HTML: Hierarchical Transformer-based Multi-task Learning for Volatility Prediction

Linyi Yang
Insight Centre for Data Analytics
University College Dublin
Dublin, Ireland
linyi.yang@insight-centre.org

Tin Lok James Ng
University of Wollongong
Austrilia
jamesng@uow.edu.au

Barry Smyth
Insight Centre for Data Analytics
University College Dublin
Dublin, Ireland
barry.smyth@insight-centre.org

Ruihai Dong
Insight Centre for Data Analytics
University College Dublin
Dublin, Ireland
ruihai.dong@insight-centre.org

ABSTRACT

The *volatility forecasting* task refers to predicting the amount of variability in the price of a financial asset over a certain period. It is an important mechanism for evaluating the risk associated with an asset and, as such, is of significant theoretical and practical importance in financial analysis. While classical approaches have framed this task as a time-series prediction one – using historical pricing as a guide to future risk forecasting – recent advances in natural language processing have seen researchers turn to complementary sources of data, such as analyst reports, social media, and even the audio data from earnings calls. This paper proposes a novel hierarchical, transformer, multi-task architecture designed to harness the text and audio data from quarterly earnings conference calls to predict future price volatility in the short and long term. This includes a comprehensive comparison to a variety of baselines, which demonstrates very significant improvements in prediction accuracy, in the range 17% - 49% compared to the current state-of-the-art. In addition, we describe the results of an ablation study to evaluate the relative contributions of each component of our approach and the relative contributions of text and audio data with respect to prediction accuracy.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Natural language processing**; **Supervised learning by regression**; **Multi-task learning**; • **Information systems** → *Multimedia information systems.*

KEYWORDS

Volatility forecasting, Hierarchical transformer, Multi-task learning

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380128

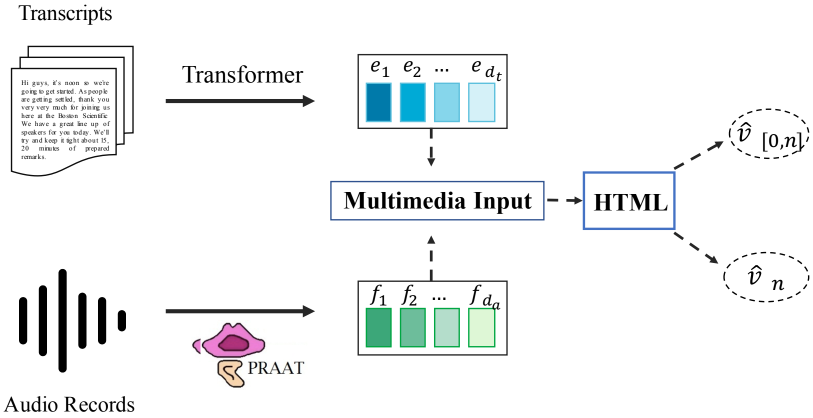
ACM Reference Format:
Linyi Yang, Tin Lok James Ng, Barry Smyth, and Ruihai Dong. 2020. HTML: Hierarchical Transformer-based Multi-task Learning for Volatility Prediction. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380128

1 INTRODUCTION

Predicting how the degree of variability in the price of a financial asset will vary over a certain period – the so-called *volatility* of the asset – is an important financial analysis task. Price volatility is generally considered to be a useful proxy for the level of risk associated with an asset and thus it plays an important role in assessing financial market risk and the pricing of financial derivatives. As a result, developing effective techniques for predicting price volatility has become increasingly important among academics and practitioners.

To a large extent, past research efforts have focused on the use of time-series modeling and prediction techniques using historical pricing data [33, 41, 66]. However, with recent advances in natural language processing (NLP) it has become possible to harness novel sources of data – from unstructured textual data in the form of financial news [16, 63, 65] and financial reports [23, 32, 47], to real-time social media [7, 43, 60, 62] – during the prediction process. Of particular relevance to this work is the information contained in *earnings call* transcripts [30, 46, 56], which typically accompany the earnings reports of publicly traded companies. Generally speaking, these are conference-calls, in which company executives discuss the latest results, offer guidance on their expectations for the coming year, and provide investors and analysts with an opportunity to ask questions. The information conveyed during conference call, and particularly the subsequent question-answer session with investors and analysts, can provide new information (Q&A parts is not well prepared by executives) into the current state of the company and it's future prospects, which, in turn, change investor perception of firm risk (price volatility). Indeed, recent work has shown how not only the text of the call can be useful [5, 25, 34], but also the vocal content and features contained within the call audio [24, 42, 46].

This work seeks to build on this recent research to further explore the utility of including textual and audio data from earnings calls for volatility forecasting. The overview of the proposed method is



Hi guys, it's noon so we're going to get started. As people are getting settled, thank you very very much for joining us here at the Boston Scientific We have a great line up of speakers for you today. We'll try and keep it tight about 15, 20 minutes of prepared remarks.

Figure 1: An overview of the proposed framework. For a given earnings call, the text and audio features are extracted from transcripts and audio records, respectively, and the resulting features are used as input features for a mult-task learner.

described in Figure 1. The primary technical contributions include a description and evaluation of a novel, deep-learning architecture for this task: Figure 2 presents our Hierarchical, Transformer-based, Multi-Task (HTML) model which combines a hierarchical, transformer [55] with multi-task learning [39]. Hierarchical transformer models [14, 37] have proven useful in many sequence-to-sequence learning tasks, including machine translation and text summarisation, and the approach is used here to extract the text features from call transcripts that will be used as inputs to the multi-task learner. Following [46], we extract 27 vocal features including pitch, intensity, jitter, and the harmonic to noise ratio using Praat [6]. The audio and text features are combined in the information fusion layer to provide input features for the multi-task learner. Multi-task learning – exploiting similarities and differences between related, simultaneous learning tasks – is used because it has proven to be successful when it comes to controlling for overfitting and improving generalisation [10], and here we simultaneously learn models to predict: (1) average *n-day* volatility (that is, the volatility of the following *n* days); and (2) single-day volatility (that is, the volatility on a single day, n-days in the future).

The remainder of this paper is organized as follows. In the next section we summarise relevant related work focusing in particular on the volatility prediction task, hierarchical model, multi-task learning, and multimedia information fusion. Section 3 presents a problem formulation before describing our proposed approach in Section 4 in detail. Before concluding, in Sections 5 and 6 we presents the results of a detailed evaluation using a benchmark dataset and in comparison to a number of state-of-the-art baseline techniques. The results of this evaluation demonstrate clear and significant prediction accuracy benefits accruing to our proposed approach, accuracy improvements in the range 17% - 49% compared to the current-state-of-the-art. Moreover, a detailed ablation study further clarifies the relative contributions of each model component and data source to overall prediction accuracy. We believe that these results establish this as a new performance benchmark for volatility forecasting.

2 RELATED WORK

This paper brings together a number of different ideas – volatility prediction, hierarchical model, multi-task learning, and multimedia information fusion – and in what follows we briefly summarise the relevant state-of-the-art in each of these areas, as it relates to the present work.

2.1 Volatility Prediction

Volatility modeling and prediction is of interest to researchers because of its theoretical importance and its practical applications. Conventional approaches [33, 41, 66] rely on historical pricing data and typically use continuous time-series models (local and stochastic volatility [12, 22, 26, 28, 49, 50]) and discrete time-series models (e.g. GARCH models [8, 17]).

Recently, research attention has focused on additional sources of volatility information. Significant improvements in NLP methods, many applied to sources of financial information [7, 15, 16, 23, 32, 43, 47, 60, 62, 65], demonstrate how mining financial news, analyst reports, earnings reports, and social media has the potential to improve many financial prediction tasks by harnessing powerful new features that are absent from traditional time-series data. Moreover, features derived from the audio features of earnings calls have also proven to be useful for volatility prediction. For example, [46] incorporates CEO's vocal features, such as emotions and voice tones in earnings conference calls in predicting volatility of a stock using a multimodal deep regression model. By modeling the textual and vocal information contained in a conference call, resulting in a substantial improvement in volatility prediction accuracy, compared to classical methods.

The work in [46] is especially relevant in this context of this work as it provides a starting point for this work, and the best available baseline against which to evaluate our progress. We argue that the model proposed by [46] does not sufficiently investigate the power of both verbal and vocal information and that it fails to fully exploit the interaction between the text and audio information. The improvements derive from three aspects. First, we show how enriched textual and audio data can be extracted from call data using co-evolutionary methods. Second, we demonstrate how the use of a pre-trained language model and hierarchical features can greatly improve the representations used for learning and prediction. Finally, a key novelty of the present work stems from the way in which textual and audio features are integrated for multi-task learning, which, as we shall see later, leads to significant prediction benefits.

2.2 Hierarchical, Multi-Task Learning

Hierarchical learning techniques have recently proved to be successful across a variety of NLP tasks. Hierarchical attention networks were first proposed by [64], as a way to generate richer and more powerful natural language representations, and since then they have been applied to good effect in tasks such as document classification, relation extraction, and machine translation. More recently, further technical improvements in hierarchical architectures based on transformers have been developed for tasks such as automatic text summarization [18, 37]. This suggests that similar techniques might prove to be useful when it comes to extracting textual feature from earnings call transcripts.

Multi-task learners solve multiple learning tasks at the same time, by exploiting commonalities and differences between the tasks, to provide an effective set of learning constraints that have been shown to reduce the risk of overfitting, improve generalize ability, and overall improve the effectiveness of the learned models compared to the models produced by single-short learners using the same training data. Multi-task learning has shown particular promise in NLP [51, 58, 61] and speech recognition [13, 52] tasks. And the idea of combining hierarchical and multi-task learning, by using a hierarchical framework consisting of several relevant tasks as a joint multi-task learning model, was first proposed by [21]; see also the work of [48] on the use of a hierarchical architecture for learning word embeddings from semantic NLP tasks.

In this paper we propose a hierarchical, multi-task learning approach consisting of two financial forecasting tasks. The primary task involves predicting asset volatility over a given time period (number of days), while our secondary task involves predicting asset volatility for a single day. Our intuition is that this multi-task learning framework will improve prediction performance by reducing the representation bias of our model, and, to the best of our knowledge, this is the first time that a hierarchical, multi-task transformer has been used for volatility prediction.

2.3 Multi-modal Information Fusion

In this work we focus on learning from different types of data – text and audio – which has often proven challenging in the past because of the challenges associated with combining fundamentally different features. However, recent progress in deep learning research has led to significant improvement in similar multi-modal learning tasks, whereby high-level embeddings from different types of data are integrated via a deep neural network [40]. For instance, the Vision-and-Language BERT (ViLBERT) [38] learns task-agnostic joint representations of image and natural language content. Elsewhere, related ideas have been used to combine text and image information for multi-modal review generation [53]. And the interaction between text and audio data in a multi-modal learning framework has been the subject of recent studies in speech communication, in which acoustic features have been shown to be highly correlated with emotion [2], trustworthiness [4], and confidence [27].

To date the use of audio data sources has been all but absent from financial applications, with the exception of [46]. Given the effectiveness of recent multi-modal approaches, and the availability of task-relevant text and audio data for volatility forecasting, it is clear that these techniques warrant further consideration, hence the approach is taken in the present work.

3 MEASURING ASSET VOLATILITY

We formulate the volatility forecasting problem as a multivariate regression task, with textual and audio data as raw inputs, and an *n-day volatility predictions* (that is the predicted average volatility over the following n days) and *single-day volatility prediction for day-n* as the dual prediction outputs.

Following [32, 46, 47], we use log volatility [35, 36] as our basic measure of average n-day volatility; see Equation 1.

$$v_{[0,n]} = \ln\left(\sqrt{\frac{\sum_{i=1}^{n}(r_i - \overline{r})^2}{n}}\right) \quad (1)$$

In Equation 1, r_i is the stock return on day i and $\overline{r}$ is the average stock return in a window of n days. The return is defined as $r_i = (P_i - P_{i-1})/P_{i-1}$, where P_i is the adjusted closing price of a stock on day i.

The single day log volatility is estimated by the daily log absolute return, as in 2, where v_n can also be considered a noisy proxy of log volatility [9].

$$v_n = \ln\left(\left|\frac{P_n - P_{n-1}}{P_{n-1}}\right|\right) \quad (2)$$

Our multi-task learning objective is to simultaneously predict these two quantities $v_{[0,n]}$ and v_n using our input data; predicting $v_{[0,n]}$ is our main task, while predicting v_n is our auxiliary task.

4 DETAILED IMPLEMENTATIONS

Figure 2 summarizes the proposed HTML model which contains four components: (1) *token-level transformer encoder*; (2) *multimedia information fusion*; (3) *sentence-level transformer encoder*; and (4) *multi-task prediction*. Briefly, to begin with, text and audio features are extracted from the raw text/audio call content: text tokens are extracted from the text data and encoded into a vector using a pre-trained language model, while a range of 27 different audio features are extracted from the audio data using Praat [6], based on the sentence-level audio clips, and in line with the approach described by [46]. The resulting text and audio features are combined by the information fusion layer and used as input for the sentence-level transformer encoder to generate a new intermediate, multimodal representation to act as the input representation for the multi-task learner. The multi-task prediction layer generates average and single-day volatility prediction based on the inputs from the sentence-level transformer encoder. A more detailed implement implementation description of each of these components is presented in the following.

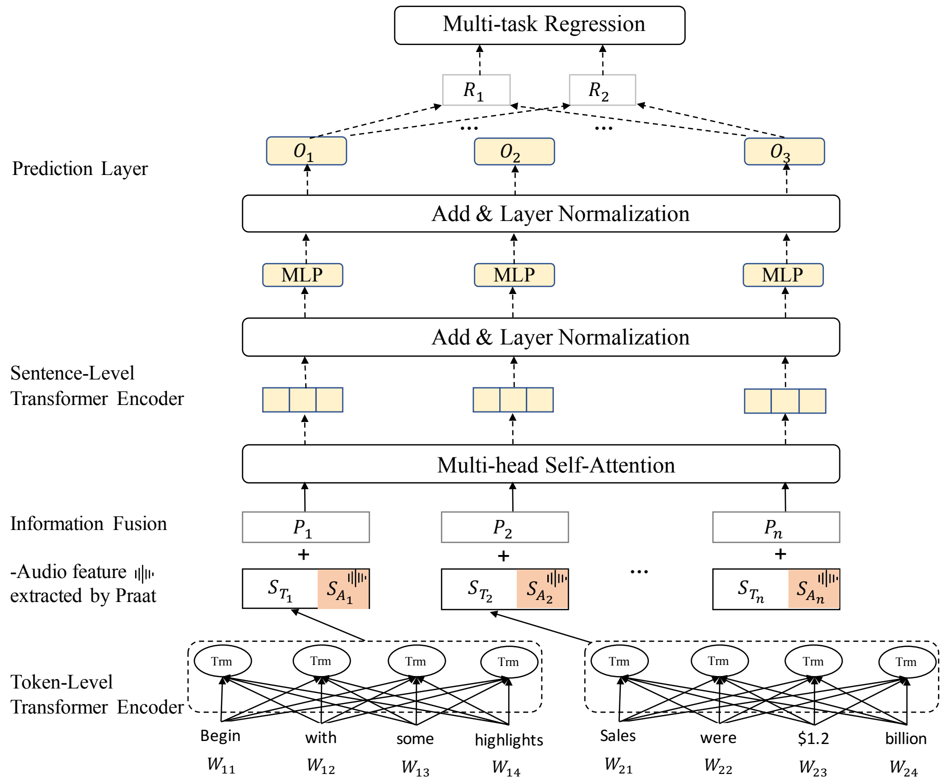


Figure 2: Hierarchical Transformer-based Multi-task Learning

4.1 Token-level Transformer Encoder

The token-level transformer encoder consists of a *Multi-Head Self-Attention Mechanism*, a *Residual Connections and Layer Normalization Layer*, a *Feed Forward Layer*, and a *Residual Connections and Layer Normalization Layer* [55]. The training of the encoder involves two phases, namely pre-training and fine-tuning. The pre-training phase can be considered as a self-supervised step and is performed using the Whole Word Masking BERT (WWM-BERT)[14] where WordPiece tokens belong to same word are masked jointly. The WWM-BERT mitigates the drawbacks of the original implementation of BERT whereby it explicitly forces the model to predict a whole word instead of WordPiece tokens in the training task. The find-tuning phase gently adjusts the pre-trained model using the output for our multi-task regression task. Since the pre-trained model already encode much information about our language, the fine-tuning phase takes substantially less time compared to training the entire model from scratch. The steps in the two-phase training are illustrated in Figure 3.

To describe the token-level transformer encoder in more detail, we let $W_i = \left(w_i^1, w_i^2, ..., w_i^{|W_i|}\right)$ be a text-based sentence, where $|W_i|$ is the length of the sentence W_i and $w_i^{|W_i|}$ is an artificial EOS (end of sentence) token. The word embedding matrix associated with sentence W_i is initialized as

$$\mathbf{E}_i = \left(\mathbf{e}_i^1, \mathbf{e}_i^2, \ldots, \mathbf{e}_i^{|t_i|}\right)$$
$$\text{where } \mathbf{e}_i^j = e\left(w_i^j\right) + \mathbf{p}_j. \tag{3}$$

Here $e(\cdot)$ maps each token to a d dimensional vector using the WWM-BERT, and p_j is the position embedding of the token w_i^j with the same dimension d. Consequently, $e_i^j \in \mathbb{R}^d$ for all j. The calculation of the position embeddings is performed in the same manner as in [55]:

$$p_{j,2m} = \sin\left(j/10000^{2m/d}\right) \tag{4}$$

$$p_{j,2m+1} = \cos\left(j/10000^{2m/d}\right) \tag{5}$$

where j is the position of the token and m is the dimension of the embedding.

A sentence representation $T_i \in \mathbb{R}^{d_t}$ of the sentence W_i is calculated by average pooling that operates over the second last layer of network due to the experimental experience, where d_t represents the default dimensions of word embeddings.

4.2 Multimedia Information Fusion

The sentence representations and the corresponding audio features are then combined. An earnings call document is represented as

$$\begin{aligned}\mathcal{D}^{(k)} &= \left(s_1^{(k)}, s_2^{(k)}, \ldots, s_M^{(k)}\right) \\ \text{where } \mathbf{s}_i^{(k)} &= \left((T_i^{(k)}, A_i^{(k)}) + P_i\right).\end{aligned} \tag{6}$$

Here T_i^k and A_i^k represent the sentence and audio features of sentence i in document $\mathcal{D}^{(k)} \in \mathbb{R}^{M \times d_s}$, and $P_i \in \mathbb{R}^{M \times d_s}$ denotes the trainable sentence-level position embedding, and M is the maximum number of sentences in any document.

4.3 Sentence-level Transformer Encoder

The sentence-level transformer encoder extracts sentence-level features for prediction. The architecture of this encoder is shown in Figure 4. In particular, the architecture consists of two layer normalization steps [1]:

$$H = \text{LayerNorm}\left(\mathcal{D}^{(k)} + \text{MultiHead}\left(\mathcal{D}^{(k)}\right)\right) \tag{7}$$

$$L^{(k)} = \text{LayerNorm}(H + \text{MLP}(H)) \tag{8}$$

where LayerNorm is layer normalization introduced in [1], MLP denotes a two-layer feed-forward network with ReLU activation function, and MultiHead denotes the multi-head attention mechanism proposed in [55].

The multi-head attention applied to the documents $\{\mathcal{D}^k\}$ is calculated as follows:

$$\text{MultiHead} = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_\text{h}\right) W^O \tag{9}$$

$$\text{head}_\text{i} = \text{Attention}\left(Q, K, V\right) \tag{10}$$

$$\begin{aligned}\text{where } Q &= \mathcal{D}^{(k)} W_i^Q, \\ K &= \mathcal{D}^{(k)} W_i^K, \\ V &= \mathcal{D}^{(k)} W_i^V\end{aligned} \tag{11}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_s \times d_s}$ are weight metrics, and the attention is computed as

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_s}}\right)\mathbf{V} \tag{12}$$

for some input query, key and value matrices $Q, K, V \in \mathbb{R}^{M \times d_s}$. The h outputs from the attention calculations are concatenated and transformed using a output weight matrix $W^o \in \mathbb{R}^{d_s h \times d_s}$.

4.4 Multi-task Prediction Layer

The multi-task prediction layer consists of two separate single-layer feed-forward networks. An average pooling is first applied to the output of the sentence-level transformer encoder where the resulting output is then fed into these two feed-forward networks. The objective function is a weighted average of the loss of the two prediction tasks:

$$\mathcal{F} = \frac{\alpha \sum_i \left(\hat{y}_i - y_i\right)^2 + (1 - \alpha) \sum_j \left(\hat{y}_j - y_j\right)^2}{2n} \tag{13}$$

where $\hat{y}_i$ and $\hat{y}_j$ are the predicted values for the main and auxiliary tasks, respectively, and y_j denote the corresponding true volatility. The weight $\alpha \in [0, 1]$ controls the importance of the auxiliary task and is tuned using the validation set. We use Adam [31] as the optimizer and adopt the trick of decay learning-rate with the steps increase to train our model until converge.

5 EVALUATION

We describe the dataset for our application and several baselines for the task of stock volatility prediction. A metric to assess and compare the performance of each method is also introduced.

5.1 Dataset

The dataset used in this paper is a public S&P 500 Earning Conference Calls dataset used by [46][1]. It contains the audio records and the corresponding text transcripts from earnings calls for 500 large public companies traded on American stock exchanges (S&P 500) during 2017. There are 2,243 earnings conference calls in 2017 in the raw dataset. However, a large proportion of raw data was discarded because the audio-text alignment is very noisy and is prone to errors. So, there are 576 unique training instances (earnings calls) in which the audio records are sufficiently closely aligned with the corresponding text transcripts in total; the remainder of instances are removed due to a lack of alignment between the audio and text content. These 576 earning calls (instances) correspond to 88,829 aligned sentences (text and audio). In addition to this call data we downloaded the dividend-adjusted closing prices needed for volatility prediction from Yahoo Finance [2]. Also, the pre-trained WWW-BERT model [3] is used to form text representation for each input token, and consequently a sentence representation is obtained.

5.2 Baselines

We compare our approach to volatility prediction to a number of important baselines, chosen to reflect the range of approaches that have been applied to the volatility forecasting task, and also including the current state-of-the-art. These baselines can be grouped depending on whether they use historical pricing data [20, 29, 39, 57] (classical approaches), more recent uses of textual data [54, 64], or even more recent uses of multi-modal data [45, 46]. In each case we outline several different baselines, which, to the best of our knowledge, collectively offer the best available volatility prediction methods at the time of writing.

5.2.1 Price-based baselines. : The following approaches all rely on historical pricing data only, as the basis of volatility prediction:

(1) **Classical Methods:** Its include the GARCH model (an classical auto-regressive volatility prediction model) [19] and its variants [29]. These are among the most common approaches for volatility prediction. They are designed for short term volatility prediction, and tend to be less effective when it comes to average (n-day) volatility prediction. Therefore, the more effective prediction results corresponding to the ARCH are reported here.

(2) **LSTM [20]:** Long short-term memory networks (LSTMs) are widely used in financial time series prediction. For volatility prediction, we choose a simple LSTM as a benchmark using the preceding, optimal, n-day historical volatility.

[1] Earning call dataset: https://github.com/GeminiLn/EarningsCall_Dataset
[2] https://finance.yahoo.com/
[3] https://github.com/google-research/bert

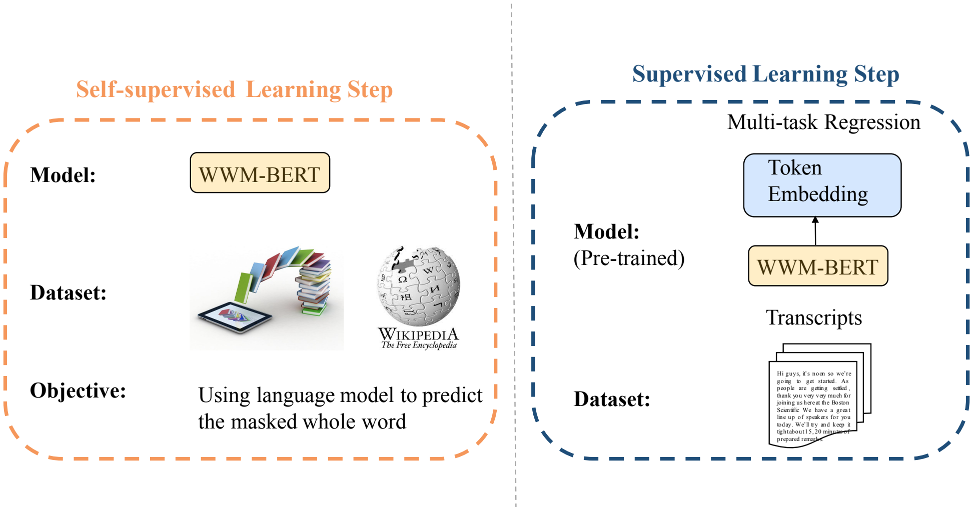


Figure 3: The pre-trained step in left is on-line available (trained on un-annotated data), and fine-tuning it on our prediction objectives.

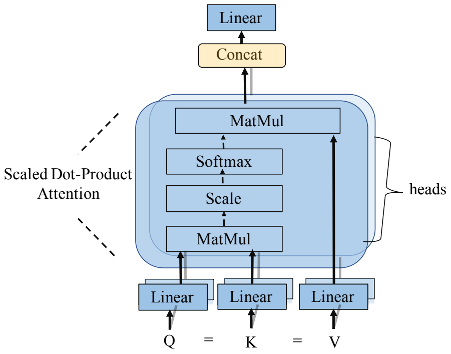


Figure 4: The sentence-level transformer mechanism. It contains self-attention mechanism and multi-head attention.

(3) **LSTM+ATT [57]:** By incorporating an attention mechanism with an LSTM we can build a prediction model that can focus on specific period of volatility in the training data, rather than assuming uniform historical data.

(4) **MT-LSTM+ATT [39]:** This multi-task variation combines average n-day volatility (the primary prediction task) with single-day volatility prediction using attention-based LSTMs as the underlying learners.

5.2.2 *Text-based baselines.*

: The following text-based approaches all rely on earnings call transcripts for volatility prediction. The baselines themselves reflect recent significant progress in this task and include the current state-of-the-art in volatility prediction tasks.

(1) **SVR+RBF(TF-IDF) [54]:** Following previous studies [54], Support Vector Regression (SVR) with a Radial Basis Function (RBF) kernel is adapted for stock volatility prediction, representing each instance as a vector of TF-IDF scores (Term Frequency-Inverse Document Frequency [59]) for each term in an earnings call transcript. The TF-IDF of a given term t is calculated as

$$\frac{\log\left(1+tc_{d_i}(t)\right)}{\|d_i\|}\log\left(1+\frac{|d_i|}{df(t)}\right)$$

where $tc_{d_i}(t)$ is the number of occurrences of term t in transcript i, $\|d_i\|$ denotes the Euclidean norm of the term weights of the transcript, and $|d_i|$ is the number of the terms in the transcript.

(2) **SVR+RBF(Glove):** This baseline again uses SVR+RBF but with each transcript term mapped to a pre-trained Glove 300-dimensional embedding [44] so that the transcript is represented as a weighted average of the embeddings. This intuition is that this provides a richer transcript representation than using the raw terms.

(3) **HAN (Glove) [64]:** For this baseline, we use a Hierarchical Attention Network with two levels of attention mechanisms, which are applied to word and sentence levels. Each word in a sentence is first converted to a word embedding using the pre-trained Glove 300-dimensional embeddings. Then each sentence, with its embedded words, is input into a Bi-GRU encoder [3, 11], while another Bi-GRU encoder is used to represent each document as a sequence of sentences. The document representation is then passed to the final regression layer for predictions.

5.2.3 *Multimodal baselines:*

These baseline all combine transcript text and audio data and the *MDRM* version represents the current state-of-the-art in volatility prediction.

(1) **SVR (Glove+Audio) [46]:** Both text and audio features are used as input features for a SVR in which both types of input are fused using a simple shallow model.

(2) **bc-LSTM (Glove+Audio) [45]:** We use a *bi-directional contextual* LSTM, proposed by [45], to extract context-dependent

multi-modal utterance features, including text features, audio features, and video features.

(3) **MDRM [46]:** This recent multi-modal deep regression model is the current state-of-the-art in volatility prediction.

The overfitting problem of audio-only model is reported in the previous work [46]. For this reason, different from the previous work, we discuss our model's performance in two scenarios: text-only and text+audio.

Table 1: Parameter Settings

Parameters	Values
Number of layers	2
Number of heads	2
Learning rate	2e-5
Batch size	4
Max sequence length	520
Dropout probability	0.5

5.3 Methodology

To facilitate a direct comparison with the current state-of-the-art (the MDRM based on [46]) we follow the evaluation carried out by [46] by splitting our dataset into mutually exclusive training/validation/testing sets in the ratio 7:1:2, and the 7:1:2 split refers to the earning calls. We sort the dataset (i.e. earning calls) in chronological order because the future data cannot be used for prediction. For each baseline (plus our HTML approach), each model is trained using the training set.

In line with best practice, model hyper-parameters are tuned using the validation set. In particular, the maximum sequence length is set as 520 following [46], and for the token-level model, we use the default settings for the hyper-parameters of WWM-BERT to encode each token. Based on that, we develop an agile transformer in the sentence-level to reduce the training and prediction time. We use a grid search to determine the optimal parameters and select the learning rate λ for Adam among {1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4}, the depth of transformer layers $h \in \{1, 2, 3, 4\}$, the number of multi-head attention $m \in \{1, 2, 3, 4\}$, and the batch size $b \in \{4, 8, 16, 32\}$. The optimal hyper-parameters are shared among all settings except the trade-off parameter α between two tasks. The different optimal parameters α are tuned on the validation set for different $n - days$ volatility predictions. Each of these tuned models is then evaluated based on its ability to predict average n-day volatility using the test set, for $n = 3, 7, 15, 30$. The resulting optimal hyperparameter values used are reported in Table 1.

The resulting predictions are compared to the actual volatility values to compute a mean squared error; see Equation 14, where $\hat{y}_i$ is the predicted value, y_i denotes the actual volatility.

$$MSE = \frac{\sum_i (\hat{y}_i - y_i)^2}{n} \tag{14}$$

6 RESULTS AND DISCUSSION

The results of this evaluation are presented in Table 2, for each of the baselines and a number of variations of our HTML model, for 3, 7, 15, and 30-day time-periods. It should be clear that significant prediction benefits accrue to the HTML model. The HTML model achieves the highest prediction performance (lowest MSE values) for each of the target time-periods. In particular, the text-only and text+audio versions of HTML generate predictions with substantially lower errors compared to the corresponding versions of the current state-of-the-art, MDRM alternative. These error improvements relative to MDRM are substantial significant, varying with the time-period as follows: 3-days (+38.4%), 7-days (+16.9%), 15-days (+49.0%), and 30-days(+38.7%). Improvements of this scale, relative to the state-of-the-art, are likely to translate into substantial practical benefits and suggest that this new HTML approach stands as a new performance benchmark for volatility forecasting.

In addition to such overall measures of performance, however, we are also interested in better understanding the different relative contributions, if any, that the design decision of the HTML model make, when it comes to prediction performance. Thus, in the following subsections we consider a number of related evaluation questions to better the relative contributions of data sources and model components.

6.1 Comparing Price-based Methods with Alternative Methods

Table 2 shows how both text-based and multimodal approaches consistently outperform methods that are purely based on historical pricing, for both short-term ($n = 3$) and long-term ($n = 30$) volatility prediction. Excluding the HTML model, the performance of price-based methods and other methods offer comparable for medium-term ($n = 7, 15$) volatility prediction performance. And, in the case of HTML, its prediction performance always exceeds that offered by pricing-based methods, regardless of n. This provides strong evidence in support of the idea that text and audio features can improve volatility prediction.

6.2 On the Utility of Audio Features

Previous research [46] has demonstrated the benefits of combining text with audio data, compared to text-only features, in volatility prediction; [46] reported significant differences, based on a one-tailed t-test, for n=3/n=7 $p \leq 0.001$ and for n=15 $p \leq 0.01$). For HTML, the benefits of using multimodel learning are statistically significant for n=3 only, however ($p \leq 0.01$). HTML delivers its most accurate short-term predictions using text+audio, but its most accurate long-term predictions come from the text-only version. This may hint that short-term volatility is more greatly influenced by the vocal cues contained within audio features, although further research is required, as short-term volatility may also be impacted opportunistic effects such as so-called post earnings announcement drift (PEAD) [5].

6.3 On the Benefits of the Hierarchical Transformer Architecture

We explore the benefits of attention mechanisms for price-based and text-based models separately. For technical analysis, the attention mechanism based on LSTM achieves some minor improvement in almost all of the settings, excluding n=7. While in the text-based methods, if we adapt a hierarchical attention network (HAN) with

Table 2: The average n-day volatility prediction errors for our approach (HTML) and the various baselines, including the MDRM state-of-the-art. The MSE in bold indicates the best MSE across all approaches, while those in italics indicate the stae-of-the-art MSEs.

Price-based Methods		**n=3**	**n=7**	**n=15**	**n=30**
Linear Regression		1.710	0.526	0.330	0.284
LSTM		1.970	0.459	0.320	0.235
LSTM+ATT		1.852	0.470	0.308	0.231
MTLSTM+ATT		1.983	0.435	0.304	0.233
Text-based Methods		**n=3**	**n=7**	**n=15**	**n=30**
SVR+RBF(TF-IDF)		1.695	0.498	0.342	0.249
SVR+RBF(Glove)		1.667	0.549	0.345	0.275
HAN(Glove)		1.426	0.461	0.308	*0.198*
Multimodal Methods		**n=3**	**n=7**	**n=15**	**n=30**
SVR(Glove+Audio)	Text+Audio	1.722	0.501	0.307	0.233
bc-LSTM(Glove+Audio)[45]	Text+Audio	1.418	0.436	0.304	0.219
MDRM [46]	Text Only	1.431	0.439	0.309	0.219
	Audio Only	1.412	0.440	0.315	0.224
	Text+Audio	*1.371*	*0.420*	*0.300*	0.217
HTML (Ours)	Text Only	1.175	0.372	**0.153**	**0.133**
	Text+Audio	**0.845**	**0.349**	0.251	0.158

a bi-directional GRU model, we note a distinct improvement. It is noteworthy that HAN outperforms the state-of-the-art multimodal results for n=30. This finding provides further evidence in support of the idea that audio features are unlikely to contribute significantly to longer term volatility predictions.

We also compare the results obtained from the attention model used in HAN and our Hierarchical Transformer, which contains self-attention and mutual-head attention, with text only data. The performance of our model is stronger on all tasks, suggesting improvements due to the progressive architecture of Hierarchical Transformer and the use of pre-trained word embeddings.

Regarding the embeddings used, the results of an ablation study on the different embeddings used by HTSL and HTML approaches used in this work are presented in Table 3. As might be expected, WWM-BERT has a beneficial effect on each prediction task compared to Glove; although adding audio features to the Glove embeddings offers similar performance benefits.

6.4 Single-Task vs Multi-Task Approaches

Also in Table 3 we can see how the multi-task approach tends to offer improved performance compared to the single-task approach. On a like-for-like basis, most of the multi-task variations in Table 3 present that we superior prediction performance when compared to the corresponding single-task variation, especially for long-term prediction tasks.

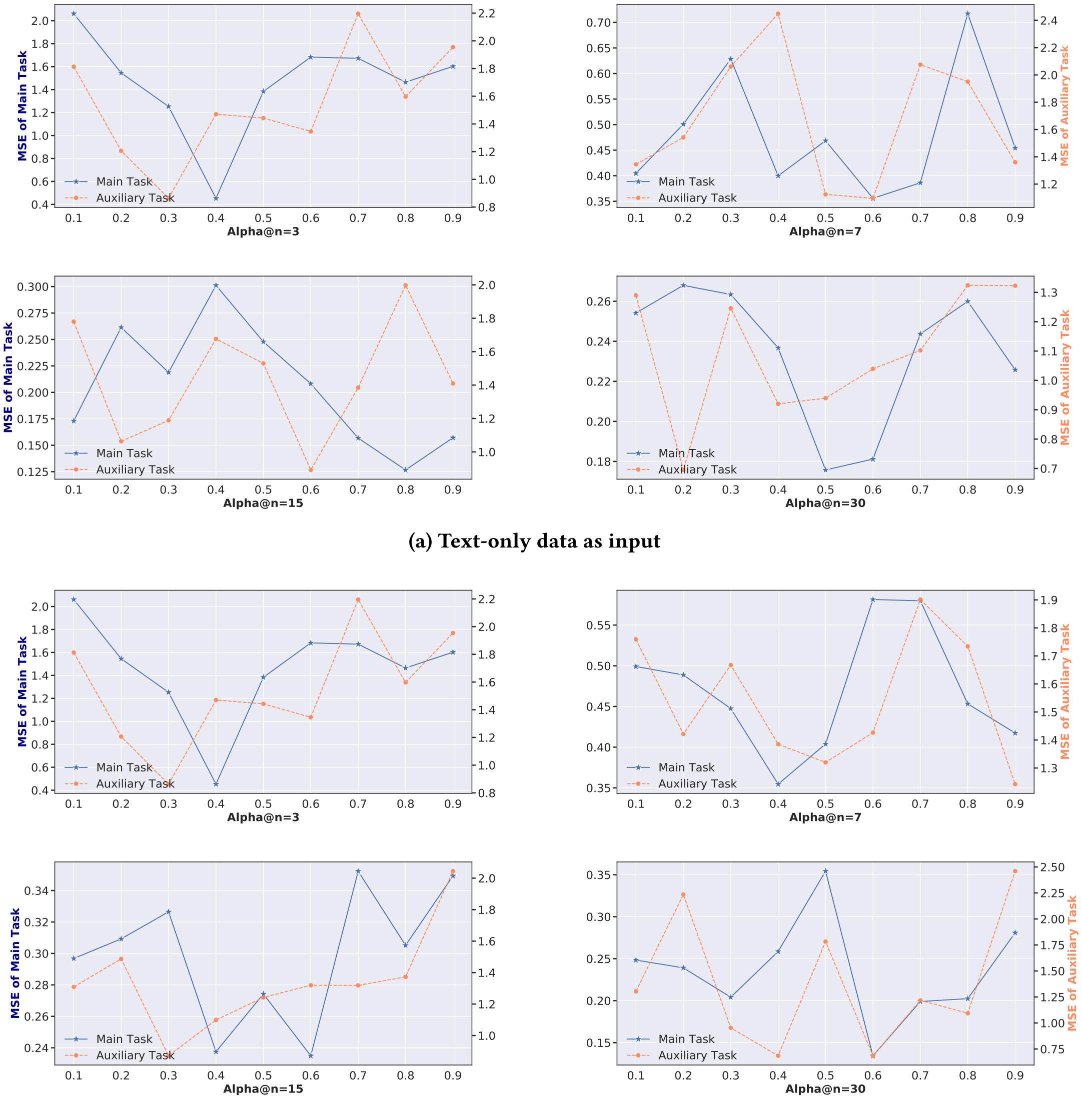
We further explore how the auxiliary (single-day prediction) task affects prediction performance. The influence of the auxiliary weight α is important, because it determines relative weight of each task during learning. The *validation* MSE results, by varying α, are presented in Figure 5. Each individual graph shows the n-day (main task) and single-day (auxiliary task) MSE for a different value of n and a range of values for α, and for text-only and multi-modal variations. Using text-only data the optimal value for alpha (minimum MSE on the main task) varies in the range 0.5 to 0.8 for different values of n, whereas for multi-modal data it tends to be lower, in the range 0.4 to 0.6, for varying n. By tuning α during the validation stage we are effectively trading-off n-day prediction performance and single-day prediction performance and overall we can see that n-day performance can be optimised by tuning in this way.

Table 3: Ablation studies on the multi-task learning and embeddings. HTSL and HTML are short for Hierarchical Transformer-based Single-task Learning and Hierarchical Transformer-based Multi-Task Learning respectively

Model	**Embeddings**	**n=3**	**n=7**	**n=15**	**n=30**
HTSL	Glove	1.558	0.469	0.291	0.181
	Glove+Audio	1.313	0.389	0.330	0.238
	WWM-BERT	1.344	0.363	0.271	0.162
	WWM-BERT+Audio	1.087	0.432	0.308	0.181
HTML	Glove	1.574	0.474	0.276	0.164
	Glove+Audio	1.278	0.370	0.282	0.201
	WWM-BERT	1.175	0.372	**0.153**	**0.133**
	WWM-BERT+Audio	**0.845**	**0.349**	0.251	0.158

7 CONCLUSIONS

Predicting the historical volatility of publicly traded companies is an important financial analysis task and considerable research effort in the past has been devoted to producing models that are capable of predicting pricing volatility for different time horizons. Recent advances in machine learning means that researcher attention has



(a) Text-only data as input

(b) Multimedia data as input

Figure 5: Validation MSE as a result of varying α. Left and right y-axis represent the validation MSE of primary task and auxiliary task respectively.

moved from conventional time-series prediction approaches, based on historical pricing data, to more sophisticated methods that incorporate alternative sources of (often unstructured) data such as text reports or social media.

In this paper we have proposed a novel hierarchical, multi-task, transformer learning model for volatility prediction, based on the text and/or audio of earning calls. The model builds on very recent work by [46] and delivers substantial performance improvements, for short and long-term volatility prediction, providing a new performance benchmark for this task. Moreover, our evaluation includes a detailed study of a variety of experimental conditions, to better understand the relative contributions of different aspects of the proposed model to prediction performance.

The utility of audio data, and vocal features, in this important financial prediction task, suggests there exists a significant opportunity to explore the use of audio features in a range of related or complementary tasks (e.g. fraud detection, asset pricing, stock recommendation etc.), where such data is readily available alongside more traditional forms of financial data.

ACKNOWLEDGMENTS

This research was supported by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_2.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Jo-Anne Bachorowski. 1999. Vocal expression and perception of emotion. *Current directions in psychological science* 8, 2 (1999), 53–57.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] P Belin, B Boehme, and P McAleer. 2017. The sound of trustworthiness: Acoustic-based modulation of perceived voice personality. *PLoS ONE* 12, 10 (2017), e0185651.

[5] Victor L Bernard and Jacob K Thomas. 1989. Post-earnings-announcement drift: delayed price response or risk premium? *Journal of Accounting research* 27 (1989), 1–36.

[6] Paul Boersma and Vincent Van Heuven. 2001. Speak and unspeak with praat. *Glot International* 5, 9/10 (2001), 341–347.

[7] Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science* 2, 1 (2011), 1–8.

[8] Tim Bollerslev. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31, 3 (1986), 307 – 327.

[9] Michael W Brandt and Christopher S Jones. 2006. Volatility forecasting with range-based egarch Models. *Journal of Business & Economic Statistics* 24, 4 (2006), 470–486.

[10] Rich Caruana. 1993. Multitask learning: A knowledge-Based source of inductive bias. In *ICML*.

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[12] John C. Cox and Stephen A. Ross. 1976. The valuation of options for alternative stochastic processes. *Journal of Financial Economics* 3, 1 (1976), 145 – 166.

[13] Li Deng, Geoffrey Hinton, and Brian Kingsbury. 2013. New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 8599–8603.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

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

[16] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

[17] Robert F. Engle. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica* 50, 4 (1982), 987–1007.

[18] Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-News: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749* (2019).

[19] Philip Hans Franses and Dick Van Dijk. 1996. Forecasting stock market volatility using (non-linear) Garch models. *Journal of Forecasting* 15, 3 (1996), 229–235.

[20] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[21] Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A joint many-task model: growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1923–1933.

[22] Steven L Heston. 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies* 6, 2 (1993), 327–43.

[23] Gerard Hoberg and Gordon Phillips. 2016. Text-based network industries and endogenous product differentiation. *Journal of Political Economy* 124, 5 (2016), 1423–1465.

[24] Jessen L Hobson, William J Mayew, and Mohan Venkatachalam. 2012. Analyzing speech to detect financial misreporting. *Journal of Accounting Research* 50, 2 (2012), 349–392.

[25] Stephan Hollander, Maarten Pronk, and Erik Roelofsen. 2010. Does silence speak? An empirical analysis of disclosure choices during conference calls. *Journal of Accounting Research* 48, 3 (2010), 531–563.

[26] Y.L. Hsu, T.I. Lin, and C.F. Lee. 2008. Constant elasticity of variance (CEV) option pricing model: Integration and detailed derivation. *Mathematics and Computers in Simulation* 79, 1 (2008), 60 – 71.

[27] Xiaoming Jiang and Marc D Pell. 2017. The sound of confidence and doubt. *Speech Communication* 88 (2017), 106–126.

[28] Herb Johnson and David Shanno. 1987. Option pricing when the variance is changing. *The Journal of Financial and Quantitative Analysis* 22, 2 (1987), 143–151.

[29] Ha Young Kim and Chang Hyun Won. 2018. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications* 103 (2018), 25–37.

[30] Michael D Kimbrough. 2005. The effect of conference calls on analyst and market underreaction to earnings announcements. *The Accounting Review* 80, 1 (2005), 189–219.

[31] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[32] Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. [n. d.]. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 272–280.

[33] Werner Kristjanpoller, Anton Fadic, and Marcel C Minutolo. 2014. Volatility forecast using hybrid neural network models. *Expert Systems with Applications* 41, 5 (2014), 2437–2442.

[34] David F Larcker and Anastasia A Zakolyukina. 2012. Detecting deceptive discussions in conference calls. *Journal of Accounting Research* 50, 2 (2012), 495–540.

[35] Hongquan Li and Yongmiao Hong. 2011. Financial volatility forecasting with range-based autoregressive volatility model. *Finance Research Letters* 8, 2 (2011), 69–76.

[36] Shouwei Liu and Yiu Kuen Tse. 2013. Estimation of monthly volatility: An empirical comparison of realized volatility, GARCH and ACD-ICV methods. (2013).

[37] Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164* (2019).

[38] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265* (2019).

[39] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* (2015).

[40] Xiaojuan Ma, Emily Yang, and Pascale Fung. 2019. Exploring perceived emotional intelligence of personality-driven virtual agents in handling user challenges. In *The World Wide Web Conference*. ACM, 1222–1233.

[41] Asaf Manela and Alan Moreira. 2017. News implied volatility and disaster concerns. *Journal of Financial Economics* 123, 1 (2017), 137–162.

[42] William J Mayew and Mohan Venkatachalam. 2012. The power of voice: Managerial affective states and future firm performance. *The Journal of Finance* 67, 1 (2012), 1–43.

[43] Nuno Oliveira, Paulo Cortez, and Nelson Areal. 2017. The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications* 73 (2017), 125–144.

[44] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[45] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 873–883.

[46] Yu Qin and Yi Yang. 2019. What you say and how you say it matters: Predicting stock volatility using verbal and vocal cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 390–401.

[47] Navid Rekabsaz, Mihai Lupu, Artem Baklanov, Alexander Dür, Linda Andersson, and Allan Hanbury. 2017. Volatility prediction using financial disclosures sentiments with word embedding-based IR models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1712–1721.

[48] Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6949–6956.

[49] Mark Schroder. 1989. Computing the constant elasticity of variance option pricing formula. *The Journal of Finance* 44, 1 (1989), 211–219.

[50] Louis O. Scott. 1987. Option pricing when the variance changes randomly: Theory, estimation, and an application. *The Journal of Financial and Quantitative Analysis* 22, 4 (1987), 419–438.

[51] Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 231–235.

[52] Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. 2017. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. *arXiv preprint arXiv:1704.01631* (2017).

[53] Quoc-Tuan Truong and Hady Lauw. 2019. Multimodal review generation for recommender systems. In *The World Wide Web Conference*. ACM, 1864–1874.

[54] Ming-Feng Tsai and Chuan-Ju Wang. 2014. Financial keyword expansion via continuous word vector representations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1453–1458.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[56] William Yang Wang and Zhenhao Hua. 2014. A semiparametric gaussian copula regression model for predicting financial risks from earnings calls. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1155–1165.

[57] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 606–615.

[58] Fangzhao Wu and Yongfeng Huang. 2016. Sentiment domain adaptation with multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 301–310.

[59] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. 2008. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)* 26, 3 (2008), 13.

[60] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1970–1979.

[61] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. 2007. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research* 8, Jan (2007), 35–63.

[62] Linyi Yang, Ruihai Dong, Tin Lok James Ng, and Yang Xu. 2019. Leveraging BERT to improve the FEARS index for stock forecasting. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*. 54–60.

[63] Linyi Yang, Zheng Zhang, Su Xiong, Lirui Wei, James Ng, Lina Xu, and Ruihai Dong. 2018. Explainable text-driven neural network for stock prediction. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*. IEEE, 441–445.

[64] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 1480–1489.

[65] Xi Zhang, Yunjia Zhang, Senzhang Wang, Yuntao Yao, Binxing Fang, and S Yu Philip. 2018. Improving stock market prediction via heterogeneous information fusion. *Knowledge-Based Systems* 143 (2018), 236–247.

[66] Jie Zheng, Andi Xia, Lin Shao, Tao Wan, and Zengchang Qin. 2019. Stock volatility prediction based on self-attention networks with social information. In *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. IEEE, 1–7.