.005	11.941	194.718	11.764
AMGN	0.032	214.156	0.969	1.593	2.872	3.398	9.612	183.759	11.898
VOD	0.055	219.033	0.687	1.342	3.137	3.933	10.869	122.252	9.601
CTSH	0.189	485.894	0.984	1.764	4.161	5.288	15.757	325.214	14.287
KLAC	0.154	499.808	1.456	2.710	5.416	5.919	16.878	354.626	16.033
PCAR	0.039	389.930	1.157	2.162	4.633	5.125	12.108	313.338	13.010
ADSK	0.268	693.772	1.644	2.765	5.167	6.644	22.377	388.131	16.554

[a] Tickers are ranked according to their liquidity (high to low).

positive/negative intraday return[1], $\overline{RQ}_{t-i}$ is the average realised quarticity[2] of the last i days, and f is a linear regression. Focusing on a long out-of-sample time horizon, Rahimikia and Poon (2020a) found the CHAR model to be the best performing HAR model among others. In Equation (5), the variable i for the BPV term in the CHAR model are the previous day ($i = 1$), the average of last week ($i = 7$), and the average of last month ($i = 21$) (Corsi and Reno, 2009).

In this study, the training period is from 27 July 2007 to 11 September 2015 (2046 days), and the out-of-sample period is from 14 September 2015 to 27 January 2022 (1604 days). RV is calculated during the NASDAQ market trading hours (from 9:30 AM to 4:00 PM Eastern Time). As a forecasting procedure, the rolling window method is applied in this study. Also, the LOB data from *LOBSTER* is used to calculate RV after applying the cleaning steps described in Rahimikia and Poon (2020a). The RV descriptive statistics of 23 NASDAQ stocks are presented in Table 5. These tickers are chosen based on their liquidity (high to low) and availability of data during the sample period.

4.2 Model Structure

Figure 3 is an abstract representation of the news-based model. $\{X_{(t,1)}, X_{(t,2)}, ..., X_{(t,k_t)}\}$ is the vector of k_t tokens from news headlines on day t. When k_t is less than 500, the padding process will fill the vector to 500 with 'NONE' so that the daily input to the neural network has the same length. The reason for using only the news headline and not the news body is that putting all the news

[1] $RV_t^+ = \sum_{i=1}^{M} r_{t,i(r>0)}^2$ and $RV_t^- = \sum_{i=1}^{M} r_{t,i(r<0)}^2$ (Patton and Sheppard, 2015).
[2] $RQ_t \equiv (\frac{M}{3}) \sum_{i=1}^{M} r_{t,i}^4$ (Bollerslev et al., 2016).

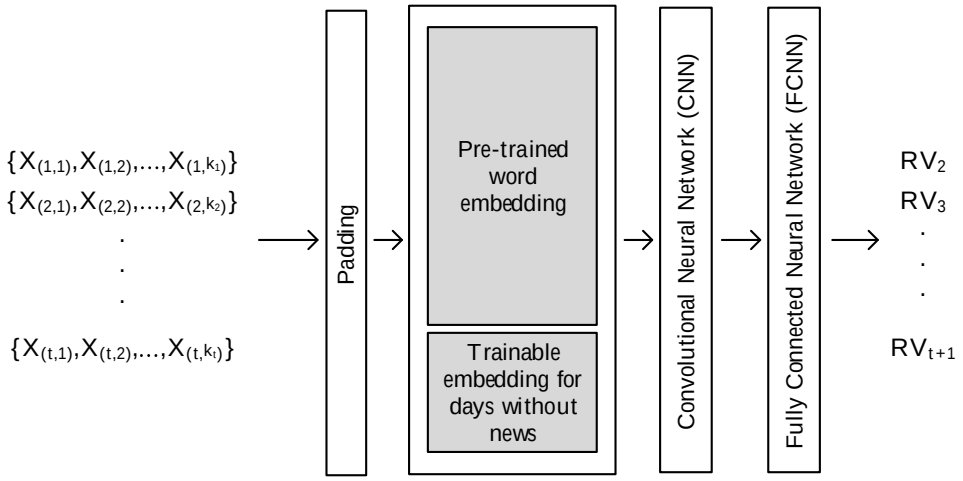


Figure 3: An abstract representation of model

Notes: $\{X_{(t,1)}, X_{(t,2)}, ..., X_{(t,k_t)}\}$ consists of news headlines of day t and $X_{(t,k_t)}$ is the k^{th} token of input t. Also, RV_{t+1} is the RV of day $t+1$ (next day RV). Padding with the maximum length of 500 is adopted to ensure that all inputs of the neural network have the same length. The word embedding block consists of two different word embeddings. To capture days without any news, a trainable word embedding is used.

bodies together makes the sequence of tokens extremely long, even for just one day. A very long token sequence stresses the computation and could result in over-fitting, especially when our training sample size is relatively small. Also, it is often felt that the news headline is the most important abstract of the news body.

As shown in Figure 3, the word embedding section is separated into days with news and days without news. For days with news, each token $X_{(t,k_t)}$ has a 1×300 word embedding vector from one of the six pre-trained word embeddings in Section 3. These vectors are fixed and made non-trainable to reduce the number of parameters to be trained. So this results in a 500×300 sentence matrix to be fed into a convolutional neural network (CNN) layer. On days when there is no news, the vector is initially filled with random numbers that can be trained by the neural network. After the CNN, we have a fully connected neural network (FCNN) that turns the signals into a single RV forecast, RV_{t+1}. This is a flexible framework in empirical asset pricing because it simply links textual data as input (independent variable) with forecasted RV as output (dependent variable). Also, it can mimic other asset pricing frameworks by switching RV to other variables of interest, it is free of any statistical assumption, and finally, it can model the high degree of nonlinearities. Finally, following Bollerslev et al. (2016), an 'insanity' filter is applied: For each rolling window, the minimum, maximum, and average of training RVs are calculated. Any RV forecast that is greater (smaller) than the maximum (minimum) value will be replaced by the rolling window average RV.

Figure 4 illustrates the structure of the CNN used in this study. Starting from the sentence matrix from Figure 3 for the news headline, 'apple looks to be further beefing up siri.', three filters of size 1, 2, and 3 are applied simultaneously with valid padding[1] and a stride size[2] of 1.[3] The reason for choosing

[1]In VALID padding (in contrast to SAME padding), the output feature map has a smaller vector size than the input word embedding vector size.

[2]Stride size defines the amount of filter movement over the word embedding vector.

[3]In the tuning process, we can choose how many sets of size 1, 2, and 3 filters to use. We have tested 25, 50, 75, and

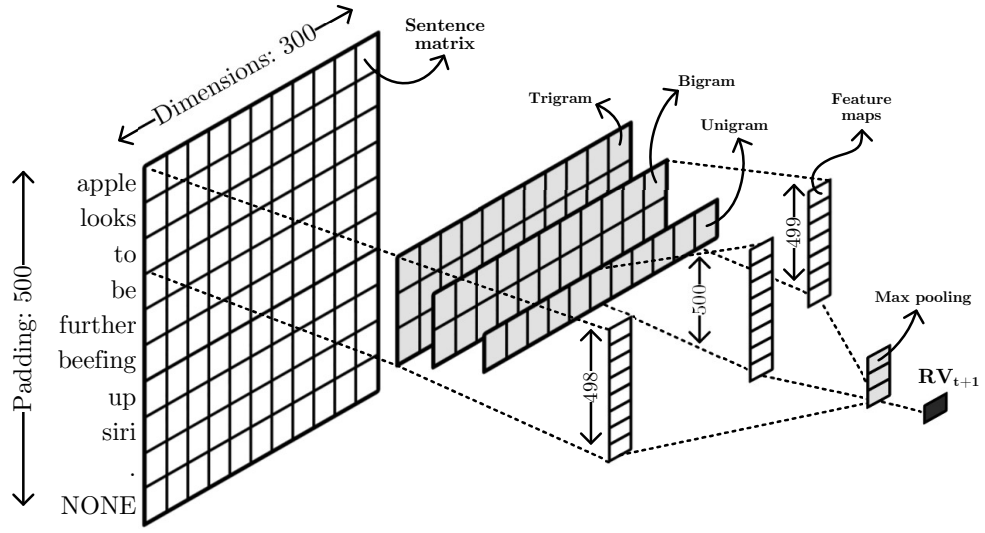


Figure 4: A detailed representation of model

Notes: The sentence matrix is a 500×300 matrix with a maximum length of padding of 500 and word embedding dimensions of 300. In this matrix, each token is defined by a vector of 300 values. This structure contains three filters of different sizes. The filters with the size of 1, 2, and 3 generate feature maps with the size of 500, 499, and 498, respectively. Global max pooling and a fully connected neural network (FCNN) are applied then as the next steps. The output of this network is the RV of the next day (RV_{t+1}).

this set of filter sizes is that 1, 2, and 3 are equivalent to unigram, bigram, and trigram.[1] The outputs are three 1-dimensional feature maps of sizes 498, 499, and 500. Specifically, following Kim (2014), let $X_i \in \mathbb{R}^M$ be the M-dimensional token vector corresponding to the i^{th} token in the news headline. We know from Figure 3 that $M = 300$, and news headlines with less than 500 tokens will be padded so that $n = 500$. Let $X_{i:i+j}$ refer to the concatenation of token vectors $X_i, X_{i+1}, \ldots, X_{i+j}$ as follows:

$$X_{i:i+j} = X_i \oplus X_{i+1} \oplus \cdots \oplus X_{i+j}, \tag{6}$$

where $\oplus$ is the concatenation operator. A convolution operation involves a filter $W \in \mathbb{R}^{hM}$, which is applied to a window size of h tokens to produce a new feature as follow:

$$C_i = f(W \cdot X_{i:i+h-1} + b), \tag{7}$$

where $b \in \mathbb{R}$ is a bias term, and f is a nonlinear function. This filter is applied to each possible window of tokens in the sentence $x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}$ to produce a feature map with $C \in \mathbb{R}^{n-h+1}$,

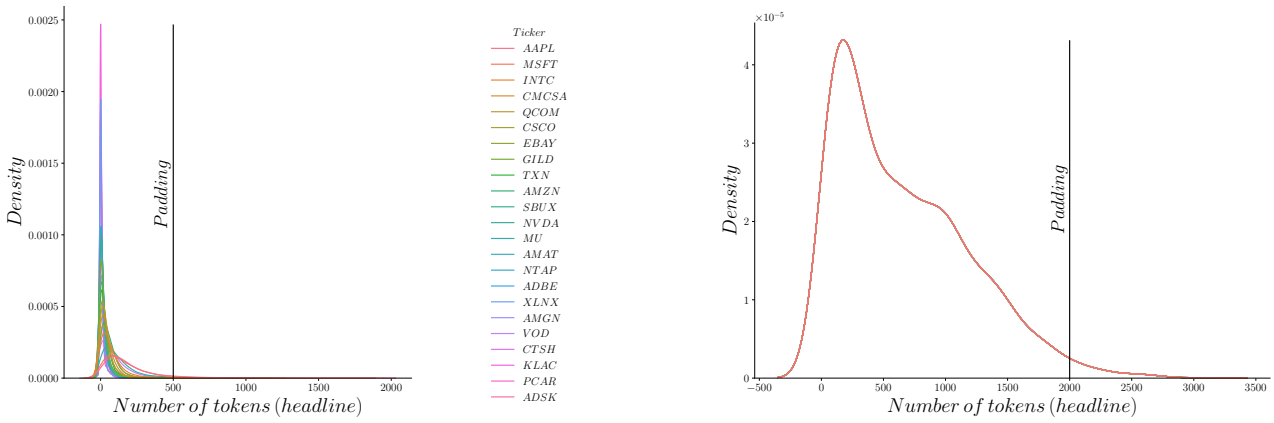
$$C = \{C_1, C_2, ..., C_{n-h+1}\}. \tag{8}$$

As the next step, global max-pooling ($\hat{C} = max\{C\}$) is applied. This step is used to ensure that the most important feature is chosen (Collobert et al., 2011). For converting the max-pooling layer to the RV of the next day (RV_{t+1}), an FCNN is used as the last layer. The activation function of both the CNN and the FCNN is a rectified linear unit (ReLU)[2], the optimisation algorithm is Adam

100 sets of filters in the empirical section.

[1] As discussed in Subsection 3.1, except for unigram, bigram is considered for developing *FinText*; therefore, theoretically, this model can reach 6-gram.

[2] Using Relu activation function for FCNN prevents the model from generating negative RVs.



(a) stock-related headlines

(b) general hot headlines

Figure 5: Distribution of Daily Tokens

Notes: The number of daily tokens is calculated, and their distributions are plotted after putting daily stock-related news (left plot) and general hot (right plot) headlines together (train data - 2046 days). The vertical line is the chosen maximum length of the padding.

(Kingma and Ba, 2014), and MSE is the objective function of this network. To prevent the model from over-fitting, L^2 regularisation with a weight decay value set equal to 3 is used for both CNN and FCNN, while the dropout rate is set equal to 0.5 between the CNN and FCNN. The timespan for headlines of the day t starts from 9:30 AM Eastern Time of day t and ends at 9:30 AM Eastern Time of the day $t+1$. Daily training of this model is computationally intensive; therefore, the training process is repeated every thirty days, and the trained model is used for the next days. In order to have reproducible results, a random number generator (RNG) with the same seed is used for all trained models.

Dow Jones Newswires Text News Feed provides a tagging system for finding news stories related to a specific stock. Therefore, in this study, considering the timespan of our analysis and the availability of this tagging system during the timespan, the tag ('about') is used for extracting the stock-related news for each ticker. This tag denotes a story about a ticker but of no particularly significant. Also, to better understand the effect of not tagged news stories on RV, we expand our analysis to general hot news. Hot news stories without any tag are chosen for general hot news. 'Hot' tag means a news story is deemed 'important' or 'timely' in some way. Also, for this group of news, only US market news stories are chosen to reduce the length of daily tokens as much as possible. What is important for us to recognise here is that this procedure assures the news stories for stock-related analysis and general hot news are distinct. Returning to padding, Figure 5 shows the distribution of daily tokens (headlines) for stock-related news in Figure 5a and general hot news in Figure 5b. What can be clearly seen from these figures is that, as expected, the daily number of tokens in headlines is higher for general hot news compared with stock-related news; therefore, to process relatively same amount of textual data for both cases, a higher padding value (2000 vs 500) is chosen for general hot news.

4.3 Explainable AI (XAI)

This subsection describes shapely additive explanations (SHAP), one of the well-known XAI methods for making ML models more transparent. Lundberg and Lee (2017) proposed the SHAP method based

on the coalition game theory. Shapley values ϕ_i, defined below, show the importance of a model input S (a set of tokens in daily news headlines) given the model output $f(S)$, the volatility forecast. In this case:

$$\phi_i = \frac{1}{|N|!} \sum_{S \subseteq N \setminus \{i\}} |S|! \ (|N| - |S| - 1)! \ [f(S \cup \{i\}) - f(S)], \tag{9}$$

where $f(S \cup \{i\}) - f(S)$ captures the marginal contribution in volatility forecast of adding token i to the set S, N contains all model inputs, $|S|!$ shows the number of different ways the chosen set of tokens may be presented, and $(|N| - |S| - 1)!$ is the number of different ways that the remaining tokens could have been added. The Shapley values ϕ_i show the magnitude and sign of the average contribution of token i; it satisfies three properties, viz. local accuracy (additivity)[1], missingness (nonexistence or null effect)[2], and consistency (symmetry)[3]. As tokens are added to the set, changes in the RV forecast reflect their relevance. The benefits of the SHAP approach include a solid theoretical foundation in game theory and no requirement for differentiable models. However, it is computationally intensive, and, like other permutation-based approaches, it does not consider feature dependencies and may generate misleading results. Here, we use a high-speed approximation algorithm, Deep SHAP based on DeepLIFT (Shrikumar et al., 2017), to calculate SHAP values.

Following Zhao et al. (2020), SHAP is applied to the classification part of the model (FCNN). The inputs, in this case, are outputs of the global max pooling layer, and the output is the forecasted RV (RV_{t+1}) in Figure 4. Therefore, the number of inputs is equal to $3h$, where h is the number of filters, and 3 is the number of filter sizes (unigram, bigram, and trigram). Applying SHAP on the full model causes misleading results because nonexistent words in the text may mistakenly receive high SHAP scores. However, by applying SHAP on the classification part, the permutation is done over extracted features from filters; therefore, the SHAP scores are attributed to the unigrams, bigrams, and trigrams. Moreover, duplication of filters is common in CNN, especially for large h values (RoyChowdhury et al., 2017), and this causes generating similar n-grams. In order to avoid this, a de-duplication step is applied by deleting repeated n-grams and calculating the new SHAP value by adding up the SHAP values.[4]

Although the SHAP values show the importance of constituent n-grams of each test sample, it does not provide information about the important n-grams over entire test samples of all 23 analysed stocks. In order to identify the volatility movers over the entire test samples for all 23 stocks, first, we store the top five n-grams with the highest absolute SHAP values over test samples for each stock. This results in 23 lists of n-grams. Next, we search for the shared n-grams among all 23 stocks and calculate their number of repetitions. Finally, we extract the top t ($t = 1 : 23$) with the highest number of repetitions. Each top repetition group contains several n-grams. By reviewing these groups and incorporating general financial knowledge, it is possible to easily find and classify the important

[1]It means that sum of the individual token attributions is equal to the forecasted RV.

[2]It means that a missing token has no attributed impact ($\phi_i = 0$).

[3]It means that if a change in a specific token has a more considerable impact on the first model compared to the second model, the importance of this token should be higher for the first model than the second model.

[4]Zhao et al. (2020) proposed two steps for de-duplication, namely 'exact de-duplication' (applied in our study) and 'merge de-duplication'. In 'merge de-duplication', overlapped n-grams are merged, and their SHAP value is calculated by adding up the SHAP values of the constituent n-grams. Due to the generally shorter input text length, our experiments show that incorporating 'exact de-duplication' is sufficient to reach more granular results.

names, language patterns, and structures as the primary movers of volatility. This is applicable to other asset pricing cases, free of statistical assumptions, and, more importantly, for each specific asset pricing case, it generates a specialised list of important n-grams.

5 Results

For each stock i and year m, the performance difference between the model j and CHAR as the best performing model in Rahimikia and Poon (2020a) is measured as follow:

$$Avg_m\ \Delta_{MSE,i,j} = \frac{1}{23}\sum_{i=1}^{23}(MSE_i(NLP\text{-}ML_j) - MSE_i(CHAR)), \tag{10}$$

$$Med_m\ \Delta_{MSE,i,j} = Median(MSE_i(NLP\text{-}ML_j) - MSE_i(CHAR)). \tag{11}$$

In equations (10) and (11), MSE can be replaced by QLIKE[1] (Patton, 2011). For MSE and QLIKE, a negative value in equations (10) and (11) indicates improvement, and a positive value indicates degradation in performance. Finally, reality check (RC) is used for comparing each model against all models in the HAR-family (viz. AR1, HAR, HAR-J, CHAR, SHAR, ARQ, HARQ, and HARQ-F) as follows:

$$\begin{aligned} H_0 &: \underset{k=1,\ldots,n}{Min}\ \mathbb{E}[L^k(RV,X) - L^0(RV,X)] \leq 0, \\ H_1 &: \underset{k=1,\ldots,n}{Min}\ \mathbb{E}[L^k(RV,X) - L^0(RV,X)] > 0, \end{aligned} \tag{12}$$

where L^k is the loss from the benchmark (HAR-family of models), L^0 is the loss from the specific model, and n is the number of benchmark models (in this study, $n = 8$). Rejection of H_0 means that the loss from the model is significantly smaller than that from all benchmark models. For this RC test, we follow the stationary bootstrap of Politis and Romano (1994) with 999 re-samplings and an average block length of 5 (Bollerslev et al., 2016).[2]

Rahimikia and Poon (2020a) showed the importance of separating normal volatility days and high volatility days when evaluating out-of-sample forecasting performance. A day is defined as a high volatility day when RV for that day is greater than $Q3 + 1.5\ IQR$, where $IRQ = Q3 - Q1$, and $Q1$ and $Q3$ are, respectively, the first and third quantiles of RV. By applying this criterion to the sample of 23 stocks, about 10% (160 days) of the out-of-sample period (1604 days) are classified as high volatility days.

5.1 Stock-Related News

Table 6 reports the out-of-sample RC results for different models with 25, 50, 75, and 100 CNN filters. A model with a lower number of filters is less complex than one with a higher number of filters. In Table 6, RC is the percentage of tickers with outstanding performance against all HAR-family of

[1] $QLIKE(RV_t, \widehat{RV_t}) \equiv \frac{RV_t}{\widehat{RV_t}} - log(\frac{RV_t}{\widehat{RV_t}}) - 1$ where RV_t and $\widehat{RV_t}$ are the true and fitted RV at time t respectively. Patton (2011) showed that this loss function, MSE, and their variations are the only class of robust loss functions for ranking volatility forecasting models.

[2] Our analysis shows that the results are not sensitive to the choice of block length.

Table 6: Out-of-sample RC results (stock-related news)

Full out-of-sample period		FinText (CBOW)				FinText(skip-gram)				Google(skip-gram)				WikiNews(FT[b]/skip-gram)				FinText (FT/skip-gram)				FinText(FT/CBOW)			
		25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100	25	50	75	100
MSE[a]	0.05	91.30	86.96	91.30	86.96	**82.61**	**82.61**	**82.61**	**82.61**	65.22	65.22	65.22	65.22	56.52	60.87	60.87	60.87	**86.96**	**86.96**	**86.96**	**86.96**	73.91	78.26	65.22	65.22
	0.10	100	100	100	100	**100**	**100**	**100**	**100**	95.65	95.65	100	95.65	95.65	91.30	91.30	91.30	**100**	**100**	**100**	**100**	95.65	91.30	95.65	91.30
QLIKE	0.05	13.04	17.39	17.39	17.39	**21.74**	**21.74**	**21.74**	**21.74**	13.04	13.04	13.04	13.04	4.35	4.35	4.35	4.35	**21.74**	**21.74**	**21.74**	**21.74**	17.39	13.04	13.04	4.35
	0.10	30.44	43.48	26.09	30.44	**39.13**	**34.78**	**39.13**	**30.44**	30.44	30.44	30.44	30.44	21.74	21.74	21.74	21.74	**34.78**	**34.78**	**34.78**	**30.44**	26.09	17.39	21.74	21.74
Normal volatility days																									
MSE	0.05	17.39	17.39	13.04	17.39	**69.57**	**65.22**	**65.22**	**65.22**	69.57	60.87	60.87	60.87	60.87	60.87	60.87	56.52	**69.57**	**65.22**	**65.22**	**60.87**	43.48	34.78	30.44	39.13
	0.10	21.74	17.39	17.39	17.39	**69.57**	**69.57**	**73.91**	**69.57**	69.57	65.22	65.22	65.22	65.22	60.87	60.87	60.87	**73.91**	**69.57**	**69.57**	**69.57**	65.22	34.78	47.83	52.17
QLIKE	0.05	0	0	0	0	**4.35**	**4.35**	**4.35**	**4.35**	4.35	4.35	4.35	4.35	0	0	0	0	**4.35**	**4.35**	**4.35**	**4.35**	4.35	0	4.35	4.35
	0.10	0	0	0	0	**4.35**	**4.35**	**4.35**	**4.35**	4.35	4.35	4.35	4.35	0	4.35	4.35	0	**4.35**	**4.35**	**4.35**	**4.35**	4.35	0	4.35	4.35
High volatility days																									
MSE	0.05	100	100	100	100	**65.22**	**69.57**	**69.57**	**69.57**	65.22	65.22	65.22	65.22	56.52	60.87	60.87	60.87	**65.22**	**69.57**	**65.22**	**69.57**	69.57	82.61	82.61	69.57
	0.10	100	100	100	100	**95.65**	**95.65**	**95.65**	**95.65**	91.30	91.30	91.30	91.30	82.61	82.61	82.61	82.61	**95.65**	**95.65**	**95.65**	**95.65**	95.65	95.65	100	100
QLIKE	0.05	39.13	43.48	52.17	47.83	**43.48**	**43.48**	**43.48**	**43.48**	34.78	39.13	39.13	43.48	13.04	30.44	30.44	30.44	**43.48**	**43.48**	**43.48**	**43.48**	21.74	34.78	21.74	8.70
	0.10	69.57	60.87	65.22	73.91	**60.87**	**69.57**	**65.22**	**65.22**	52.17	52.17	56.52	60.87	43.48	52.17	52.17	56.52	**65.22**	**69.57**	**69.57**	**69.57**	43.48	56.52	47.83	39.13

Notes: [a] Percentage of tickers with outstanding performance considering different numbers of CNN filters (25, 50, 75, and 100) at the 5% and 10% significance levels of the RC compared to all HAR-family of models.
[b] FastText algorithm.

models at the 5% and 10% significant levels with MSE(QLIKE) as the loss function.[1] Figure 9a in Figure 9 shows the word cloud of stock-related headlines for all 23 stocks together over the out-of-sample period. As expected, it is obvious that the most repeated tokens are associated with news stories directly about companies and their operations.

The top panel of Table 6 is for the full out-of-sample period, the middle one is for normal volatility days, and the bottom panel is for high volatility days. 'FT' also refers to the FastText algorithm. Focusing on both MSE and QLIKE loss functions, it's clear that generally, 'FinText(skip-gram)' and 'FinText(FT/skip-gram)' are reaching the highest RC values for the full out-of-sample period, normal volatility days and also high volatility days, and this is more prominent for MSE than QLIKE loss function. Closer inspection of Table 6 also shows substantially higher RCs for high volatility days compared with normal volatility days. Moreover, the CBOW model ('FinText(CBOW)' and 'FinText(FT/CBOW)') generally shows good performance for high volatility days, but their performance for normal volatility days is poor. More importantly, as discussed in Subsection 3.1, although *FinText* used a substantially smaller corpus compared with the other major well-known pre-trained word embeddings, RC values show its superior performance in RV forecasting compared with different variations of these word embeddings. Finally, the results of this experiment show no clear-cut pattern of improvement in RCs by increasing the complexity of models; therefore, for the sake of clarity, 50 as the number of filters is chosen for the subsequent analysis.

Along with the higher out-of-sample forecasting performance from RC results, however, there is concern over the temporal amount of this improvement. In Figure 6, Each line represents the yearly difference between the average of the out-of-sample MSEs (left plots) and QLIKEs (right plots) of the specified model with the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020a)) for 23 tickers. A negative value shows improvement, and a positive value shows degradation in performance. The top, middle, and bottom plots show the results for the full out-of-sample period, normal volatility days, and high volatility days, respectively. The horizontal dashed line represents no improvement. Figure 6 is quite revealing in several ways. First, as expected, for most out-of-sample years, all models show improvement in forecasting performance for high volatility days for both MSE and QLIKE loss functions. As discussed, this improvement is statistically significant for the majority of stocks. However, this is not valid for normal volatility days. Second, *FinText* is a clear winner, and the temporal forecasting improvement is more prominent for the 'FinText(skip−gram)' and 'FinText(FT/skip−gram)'. This finding is consistent with our proposed gold-standard financial benchmark results in Subsubsection 3.2.3.

Perhaps the most interesting aspect of Figure 6 is the clear degradation in forecasting performance during the out-of-sample period from 2015 to 2022 for high volatility days. Undeniably, during the COVID-19 outbreak, this degradation reaches the highest amount for high volatility days. To further investigate the probable reason behind this, the monthly average share percentage of OOV n-grams for each stock over out-of-sample news stories is calculated. Figure 7 depicts the average value over all 23 stocks for the *FinText* (top chart), Google Word2Vec (middle chart) and Facebook WikiNews

[1] A note of caution is due here since Facebook WikiNews word embedding covers data from 2007 to 2017, so there may be data leaking through word embedding, causing spurious better out-of-sample performance of this model for around two years.

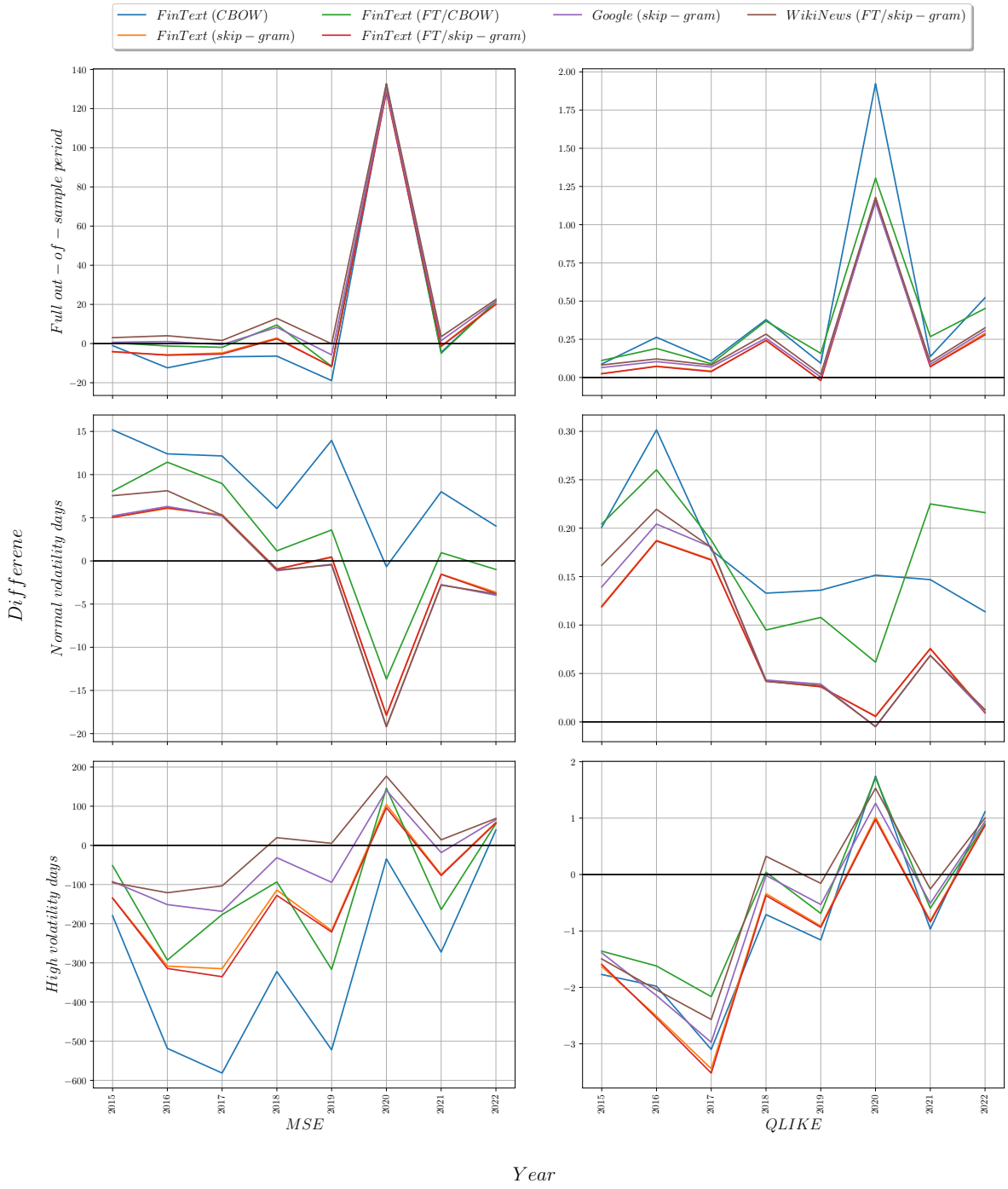


Figure 6: Yearly performance comparison (stock-related news)

Notes: The line represents the yearly difference between the average of the out-of-sample MSEs (left plots) and QLIKEs (right plots) of the specified model with the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020a)) for 23 tickers (negative value shows improvement, and positive value shows degradation in performance). The top, middle, and bottom plots show the results for the full out-of-sample period, normal volatility days, and high volatility days, respectively. The horizontal dashed line represents no improvement.

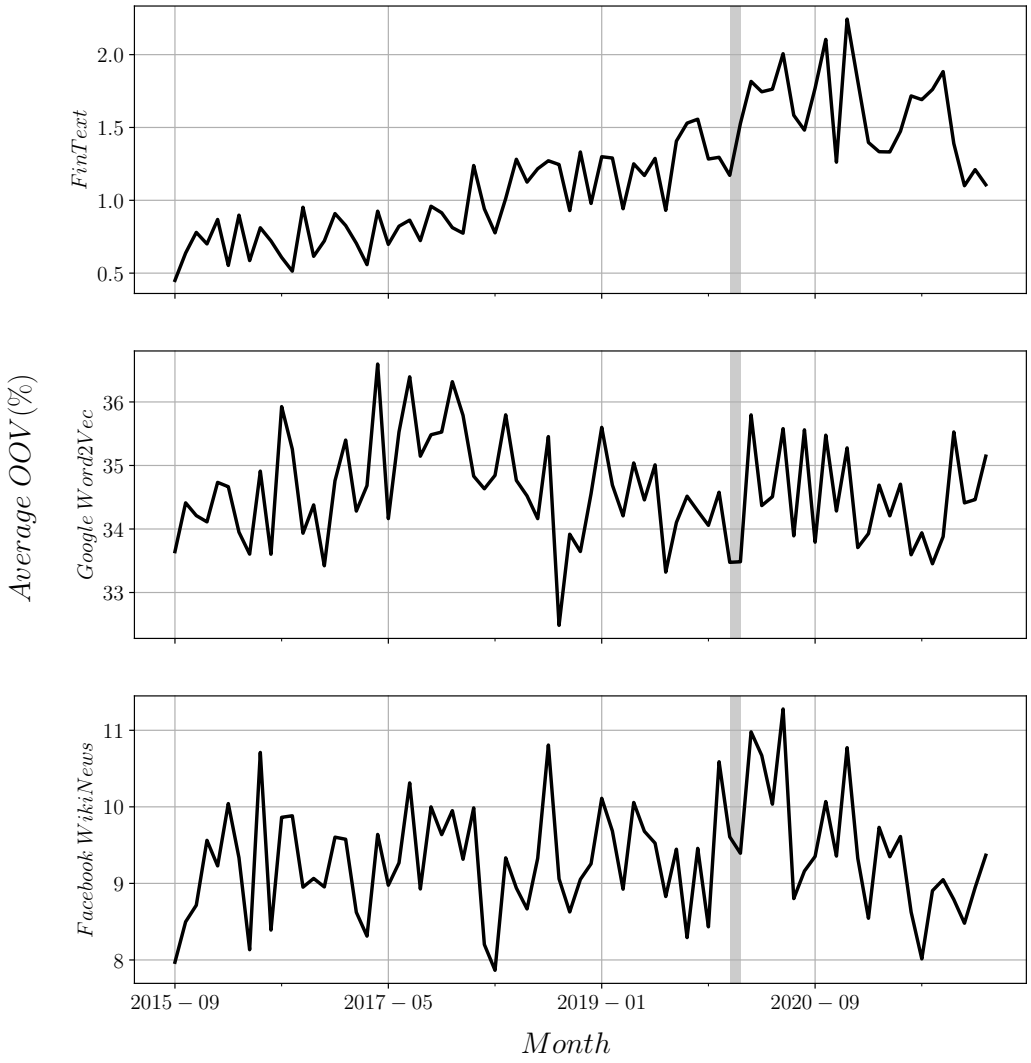


Figure 7: Average OOV over out-of-sample period (stock-related news)

Notes: The monthly average share percentage of OOV n-grams for each stock over out-of-sample news stories is calculated. The average OOV(%) is the average value over all 23 stocks for the *FinText* (top chart), Google Word2Vec (middle chart) and Facebook WikiNews (bottom chart) word embeddings. December 2020 is highlighted as the beginning of the COVID-19 outbreak.

Table 7: Out-of-sample RC results of models (general hot news)

Full out-of-sample period		FinText (CBOW) 25	50	75	100	FinText(skip-gram) 25	50	75	100	Google(skip-gram) 25	50	75	100	WikiNews(FT[b]/skip-gram) 25	50	75	100	FinText (FT/skip-gram) 25	50	75	100	FinText(FT/CBOW) 25	50	75	100
MSE[a]	0.05	21.74	13.04	13.04	17.39	43.48	34.78	39.13	39.13	**39.13**	**39.13**	**39.13**	**39.13**	**34.78**	**34.78**	**34.78**	**39.13**	43.48	39.13	34.78	39.13	26.09	30.44	30.44	34.78
	0.10	56.52	47.83	43.48	39.13	91.30	91.30	91.30	91.30	**95.65**	**95.65**	**95.65**	**91.30**	**91.30**	**91.30**	**86.96**	**86.96**	91.30	91.30	91.30	91.30	78.26	86.96	82.61	82.61
QLIKE	0.05	0	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0	0	0	0	0	0	0	0
	0.10	0	0	0	0	4.35	4.35	4.35	4.35	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	4.35	4.35	4.35	4.35	0	0	0	4.35
Normal volatility days																									
MSE	0.05	4.35	4.35	4.35	0	69.57	47.83	47.83	47.83	**78.26**	**69.57**	**69.57**	**65.22**	**86.96**	**82.61**	**78.26**	**73.91**	69.57	43.48	47.83	43.48	56.52	39.13	34.78	43.48
	0.10	4.35	8.70	8.70	8.70	73.91	47.83	52.17	47.83	**78.26**	**73.91**	**73.91**	**73.91**	**86.96**	**86.96**	**86.96**	**82.61**	73.91	52.17	52.17	43.48	56.52	39.13	47.83	43.48
QLIKE	0.05	0	0	0	0	8.70	4.35	4.35	4.35	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	4.35	4.35	4.35	4.35	0	0	0	0
	0.10	0	0	0	0	8.70	4.35	4.35	4.35	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	**4.35**	8.70	4.35	4.35	4.35	0	0	0	0
High volatility days																									
MSE	0.05	43.48	47.83	47.83	52.17	34.78	39.13	39.13	39.13	**34.78**	**34.78**	**34.78**	**39.13**	**30.44**	**34.78**	**34.78**	**34.78**	34.78	39.13	39.13	39.13	30.44	39.13	39.13	39.13
	0.10	65.22	91.30	91.30	86.96	69.57	69.57	69.57	69.57	**69.57**	**73.91**	**73.91**	**73.91**	**60.87**	**60.87**	**60.87**	**60.87**	69.57	69.57	69.57	69.57	56.52	82.61	73.91	69.57
QLIKE	0.05	0	0	4.35	4.35	8.70	8.70	8.70	8.70	**8.70**	**8.70**	**8.70**	**8.70**	**0**	**0**	**0**	**0**	4.35	4.35	4.35	8.70	0	0	0	0
	0.10	8.70	4.35	8.70	17.39	17.39	17.39	21.74	21.74	**17.39**	**17.39**	**21.74**	**26.09**	**13.04**	**13.04**	**17.39**	**17.39**	21.74	17.39	21.74	21.74	8.70	17.39	4.35	4.35

Notes: [a] Percentage of tickers with outstanding performance considering different numbers of CNN filters (25, 50, 75, and 100) at the 5% and 10% significance levels of the RC compared to all HAR-family of models.
[b] FastText algorithm.

(bottom chart) word embeddings. December 2020 is highlighted as the beginning of the COVID-19 outbreak. It is clear that for *FinText*, the average OOV is increasing over time with a sudden jump during the Covid-19 outbreak. This sudden jump, with less intensity, is also noticeable for Google Word2Vec and Facebook WikiNews. This finding is suggestive of a link between the increase in the average percentage of unknown vocabulary during the COVID-19 spread and the sudden decrease in the forecasting performance in Figure 6. Still, except for *FinText* with an overall increasing average OOV during the time, Figure 7 cannot explain why the RV forecasting performance is degrading over time. This result may be explained by the fact that all word embeddings are trained by a specific corpus covering a limited time horizon; therefore, generally, it is expected to observe a decrease in performance over time. The average OOV trend may explain a part of this degradation (like the COVID-19 outbreak). However, the rest is liked to the changes in semantics over time, which are defined by word vectors in the word embeddings.

Further analysis of Figure 7 reveals interesting findings. From this figure, Google Word2Vec and *FinText* show the highest and lowest average OOV over time, respectively. The lower average OOV for *FinText* is likely related to the source of out-of-sample data, which is structurally similar to the *FinText* training data. For Google Word2Vec, although a substantially larger corpus with around 100 billion words is used for training this word embedding, it shows a higher average. This finding draws our attention to the importance of the corpus and its field-specific quality for training word embeddings and, in general, NLP models in finance. In view of all that has been mentioned so far, one may suppose that stock-related news is the only possible source of news for this study. Because of the high flexibility of these models in asset pricing, switching from stock-related news to general hot news for RV forecasting is covered in the following subsection.

5.2 General Hot News

As discussed, it is widely accepted that news is a potential contributor to volatility, although it needs to be clarified what type of news has more contribution and how. Therefore, this subsection seeks to address the importance of general hot news for RV forecasting. It is important to stress that general hot news is entirely distinct from stock-related news. Also, unlike stock-related news, all models here are re-trained using the same group of in-sample news stories, and the same group of news stories are used for forecasting RV over the out-of-sample period. Figure 9b in Figure 9 shows the word cloud of general hot headlines over the out-of-sample period. In line with our expectations, general hot news covers different topics focusing on major economic, financial, political, and geopolitical events.

The out-of-sample RC results for general hot news are presented in Table 7. This table is quite revealing in several ways. First, mainly for the MSE loss function, normal volatility days benefited the most from general hot news. Second, both Google Word2Vec and Facebook WikiNews ('Google(skip-gram)' and 'WikiNews(FT/skip-gram') generally show higher forecasting performance for normal volatility days. More precisely, although *FinText* is still a pioneer in the full out-of-sample period due to slightly higher performance for high volatility days, it is interesting that with a noticeable margin, it shows a lower performance for normal volatility days.

Table 7 depicts the amount of temporal improvement over the out-of-sample period. Following

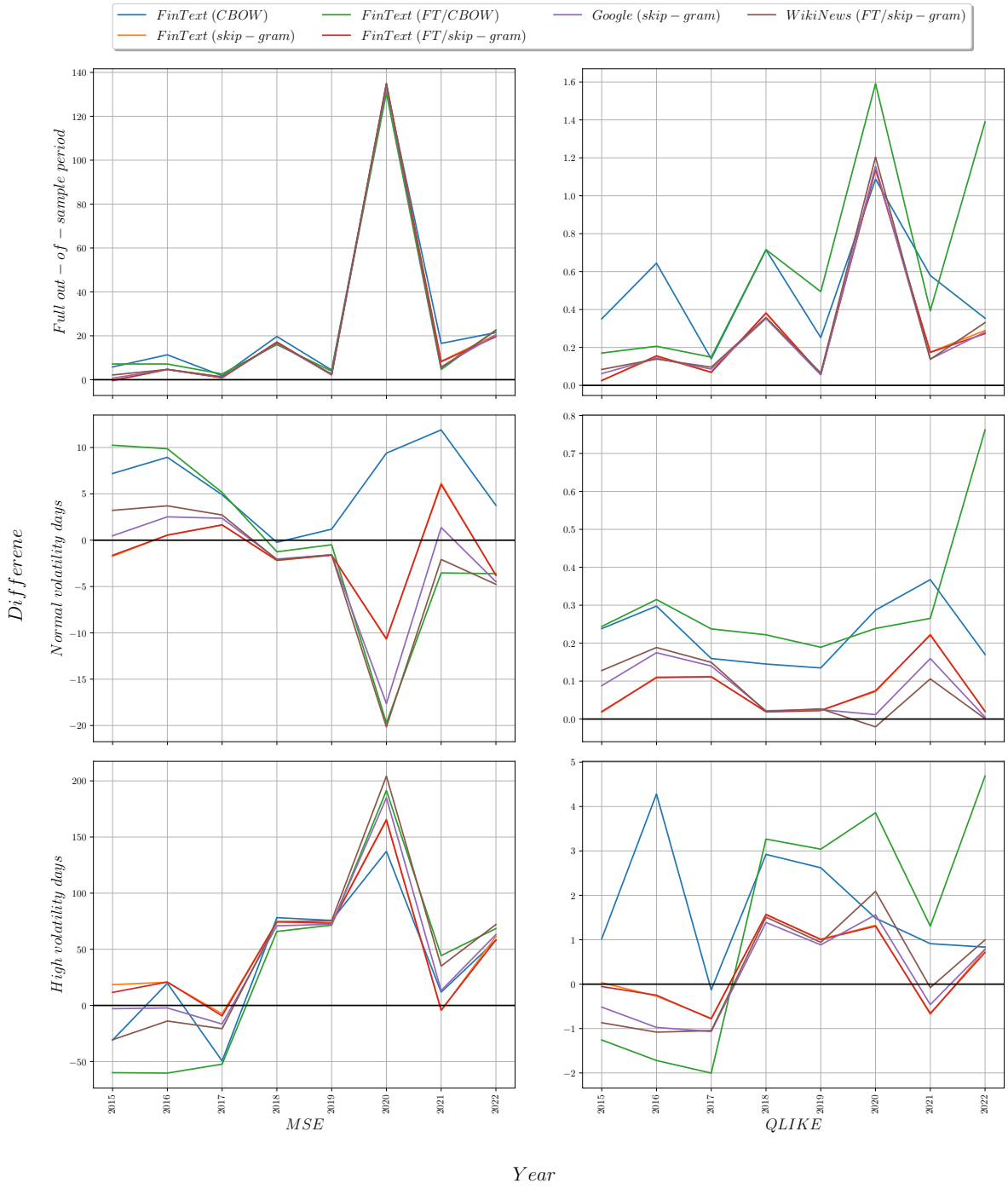
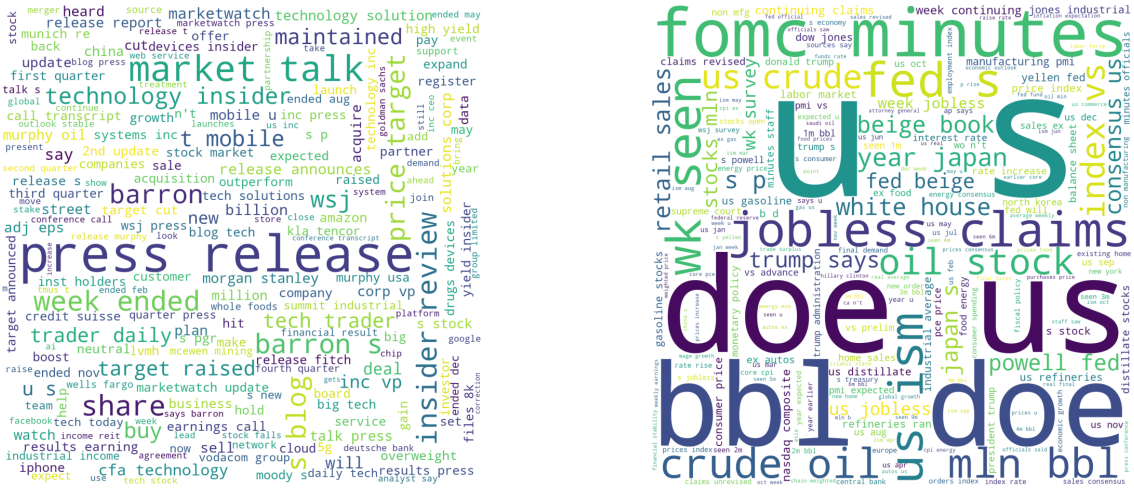


Figure 8: Yearly performance comparison (general hot news)

Notes: The line represents the yearly difference between the average of the out-of-sample MSEs (left plots) and QLIKEs (right plots) of the specified model with the CHAR model (the best performing HAR-family model in Rahimikia and Poon (2020a)) for 23 tickers (negative value shows improvement, and positive value shows degradation in performance). The top, middle, and bottom plots show the results for the full out-of-sample period, normal volatility days, and high volatility days, respectively. The horizontal dashed line represents no improvement.



(a) Stock-related headlines

(b) General hot headlines

Figure 9: Out-of-sample word cloud

Notes: (a) Word cloud of stock-related headlines for all 23 stocks together over the out-of-sample period. (b) Word cloud of general hot headlines over the out-of-sample period.

Subsection 5.1, 50 as the number of filters is chosen for this analysis because of no clear-cut pattern of improvement in RCs by increasing the complexity. What can be clearly seen from this figure is the better forecasting performance of general hot news for normal volatility days compared with stock-related news in Figure 6. Also, as expected, this improvement is primarily noticeable for the MSE loss function, not the QLIKE loss function. Although *FinText* is trained by the textual data with a similar structure as out-of-sample data, the better forecasting performance of both Google Word2Vec ('Google(skip-gram)') and Facebook WikiNews ('WikiNews(FT/skip−gram)') is clear over time for normal volatility days. This improvement can be explained, at least in part, by the size and mixture of their corpora because both contain substantially larger and more diverse textual data. Therefore, even though a variety of news stories made our corpus, there is still a potential to improve its general-knowledge-oriented performance by adding more diverse data.

So far, the evidence shows that, in general, for high volatility days, stock-related news has a higher potential predictive contribution in RV, but for normal volatility days, the predictive source switches from stock-related news to general news, which can mostly be considered as market movers. However, it does not ascertain whether the mixture of these two news types simultaneously has the potential to improve the RV forecasting performance or not. A summary of simple ensemble models is provided in the following subsection to investigate this further.

5.3 Ensemble Models

In this study, the ensemble model forecast is simply the arithmetic mean of the RV forecasts. Table 8 displays the RC results for different combinations of model complexities. The stock-related news group covers the *FinText* model with word embedding trained by the Skip-gram model and FastText algorithm, and the general hot news group covers the Facebook WikiNews model as the best-performing group of models in Subsection 5.1 and Subsection 5.2, respectively. Detailed examination of ML by

Rahimikia and Poon (2020b) and Christensen et al. (2021) showed the potential of ML for RV forecasting. Rahimikia and Poon (2020b) have also shown that ML substantially improves RV forecasting performance for normal volatility days but not high volatility days; Therefore, from Rahimikia and Poon (2020b), two proposed ML forms, namely FCNN and OB-ML, are added to this table. FCNN form mimics the simple HAR model structure with three independent variables and only replaces the OLS regression with a simple FCNN model for switching the linear OLS to nonlinear FCNN. OB-ML is a more complex form covering not only 6 HAR-family variables (RV, BPV, BPV jump, negative RV, positive RV, and realised quarticity) but also 132 LOB features as independent variables. This study showed the potential of a simple LSTM model for RV forecasting in this high-dimensional environment. Taken together, stock-related and general hot news incorporate just textual news for RV forecasting. However, FCNN and OB-ML only incorporate historical time series of commonly used variables from HAR-family and LOB.

The top part of Table 8 shows the RC results at the 10% significance levels for both MSE and QLIKE loss functions for the full out-of-sample period, following normal volatility days and high volatility days in the middle and bottom parts. The highest RC values for the top, middle and bottom parts are marked in bold. What stands out in this table is that for the full out-of-sample period and also normal volatility days, the ensemble of stock-related news and OB-ML reaches the highest RC values for both MSE and QLIKE loss functions. It is also apparent from this table that the ensemble of stock-related and general hot news reaches the highest RC regarding the MSE loss function. However, for both MSE and QLIKE loss functions, the stock-related news and OB-ML ensemble still show reasonably high RC values. Finally, although the ensemble of stock-related news and FCNN shows RV forecasting performance improvement for normal volatility days, this improvement, mainly focusing on the QLIKE loss function, is negligible for high volatility days. Therefore, keeping the same limited number of independent HAR-family variables and only changing the OLS linear form to FCNN nonlinear form is helpful to improve the performance but not as much as the extra substantial forecasting power LOB features bring inside. This finding is expected and shows the power of ML models for modelling a high number of independent variables simultaneously.

Taken together, stock-related news tends to be suitable for forecasting high volatility days. However, with financial numbers, the process is different, and it substantially improves the RV forecasting performance for normal volatility days. From Table 8 and the ensemble of stock-related and general hot news results, it is evident that the improvement from modelling financial numbers, especially in a high dimensional environment, is much more significant than the potential of general hot news for improving the RV forecasting performance of normal volatility days. Collectively, what emerges from the results reported here is the importance of the information content of both financial numbers and news for forecasting RV. Although the reported ensemble results show substantial improvement in RV forecasting, the nature of this improvement remains unclear. What is interesting about the textual data is its readability; therefore, one should not, of course, accept this improvement without questioning the real impact of n-grams on RV forecasting. Section 6 explores this for both stock-related and general hot news.

Table 8: Ensemble models RC results

		General hot news				FCNN in Rahimikia and Poon (2020b)					OB-ML in Rahimikia and Poon (2020b)				
		25	50	75	100	5	10	15	20	25	5	10	15	20	25
Full out-of-sample period															
	25	100[a]	100	100	100	87.0	87.0	95.7	91.3	91.3	95.7	100	100	**100**	100
		34.8[b]	34.8	34.8	30.4	26.1	26.1	26.1	26.1	26.1	30.4	56.5	78.3	**82**.6	78.3
	50	100	100	100	100	91.3	91.3	100	95.7	95.7	100	100	100	**100**	**100**
		34.8	34.8	34.8	34.8	26.1	26.1	21.7	26.1	26.1	39.1	60.9	78.3	**82.6**	**82.6**
	75	100	100	100	100	91.3	91.3	100	95.7	95.7	100	100	100	**100**	**100**
		34.8	34.8	34.8	34.8	26.1	26.1	21.7	26.1	26.1	34.8	60.9	78.3	**82.6**	**82.6**
	100	100	100	100	100	91.3	91.3	100	95.7	95.7	100	100	100	**100**	**100**
		34.8	30.4	30.4	30.4	26.1	26.1	21.7	26.1	26.1	34.8	60.9	78.3	**82.6**	**82.6**
Normal volatility days															
Stock-related news	25	87.0	78.3	78.3	78.3	95.7	95.7	95.7	87.0	87.0	100	**100**	**100**	100	100
		4.3	4.3	4.3	4.3	52.2	52.2	52.2	47.8	47.8	73.9	**82.6**	**82.6**	78.3	78.3
	50	78.3	73.9	73.9	69.6	95.7	95.7	95.7	87.0	87.0	100	**100**	100	100	100
		4.3	4.3	4.3	4.3	52.2	52.2	47.8	47.8	47.8	73.9	**82.6**	78.3	73.9	65.2
	75	78.3	73.9	73.9	69.6	95.7	95.7	95.7	87.0	87.0	100	**100**	100	100	100
		4.3	4.3	4.3	4.3	52.2	52.2	52.2	47.8	47.8	73.9	**82.6**	78.3	73.9	60.9
	100	78.3	73.9	73.9	69.6	95.7	95.7	95.7	87.0	87.0	100	**100**	100	100	100
		4.3	4.3	4.3	4.3	52.2	52.2	47.8	47.8	47.8	69.6	**82.6**	78.3	73.9	56.5
High volatility days															
	25	**91.3**	**91.3**	**91.3**	**91.3**	78.3	78.3	78.3	78.3	78.3	73.9	78.3	78.3	87.0	87.0
		60.9	60.9	60.9	60.9	8.7	8.7	8.7	8.7	8.7	21.7	26.1	39.1	56.5	65.2
	50	**91.3**	**91.3**	**91.3**	**91.3**	78.3	78.3	78.3	78.3	78.3	73.9	78.3	78.3	**91.3**	87.0
		60.9	60.9	60.9	60.9	17.4	17.4	13.0	13.0	13.0	21.7	30.4	47.8	56.5	69.6
	75	**91.3**	**91.3**	**91.3**	**91.3**	78.3	78.3	78.3	78.3	78.3	73.9	78.3	78.3	**91.3**	87.0
		60.9	60.9	60.9	60.9	17.4	17.4	13.0	13.0	13.0	21.7	30.4	47.8	56.5	69.6
	100	**91.3**	**91.3**	**91.3**	**91.3**	78.3	78.3	78.3	78.3	78.3	73.9	78.3	78.3	**91.3**	87.0
		60.9	60.9	60.9	60.9	13.0	13.0	13.0	13.0	13.0	21.7	30.4	47.8	65.2	**73.9**

Notes: [a] Percentage of tickers with outstanding performance at the 10% significance levels of the RC compared to the all HAR-family of models for the MSE loss function. [b] Percentage of tickers with outstanding performance at the 10% significance levels of the RC compared to the all HAR-family of models for the QLIKE loss function.

6 XAI Results

Over the last few years, great efforts have been made to understand ML models better, which are often described as black-box. Here, we will explore one of the most prominent XAI methods, SHAP, to analyse the impact of textual news on RV forecasting. For stock-related news, the *FinText* model utilising the FastText algorithm and Skip-gram model with 50 filters is chosen as one of the best performing models in Subsection 5.1. Also, for general hot news, the Facebook WikiNews model utilising the FastText algorithm and Skip-gram model with 50 filters is chosen as one of the best performing models in Subsection 5.2.[1] Subsection 6.1 gives a brief comparison of the XAI and LM dictionary results, and Subsection 6.2 summaries the XAI results by classifying the RV movers separately for stock-related and general hot news.

6.1 XAI vs LM dictionary

By far, the most widely accepted and influential account of sentiment analysis in finance is to be found in the work of Loughran and McDonald (2011). They developed a dictionary of different sentiments by monitoring a large sample of 10-Ks. Although they extended this analysis by categorising positive, uncertainty, litigious, strong modal, moderate modal, weak modal, and constraining words, the negative category is the most extensive one and is significantly related to announcement returns. Figure 10 displays the scatter diagram of the SHAP values for the top three LM negative words with the highest count value ('loss', 'termination', and 'against'). Each chosen word is also grouped with its variations. Therefore, {'loss', 'losses'}, {'termination', 'terminate', 'terminates', 'terminated'}, and {'against'} represent 'loss', 'termination', and 'against' top negative words, respectively. Figure 10 rows represent these top three negative groups. The left (right) column represents the SHAP values for stock-related (general hot) news published during the out-of-sample period. The x-axis is the reported explainer value, and the y-axis is the ticker name. The vertical line represents no impact on RV, while the left (right) hand side represents the negative (positive) impact of a specified group of words on RV. A larger positive (negative) value means a larger increase (decrease) in the RV forecast. The total number of negative and positive SHAP values is depicted at the top for each figure. As explained in Subsection 6.1, SHAP is applied to the classification part of the model; therefore, an n-gram is chosen for this representation when it contains at least one of the words in each group of top negative words. Also, there are two reasons for the differences in the number of tickers in each plot. A ticker is missing because the specified word group had no appearance in the news stories, or the model did not identify this word group as important for this ticker.

An inspection of the data in Figure 10 reveals that our model allows for a greater variety of contextual relationships between each group of top negative words and RV forecast. 'Loss' negative word group pushes the RV to higher and lower values for 355 and 263 appearances in stock-related news (top left plot) and for 101 and 204 appearances in general hot news (top right plot) over the out-of-sample period. Moving to the 'Termination' negative group, these values change to 41 and 13 for stock-related news and 0 and 2 for general hot news. Finally, for the 'against' word group, these values change to 146 and 148 for stock-related news and 92 and 273 for general hot news. In contrast, the

[1] Our analysis shows that the results are not sensitive to the model complexity; therefore, the same number of filters are chosen for both stock-related and general hot news.

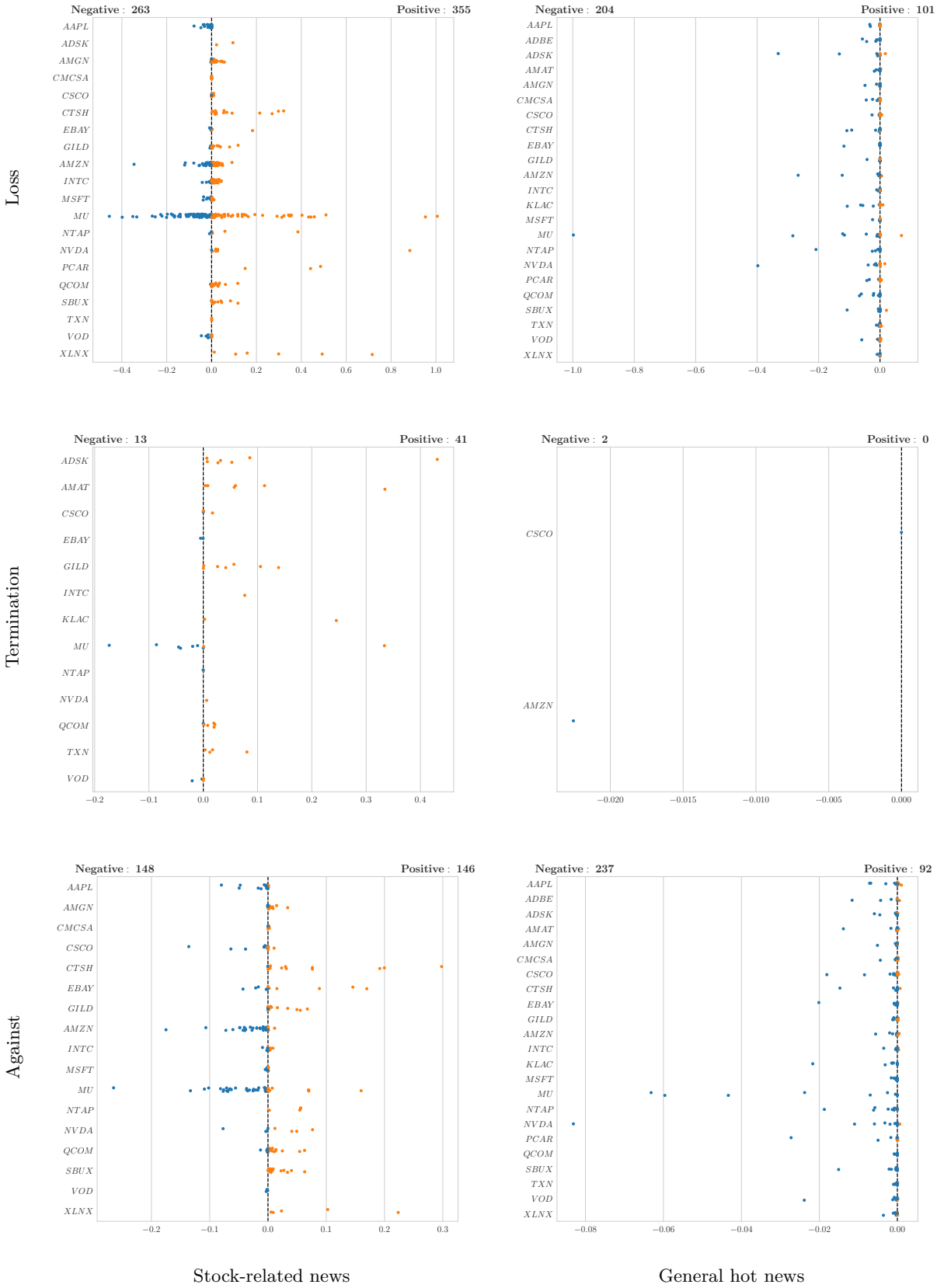


Figure 10: Explainer results for the top negative LM words

Notes: Rows represent the most repeated negative words in the LM dictionary. The left (right) column represents the SHAP values for stock-related (general hot) news published during the out-of-sample period. The x-axis is the reported explainer value, and the y-axis is the ticker name. The vertical line represents no impact on RV, while the left (right) hand side represents the negative (positive) impact of a specified group of words on RV. The total number of negative and positive SHAP values is depicted at the top for each figure.

dictionary approach will classify 'loss', 'termination', 'against', and their variations always as negative terms and possibly with a prediction that it always increases RV forecast by a fixed amount. A greater focus on stock-level results (y-axis) could produce interesting findings. For all plots in Figure 10, the presence of idiosyncratic behaviour among stocks is noticeable. For example, for the top left plot, in most appearances, 'loss' has a negative impact on RV for the AAPL ticker, but this changes to a mostly positive for the CTSH ticker. Also, as mentioned, there are some missing tickers in the plots, which may be because the specified word is not serving the model as an essential contributor to RV forecasting. Finally, perhaps the most interesting aspect of this news-based model is its flexibility in asset pricing. All results in Figure 10 will likely change if RV is switched with another dependent variable. However, for dictionary-based sentiment analysis, all appearances of a word have the same sentiment and importance regardless of the framework of analysis and the specified dependent variable.

So far, by providing a few examples, the news-based model, in conjunction with XAI, has shed light on the so-called black-box model and identified several advantages of this new approach. What is not yet clear is the global importance of n-grams among all analysed stocks over the out-of-sample period for both stock-related and general hot news. This not only provides some insight into the volatility movers in the textual news but also gauges how much these models are reliable. The primary RV movers are investigated separately for stock-related and general hot news in Subsection 6.2.

6.2 Volatility Movers

It has been augured that complex method for modelling textual data in finance potentially add more noise than signal (Loughran and McDonald, 2016). We believe signal-to-noise of ML models is reasonable if they can simultaneously improve performance and also generate readable and acceptable knowledge in finance by XAI or other similar approaches. Otherwise, harnessing these models for solely showing performance improvement is insufficient, at least for the finance domain. Therefore, this subsection focuses on generating fresh insight into the primary volatility movers in distinct stock-related and general hot news.

Table 9 and Table 10 provide the results obtained from the proposed XAI in Subsection 4.3 to identify the volatility movers over the entire test samples for all 23 stocks. For stock-related news, 13 is chosen as the top t repetitions to ensure that the reported volatility moves cover the important n-grams for more than half of the studied stocks. For general hot news, as discussed, the same news stories are used for all stocks; therefore, each top group contains substantially more n-grams. For clarity, 3 is chosen as the top t repetitions for general hot news.[1] What is important for us to recognise here is that numerical n-grams play a pivotal role in both obtained lists. This finding is expected and supports the importance of keeping the numerical values in our proposed preprocessing steps in Subsection 3.1. However, for clarity, the numerical n-grams are removed from Table 9 and Table 10. Also, there is no difference in importance among the n-grams in each class and among classes in these tables.

[1]Our further analysis shows that changing t does not substantially impact the proposed classification for volatility movers for both stock-related and general hot news. More specifically, for general hot news, increasing this value expands our knowledge about volatility movers, but the proposed classes remain roughly unchanged. This is also valid for stock-related news.

The stock-related volatility movers in Table 9 were obtained after carefully reviewing and finding the unique n-grams in the final list. By incorporating general financial knowledge, n-grams in stock-related news are classified as 'Analyst opinion', 'Event', 'Verb', 'Market', 'Abbreviation', 'Country/Company', 'Announcement', 'Numeric', 'Calendar', 'Insider', and 'Mixed'.[1] The key findings can be listed as follows: 'Analyst opinion' and 'Event' classes contain the majority of volatility mover n-grams. This is expected and clearly shows the importance of analyst opinions about the potential of stocks and news headlines about company events like earning calls and different financial reports. The appearance of verbs like 'registers', 'announces', 'files', 'raises', and 'surrenders' is also quite pivotal in this list. Moving to the next class, some market-related n-grams in the 'Market' class, like 'stocks to buy', 'premarket', and 'stock market opens', are discovered as influential volatility movers. The remaining classes, although still important, have a fewer number of n-grams inside. One of the noticeable findings about the rest of the classes is the appearance of 'China' as the only country in the volatility movers list, which shows the importance of its appearance even in stock-related news stories. Although it is not easy to justify the appearance of all n-grams in this table, it is undeniable that the mentioned classes can potentially play an essential role in changing volatility based on financial knowledge.

Table 10 presents the general hot news volatility movers. The final list of n-grams is classified as 'Person', 'Place', 'Legal entity', 'Level', 'Verb', 'Index', 'Data', 'Numeric', and 'Mixed'. This table is quite revealing in many ways. 'Person' class demonstrates American political faces, including Donald Trump[2], Barak Obama[3], Joe Biden[4], de Blasio[5], Meadows[6], McConnel[7], Pompeo[8], and Pelosi[9], Federal Reserve chair, president and CEOs, including Powell[10], Yellen[11], Bullard[12], Mester[13], Williams[14], Kashkari[15], and Bostic[16], and international political faces including Cummings[17] and Kim[18] are the primary volatility movers. The appearance of Cramer[19] as a television personality is also interesting but not surprising. However, the appearance of William G. Kaelin (an American Nobel Laureate physician-scientist) and Reinhard Genzel (a German astrophysicist) is not matched with the rest of the n-grams in this class. These two misleading n-grams show that similar to econometric models, ML models or XAI approaches are not error-free, and the results must be interpreted cautiously.

Further analysis of Table 10 shows the importance of the appearance of some places in our analysis. This finding is again expected and suggests the importance of this specific group of countries as

[1] It is important to stress that this is just a rough classification based on our opinion, but we believe moving some of these n-grams between these classes does not change the whole picture.

[2] Donald Trump is the 45th president of the United States.

[3] Barack Obama is the 44th president of the United States.

[4] Joe Biden is the 46th president of the United States.

[5] Bill de Blasio is the 109th mayor of New York City.

[6] Mark Meadows is the 29th White House chief of staff.

[7] Mitch McConnell is an American politician and attorney.

[8] Mike Pompeo is the 70th United States secretary of state.

[9] Nancy Pelosi is speaker of the United States House of Representatives.

[10] Jerome Powell is the 16th chair of the Federal Reserve.

[11] Janet Yellen is the 15th chair of the Federal Reserve.

[12] James B. Bullard is the 12th president of the Federal Reserve Bank of St. Louis.

[13] Loretta J. Mester is the president and CEO of the Federal Reserve Bank of Cleveland.

[14] John C. Williams is the president and chief executive officer of the Federal Reserve Bank of New York.

[15] Neel Kashkari is the 12th president and CEO of the Federal Reserve Bank of Atlanta.

[16] Raphael Bostic is the 15th president and CEO of the Federal Reserve Bank of Atlanta.

[17] Dominic Cummings is a British political strategist who served as Chief Adviser to British Prime Minister.

[18] Kim Jong-un is the supreme leader of North Korea.

[19] Jim Cramer is the host of Mad Money on CNBC.

Table 9: Volatility movers (stock-related news)

Analyst opinion	outperform by	price target	of earnings	sees #q adj	jumps	rtgs	*NUMBERS*[c]
equal-weight	neutral by	price target raised	earnings call	other events >	rises	chmn	**Calender**
overweight	equal-weight by	price target announced	earning call transcripts	filing >	sues	dir	week ended
neutral	equal-weigth from	price target cut	transcript	report	unveils	corp	week ended MONTH[d]
cut to neutral	overweight by	target announced	transcript, >	compensation filing	**Market**	shrs	ended MONTH
maintained at equal-weight	outperform from	target raised	events >	profit	stock market opens	yr	review for week
maintained at overweight	overweight from equal-weight	target annonced at	earning season	revenue	premarket	**Country/Company**	for week ended
maintained at outperform	overweight from neutral	announces at	earnings preview	**Verb**	stock falls	china	**Insider**
maintained at neutral	equal-weight from overweight	cut to hold	earnings tomorrow	sees	stock surges	goldman sachs	insider review
from neutral	perform from outperform	raised to neutral	earnings DAY	announces	stock soars	moody's	insider review for
from hold	to neutral	raised to outperform	u.s. eanings DAY	registers	stocks to watch	credit suice	insider sales
buy from neutral	to neutral from	raised to buy	u.s. earnings	appoints	stocks to buy	nasdaq	substantial insider sales
buy from hold	to outperform	outlook	earnings beat	assigns	to watch	cfa	**Mixed**
hold from buy	to outperform from	outlook stable	reports earnings tomorrow	soars	tech stocks	wsj	long-term
from equal-weight	neutral from	outlk	reports earnings	surrenders	chip stocks	**Announcement**	technology
from outperform	neutral from buy	analyst says	for u.s. earnings	release	morning movers	announces completion of	growth
from neutral by	neutral from overweight	from hold	conference (transcript)	update	morning report	moody's announces	sales
from overweight	outperform from	**Event**	holders, #q	acquires	on the street	agreement	sales: morning
at outperform	outferform from neutral	#[a]q	files 8k	backs	**Abbreviation**	definitive agreement	shares
at overweight	initiated at outperform	fourth quarter	13f	declares	inst	entry into definitive	correction
at overweight by	initiated at neutral	second quarter	dividend	files	inst holders	deal	price
at equal-weight	initiated at equal-weight	quarter	cash dividend	launches	inc	**Numeric**	stake
at neutral	overweight by keybanc	>[b]	rev	completes	mgmt	billion	results
at neutral by	overweight by morgan	#q, YEAR	eps	raises	exec	bln	new
at equal-weight by	equal-weight by morgan	holders #q	adj eps	boosts	exec mgmt	million	vs
at outperform by	outperform	earnings	#q adj eps	gains	changes exec mgmt	mln	

Notes: [a] '#' indicates number. [b] '>' commonly indicates quantities in news stories, especially the news headlines about financial reports and earning calls. [c] For clarity, the numerical n-grams are removed and replaced by 'NUMBERS' in this table. [d] 'MONTH' indicates different months.

Table 10: Volatility movers (general hot news)

Person	saudis	fall #%	% rate	ranks	payroll-tax cut	vacancies
trump (ref. Donald Trump)[a]	japan	fell #%	% food	correct	shutdown	problems
donald (ref. Donald Trump)	eu	raising #%	high	win	offering	major
obama (ref. Barack Obama)	u.k	above $[c]	still high	update	speech	minor
joe biden (ref. Joe Biden)	spain	at $	higher	search	trade speech	source
biden (ref. Joe Biden)	asia	in $	min	influence	hearing	week (wk)
powell (ref. Jerome Powell)	north korea	of $	dip	becoming	us crude	global
yellen (ref. Janet Yellen)	**Legal entity**	on $	up	**Index**	gold	approval
de blasio (ref. Bill de Blasio)	gop (ref. Republican Party)	than $	down	s&p500 down	crisis	leverage
cummings (ref. Dominic Cummings)	ism (ref. Institute of Supply Management)	from $	low	s&p500 falls	coalitions	standard
bullard (ref. James B. Bullard)	doe (ref. Department of Energy)	to $	least	s&p500 drops	airstrike	tremendous
cramer (ref. Jim Cramer)	opec	up to $	**Verb**	s&p500 gains	shifting coalitions	obstruction
mester (ref. Loretta J. Mester)	fed (ref. Federal Reserve)	around $	seen	s&p500 rises	campaign staffer	analysis
meadows (ref. Mark Meadows)	omb (ref. Office of Management and Budget)	by #	sees	s&p500 up	law	statement
williams (ref. John C. Williams)	health organization	% over	forces	s&p500 adds	lawmakers	sources
mcconnell (ref. Mitch McConnell)	u.s. treasury	% target	achieve	s&p500 climbs	attorney general	adversity sources
pompeo (ref. Mike Pompeo)	treasury	% from	cut	s&p500	trump adminstation	advisor
pelosi (ref. Nancy Pelosi)	cdc (ref. Centers for Disease Control and Prevention)	% rate	grows	**Data**	compensation	emails
kim (ref. Kim Jong-un)	occ (ref. Options Clearing Corporation)	% vs	resigns	ex-autos (ref. Retail Sales ex Autos)	banks	state tv
kashkari (ref. Neel Kashkari)	sec (ref. Securities and Exchange Commission)	% through	says	inflation	economy	companies
bostic (ref. Raphael Bostic)	wsj (ref. The Wall Street Journal)	about $	charges	jobs	police	opioid companies
g kaelin (ref. William G. Kaelin)	st. louis (ref. Federal Reserve Bank of St. Louis)	under $	dies	gdp	dodd_frank	relationships
genzel (ref. Reinhard Genzel)	bloomberg	for $	passes	deficit	rico's	sru_related problems
Place	wolfe (ref. Wolfe Research)	yr to $	propose	**Numeric**	corridosrs	first
us	**Level**	% on $	releases	trillion	officials	chance
korea	below #[b] %	% to $	triggers	billion	organization	groups
syria	above #%	% at #	proceed	million	pm	house
israel	to #%	% in	totaled	mln bbl	program	direction
china	up #%	% on	transferred	*NUMBERS*[d]	schools	dreamers'
iran	achieve #%	% to	pass	**Mixed**	prison	others
russia	slide #%	% on year	talk	spac	groups	speaker

Notes: [a] The probable complete phrase the specified n-gram is referring to. [b] '#' indicates number. [c] '$' indicates the amount of money in the dollar.
[d] For clarity, the numerical n-grams are removed and replaced by 'NUMBERS' in this table.

volatility movers. Although our textual data is primarily targeting US companies, what is interesting is the appearance of 'China' in both Table 9 and Table 10. If we turn to 'Legal entity', it covers a variety of major offices, departments, commissions, and companies. The appearance of the Health Organization and CDC (Centers for Disease Control and Prevention) could be attributed to the COVID-19 pandemic. Again, although nearly all n-grams in this class are justifiable, the appearance of the Wolfe (ref. Wolfe Research) is not in line with the rest, and it could be due to the model or XAI error. The next major class with a high number of n-grams is 'Level', covering a variety of levels and changes. From this class, as expected, it is undeniable that the appearance of percentages or currency values and some specific words like 'below', 'above', 'fall', and 'under' in headlines play an essential role in volatility movements.

Moving to the next classes, similar to the stock-related news, the 'Verb' and 'Numeric' classes in general hot news demonstrate the importance of this specific group of verbs and also numbers as volatility movers. Last but not least, the 'Mixed' class includes a variety of n-grams like 'SPAC'[1], 'Payroll-tax cut', 'Airstrike', 'Shutdown', 'Coalitions', 'Crisis', 'Trade Speech', 'Attorney General', and 'Hearing'. Detailed analysis of these n-grams is beyond the scope of this study, but we believe not all, but most, are in accordance with the expectations. Finally, the 'Index' and 'Data' classes reveal the importance of changes in the S&P500 index and financial and economic data like inflation, GDP, and deficit as important n-grams.

The evidence presented in this section provides a framework for in-depth and transparent analysis of ML models for textual analysis in finance and consistently points towards the specific classes of n-grams as the primary volatility movers in both stock-related and general hot news stories. Because of reachable ultimate transparency by dictionary-based approaches and their simple structure, XAI approaches in ML, at least until now, are not a replacement for dictionaries. But we believe the discovered information is valuable for improving our understanding of NLP in asset pricing.

7 Robustness Checks

In this section, we assess the impact of model parameters and forecasting structures on the RV forecasting performance we have reported so far. The bar chart in Figure 11 shows the percentage of tickers (among 23 stocks in this study) with the outstanding performance considering the MSE (QLIKE) loss function in the left (right) plots at the 5% significance level of the RC compared to all HAR-family of models as the benchmark. The top, middle, and bottom rows show the results for the full out-of-sample period, normal volatility days, and high volatility days, respectively. 25, 50, 75, and 100 are the number of filters for each group of models covering different model complexities from the lowest complexity (25 filters) to the highest one (100 filters).[2]

Two benchmark groups and five different variations are covered for robustness checks. 'Skip−gram' and 'FT/Skip−gram' benchmark groups are the best-performing stock-related models in Subsection 5.1 incorporating our developed *FinText* word embedding as the embedding layer. In view of all that has been mentioned so far, one may ask whether adding more news from past days improve

[1]Special-purpose acquisition company.

[2]Due to computational limitations, robustness checks are only presented for stock-related news.

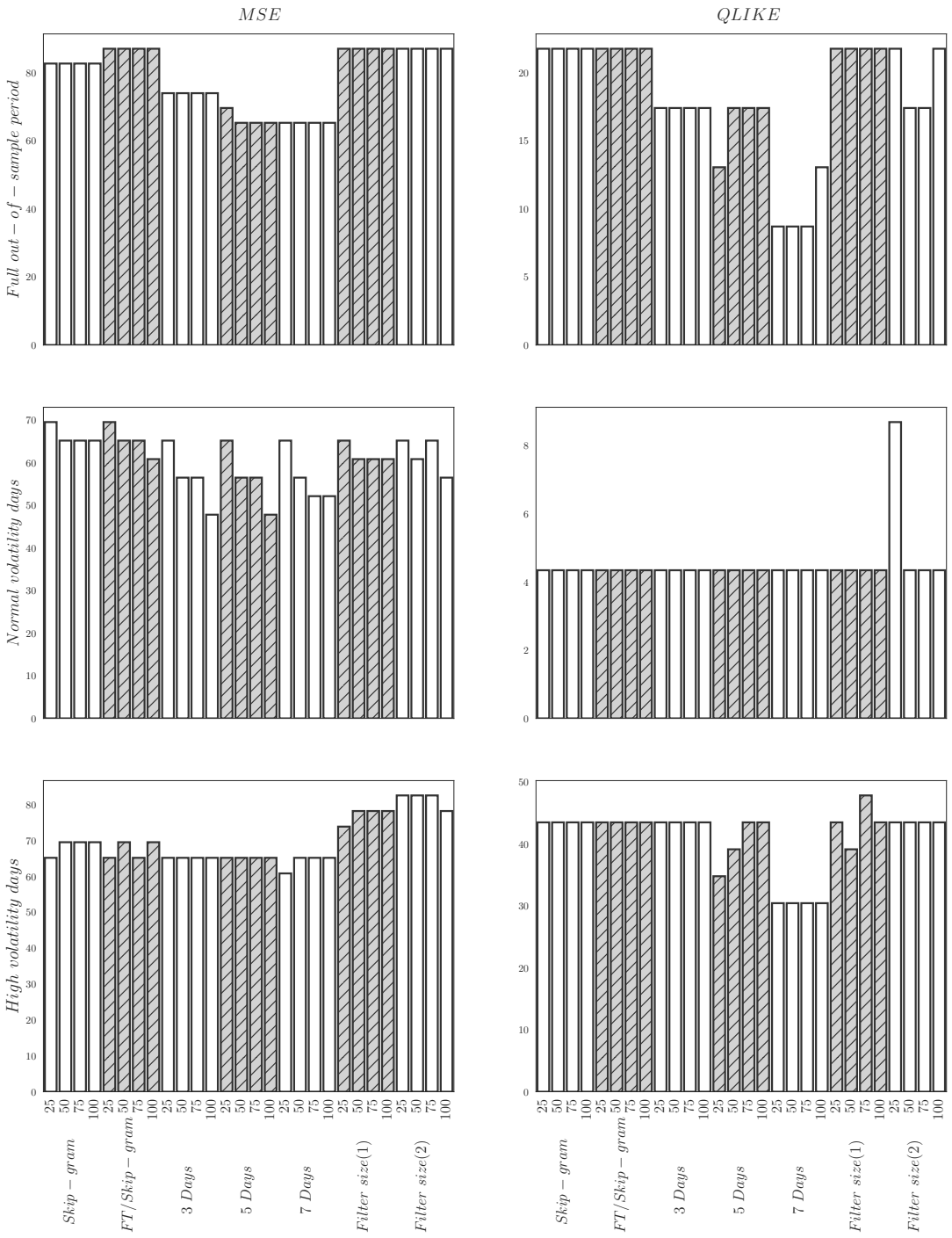


Figure 11: RC robustness checks (stock-related news)

Notes: The bar chart is the percentage of tickers (among 23 stocks in this study) with the outstanding performance considering the MSE (QLIKE) loss function in the left (right) plots at the 5% significance level of the RC compared to all HAR-family of models as the benchmark for stock-related news. The top, middle, and bottom rows show the results for the full out-of-sample period, normal volatility days, and high volatility days, respectively. 25, 50, 75, and 100 are the number of filters for each group of models.

the RV forecasting performance or not. To answer this question, considering the 'FT/Skip−gram' as the base model, the '3 days' group changes the input from news headlines of the previous day to the previous three days. Similarly, the '5 days' and '7 days' groups take as input the headlines from the previous 5 and 7 days, respectively. Moreover, as discussed in Subsection 4.2, the reason for choosing this specific set of filter sizes is that 1, 2, and 3 are equivalent to unigram, bigram, and trigram. What is also not clear is the impact of longer n-grams in RV forecasting performance. Therefore, the third ('Filter size(1)') and fourth ('Filter size(2)') modifications change the filter size to 4, 5 and 6, and 7, 8, and 9, respectively.

Figure 11 reveals that moving to '3 days', '5 days' and '7 days' input duration causes degradation in forecasting performance in full out-of-sample period for both MSE and QLIKE loss functions. Comparing the normal volatility days and high volatility days reveals that for the MSE loss function, this degradation is mainly caused by the degradation in forecasting performance of normal volatility days. However, for the QLIKE loss function, the degradation in forecasting performance of high volatility days is the major contributor. It is likely that the market already assimilates the information in older news stories; therefore, incorporating information in the past 3, 5, and 7 days degrades the RV forecasting performance due to the extra noise it adds to the fresh news. Let us now model longer combinations of terms by increasing the filter size values. Although this modification does not change the performance for the full out-of-sample period, some minor improvement in high volatility days is noticeable, mainly for the MSE loss function.

The robustness tests presented thus far provide evidence that increasing the complexity of models by changing the filter size values or feeding more news to the model by adding past days degrades the performance or at least does not substantially improve it. Therefore, the simple proposed model and forecasting structure in Subsection 4.2 can be considered as a preliminary point for future studies.

8 Conclusions

In this paper, we developed a financial word embedding called *FinText*[1] using Word2Vec and FastText algorithms covering around 15 years of news stories from 2000 to 2015. Our financial word embedding performed less well on general-purpose benchmarks when compared with Google's and Facebook's word embeddings. However, when challenged with detecting unique financial relationships, *FinText* is better and more sensitive in detecting financial jargon. Also, for the first time, we proposed a gold-standard financial benchmark. This financial benchmark contains 2660 unique analogies for testing natural language processing models in finance. We showed that the best-performing *FinText* model reaches around 14 and 55 times better accuracies than Google's and Facebook's word embeddings, respectively.

Our next goal in this study was to test these pre-trained word embeddings in an asset pricing framework. The literature on volatility forecasting has revealed that news is a potential contributor to volatility; therefore, realised volatility forecasting was chosen for this analysis. Using data for 23 NASDAQ stocks from 27 July 2007 to 27 January 2022 and stock-related news, we found evidence

[1] *FinText* word embeddings are available for download from FinText.ai.

that our proposed word embedding performs better, especially for high volatility days, although it is developed by a substantially smaller textual corpus compared with Google's and Facebook's word embeddings. This finding draws our attention to the importance of the corpus and its field-specific quality for training word embeddings and, generally, natural language processing models in finance. We also observed an evident degradation in forecasting performance during the out-of-sample period from 2015 to 2022, reaching the highest amount during the COVID-19 outbreak. By monitoring the out-of-vocabulary terms over time, we concluded that because word embeddings are trained by a specific corpus covering a limited time horizon, it is generally expected to observe a decrease in performance over time, especially during COVID-19, when new terminologies appear in the news.

Moving to general news, an entirely distinct source of news stories mainly focusing on major economic, financial, political, and geopolitical events, the forecasting power switched from high volatility days to normal volatility days for all models reaching the highest amount of improvement for Google's and Facebook's word embeddings. This improvement can be explained, at least in part, by the size and mixture of the corpora because both contain substantially more extensive and diverse textual data inside. The investigation of ensemble models by mixing financial and textual data identified the importance of the information content of both financial numbers and textual news simultaneously for volatility forecasting. Therefore, news stories' information can potentially improve forecasting performance, but it is not a replacement for numerical financial data.

We used Explainable AI to measure the impact of n-grams on realised volatility forecasts. We also identified the global importance of n-grams among all analysed stocks over the out-of-sample period for both stock-related and general news. For stock-related news, we identified specific classes of n-grams like analyst opinions, company events, numbers, and announcements as the volatility movers. This changed to specific classes like person names, places, and legal entities for general news. Discovering such clear and in-depth information about the classes of n-grams attributing the most in volatility forecasting is not feasible in the classical dictionary-based approaches in finance. Finally, The robustness tests showed that increasing the complexity of models or feeding more news to the model degrades the performance or at least does not substantially improve it.

In conclusion, we demonstrated that our purpose-built financial word embedding is more knowledgeable in finance. Also, by introducing a simple machine learning framework, we showed its potential to improve the volatility forecasting performance of well-known econometric models. Because of the simplicity and transparency of dictionary-based approaches, more complex natural language processing models in asset pricing, at least until now, can not be considered a replacement for dictionaries. Nevertheless, we demonstrated that the in-depth discovered information is valuable for improving our understanding of textual analysis in finance. We hope that this study paves the way for other state-of-the-art natural language processing and machine learning research in different financial contexts.

References

Adämmer, P. and R. A. Schüssler (2020). Forecasting the equity premium: mind the news! *Review of Finance 24*(6), 1313–1355.

Agirre, E., E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa (2009). A study on similarity and relatedness using distributional and wordnet-based approaches.

Ban, G.-Y., N. El Karoui, and A. E. Lim (2018). Machine learning and portfolio optimization. *Management Science 64*(3), 1136–1154.

Bianchi, D., M. Büchner, and A. Tamoni (2021). Bond risk premiums with machine learning. *The Review of Financial Studies 34*(2), 1046–1089.

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics 5*, 135–146.

Bollerslev, T., A. J. Patton, and R. Quaedvlieg (2016). Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics 192*(1), 1–18.

Bubna, A., S. R. Das, and N. Prabhala (2020). Venture capital communities. *Journal of Financial and Quantitative Analysis 55*(2), 621–651.

Bybee, L., B. T. Kelly, A. Manela, and D. Xiu (2020). The structure of economic news. Technical report, National Bureau of Economic Research.

Chen, L., M. Pelger, and J. Zhu (2020). Deep learning in asset pricing. *Available at SSRN 3350138*.

Christensen, K., M. Siggaard, and B. Veliyev (2021). A machine learning approach to volatility forecasting. *Available at SSRN*.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011). Natural language processing (almost) from scratch. *Journal of machine learning research 12*(ARTICLE), 2493–2537.

Cong, L. W., T. Liang, and X. Zhang (2019). Textual factors: A scalable, interpretable, and data-driven approach to analyzing unstructured information. *Interpretable, and Data-driven Approach to Analyzing Unstructured Information (September 1, 2019)*.

Conrad, C. and R. F. Engle (2021). Modelling volatility cycles: the (mf)^ 2 garch model. *Available at SSRN*.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics 7*(2), 174–196.

Corsi, F. and R. Reno (2009). Har volatility modelling with heterogeneous leverage and jumps. *Available at SSRN 1316953*.

Engle, R. F. and S. Martins (2020). Measuring and hedging geopolitical risk.

Engle, R. F. and V. K. Ng (1993). Measuring and testing the impact of news on volatility. *The journal of finance 48*(5), 1749–1778.

Gentzkow, M., B. Kelly, and M. Taddy (2019). Text as data. *Journal of Economic Literature 57*(3), 535–74.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

Gu, S., B. Kelly, and D. Xiu (2021). Autoencoder asset pricing models. *Journal of Econometrics 222*(1), 429–450.

Hill, F., R. Reichart, and A. Korhonen (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics 41*(4), 665–695.

Jiang, J., B. T. Kelly, and D. Xiu (2020). (re-) imag (in) ing price trends. *Chicago Booth Research Paper* (21-01).

Joulin, A., E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov (2016). Fasttext.zip: Compressing text classification models.

Ke, Z. T., B. T. Kelly, and D. Xiu (2019). Predicting returns with text data. Technical report, National Bureau of Economic Research.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR abs/1408.5882*.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, K., X. Liu, F. Mai, and T. Zhang (2020). The role of corporate culture in bad times: Evidence from the covid-19 pandemic. *Journal of Financial and Quantitative Analysis*, 1–68.

Lim, B., S. Zohren, and S. Roberts (2019). Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science 1*(4), 19–38.

Loughran, T. and B. McDonald (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance 66*(1), 35–65.

Loughran, T. and B. McDonald (2016). Textual analysis in accounting and finance: A survey. *Journal of Accounting Research 54*(4), 1187–1230.

Lundberg, S. and S.-I. Lee (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

Morin, F. and Y. Bengio (2005). Hierarchical probabilistic neural network language model. In *Aistats*, Volume 5, pp. 246–252. Citeseer.

Obaid, K. and K. Pukthuanthong (2021). A picture is worth a thousand words: Measuring investor sentiment by combining machine learning and photos from news. *Journal of Financial Economics*.

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics 160*(1), 246–256.

Patton, A. J. and K. Sheppard (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics and Statistics 97*(3), 683–697.

Poh, D., B. Lim, S. Zohren, and S. Roberts (2021). Building cross-sectional systematic strategies by learning to rank. *The Journal of Financial Data Science 3*(2), 70–86.

Politis, D. N. and J. P. Romano (1994). The stationary bootstrap. *Journal of the American Statistical association 89*(428), 1303–1313.

Rahimikia, E. and S.-H. Poon (2020a). Big data approach to realised volatility forecasting using har model augmented with limit order book and news. *Available at SSRN 3684040*.

Rahimikia, E. and S.-H. Poon (2020b). Machine learning for realised volatility forecasting. *Available at SSRN 3707796*.

RoyChowdhury, A., P. Sharma, E. Learned-Miller, and A. Roy (2017). Reducing duplicate filters in deep neural networks. In *NIPS workshop on Deep Learning: Bridging Theory and Practice*, Volume 1, pp. 1.

Shapiro, A. H., M. Sudhof, and D. J. Wilson (2020). Measuring news sentiment. *Journal of Econometrics*.

Shrikumar, A., P. Greenside, and A. Kundaje (2017). Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pp. 3145–3153. PMLR.

Sirignano, J. and R. Cont (2019). Universal features of price formation in financial markets: perspectives from deep learning. *Quantitative Finance 19*(9), 1449–1459.

Wood, K., S. Roberts, and S. Zohren (2021). Slow momentum with fast reversion: A trading strategy using deep learning and changepoint detection. *arXiv preprint arXiv:2105.13727*.

Wu, W., J. Chen, Z. Yang, and M. L. Tindall (2020). A cross-sectional machine learning approach for hedge fund return prediction and selection. *Management Science*.

Zhang, Z. and S. Zohren (2021). Multi-horizon forecasting for limit order books: Novel deep learning approaches and hardware acceleration using intelligent processing units. *arXiv preprint arXiv:2105.10430*.

Zhang, Z., S. Zohren, and S. Roberts (2018). Bdlob: Bayesian deep convolutional neural networks for limit order books. *arXiv preprint arXiv:1811.10041*.

Zhang, Z., S. Zohren, and S. Roberts (2020). Deep reinforcement learning for trading. *The Journal of Financial Data Science 2*(2), 25–40.

Zhao, W., T. Joshi, V. N. Nair, and A. Sudjianto (2020). Shap values for explaining cnn-based text classification models. *arXiv preprint arXiv:2008.11825*.

Table A1: Textual data cleaning rules

Primary	
Extracting body of news from XML	Removing XML-Encoding Characters (XMLENCOD)
Converting XML to text (parsing)	Converting uppercase letters to lowercase letters
Removing tables	
Begins with	
(END) XX	(email|e-mail): XX
for (more|further) (information|from marketwatch), please visit: XX	(phone|fax|contact|dgap-ad-hoc|dgap-news): XX
(EMAIL; @XX)	image available: XX
copyright XXXX, XX URL	source: XX
(more to follow) XX	to read more, visit: XX
end of (message|corporate news) XX	(view source|view original content) (with|on) XX
source: XX URL	(investor relations|investor contact) XX
XX contributed to this article XX	like us on XX
view source version on XX	(copyright|(c)|©) XX
(=————————————————————)	XX can be found at URL XX
view original content with multimedia XX	by dow jones newswires XX
readers can alert XX	(write to|follow) XX at EMAIL
view original content: XX	(phone|tel|telephine|mobile|contact|inquiries|comment): XX
media inquiries: XX	(contact information|media contact|contact client services|internet) XX
readers: send feedback to XX	click here to subscribe to XX
follow us on XX	to learn more about XX
contact(s): XX	(website|web site): URL XX
please refer to URL XX	contact us in XX
find out more at URL XX	to receive news releases by (e-mail|email) XX
XX enquiries: XX	full story at XX
-by XX, dow jones newswires XX	
Ends with	
(more to follow)	view original content XX:
(fax|tell|contact|dgap-ad-hoc|dgap-news):	(contacts|web site):
ratings actions from baystreet:	(=|- -|_| · |-)
cannot parse story	for notes, kindly refer
lipper indexes:to subscribe to	following is the related link:
for full details, please click on	
General	
(linkedin|facebook|fb): XX	(URL (and|&) XX)
(twitter|ig): XX	(EMAIL (and|&) EMAIL)
(attachment|attachments): XX	this information was brought to you by XX
please visit XX	write to EMAIL
follow us on XX	to receive our XX URL
All rights reserved	more at, XX URL
Final checks	
Removing links and emails	Removing short news (lower than 25 characters)
Removing both the leading and the trailing space(s)	Removing phone numbers