Adapting Speech Models for Stock Price Prediction

Frederic Voigt
University of the West of Scotland / Hamburg University of Applied Sciences
Department of Computer Science
Hamburg, Germany
b01742821@studentmail.uws.ac.uk

Jose Alcarez Calero
University of the West of Scotland
School of Computing, Engineering & Physical Sciences
Paisley, Scotland
jose.alcaraz-calero@uws.ac.uk

Keshav Dahal
University of the West of Scotland
School of Computing, Engineering & Physical Sciences
Paisley, Scotland
keshav.dahal@uws.ac.uk

Qi Wang
University of the West of Scotland
School of Computing, Engineering & Physical Sciences
Paisley, Scotland
qi.wang@uws.ac.uk

Kai von Luck
Hamburg University of Applied Sciences
Department of Computer Science
Hamburg, Germany
kai.vonluck@haw-hamburg.de

Peer Stelldinger
Hamburg University of Applied Sciences
Department of Computer Science
Hamburg, Germany
peer.stelldinger@haw-hamburg.de

***Abstract*—Large language models (LLMs) have demonstrated remarkable success in the field of natural language processing (NLP). Despite their origins in NLP, these algorithms possess the theoretical capability to process any data type represented in an NLP-like format. In this study, we use stock data to illustrate three methodologies for processing regression data with LLMs, employing tokenization and contextualized embeddings. By leveraging the well-known LLM algorithm Bidirectional Encoder Representations from Transformers (BERT) [1], we apply quantitative stock price prediction methodologies to predict stock prices and stock price movements, showcasing the versatility and potential of LLMs in financial data analysis.**

***Index Terms*—finance, quantitative stock price prediction, natural language processing, stock movement prediction, fintech, machine learning, large language models**

I. Introduction

Few subfields in machine learning (ML) have garnered as much attention in recent years as NLP and the LLMs integral to its advancements. Although these models excel in NLP tasks, they are fundamentally versatile algorithms capable of processing any data type that is appropriately formatted.

At a conceptual level, LLMs process sequentially arranged data points - specifically, word-tokens - that encapsulate their interrelationships and correlations through the positions of their respective embedding vectors in the vector space. Numerous tasks within NLP involve the derivation of subsequent textual outcomes (i.e. prediction of future developments of the input sequence as for example in next-token prediction) or overarching semantic interpretations, such as sentiment analysis, based on these structured inputs. Upon examining these internal mechanisms, it becomes evident that numerous research fields within ML exhibit analogous processing requirements for their respective input data. Stocks for example are highly correlated, dependent on other stocks [2] and sequentially ordered trough their temporal price development. Based on these shared characteristics with NLP data, it is obvious why LLM algorithms can be potentially interesting for processing stock data. Voigt et al. [3] have demonstrated that language data and stock market data not only share conceptual but also structural similarities, suggesting the potential applicability of LLMs to financial datasets. Despite being relatively underexplored in contemporary research, numerous time series problems can be reformulated to align with the operational framework of LLMs.

LLMs typically incorporate three coarse-grained processing stages. Initially, the input text undergoes tokenization, during which each of the l input word-tokens is mapped to an index within a predefined vocabulary $V \subset \mathbb{N}$. Subsequently, each token index is assigned a contextualized (pre-trained) embedding tensor (e.g. using the Word2Vec [4], [5] algorithm). In the final stage, these embedding vectors are concatenated to form a two-axis tensor, which is then fed into the model for further processing. A sketch of the basic processing idea can be seen in the first row in Figure 1.

We propose a series of methodologies to adapt each step of the pipeline of LLMs in order to make them usable for stock data. First, we employ stock data as embedding values, inputting these embeddings over a specified time lag Δt and for a set of company stocks C. This approach replaces the traditional method of concatenating embedding vectors with stock data. Secondly, we explore the application of scaling contextualized embeddings e_i for various companies $c_i \in C$. The method, originally proposed in [3] serves as an analog to the technique used in Word2Vec for generating word vector embeddings, adapted for financial entities. Lastly, our third approach involves the tokenization of stock regression data, which aims to replicate the entire pipeline of an LLM. A visualization of these concepts is shown in Figure 1. We have decided to adapt the BERT model [1] for this work, as it is one of the first models in the LLM landscape.

To summarize our contributions, we delineate three methodologies that leverage LLMs for broader applications to regression and time-series analysis, specifically using stock data as

an example. We validate our ideas using the BERT model trained on 60-minute resolution intraday stock data of Standard and Poor's 500 (S&P-500) companies. As tasks we use the prediction of future prices (stock price prediction) and the prediction of whether a stock price will rise or fall in the future (stock movement prediction). We conceptualize end to end models that do not require further feature selection, domain knowledge or work from (expensive) financial experts. The primary objective of this work is to demonstrate the applicability of established NLP algorithms in the domain of financial analytics by utilizing language models and stock data. We aim to present initial findings that support the feasibility of this approach. It is important to note that our goal is not to identify superior stock movement prediction (SMP) / stock price prediction (SPP) methodologies, but rather to highlight the potential and usability of LLMs in this novel context. Given this objective, our initial focus is on quantitative stock price prediction, which exclusively utilizes numerical, historical stock data [6]. This approach is in contrast to fundamental stock analysis [7], which incorporates a broader spectrum of information, including annual reports, social media data, or other relevant sources. For the integration of such fundamental data, additional methodologies are discussed in Section VII.

II. Related Work

Voigt et al. [3] pioneered the concept of adapting speech models to multivariate regression data (i.e. stock data) by reformatting such data into sentence-like structures. This approach involves scaling multidimensional contextualized embedding vectors $\hat{e}_i$ for each c_i dependent on the price information $x_i^{(t)}$ of c_i at timestep t, thereby incorporating regression data. Although [3] introduced these foundational ideas, their work lacked experimental validation of the adapted speech models (ASMs) and did not explore the use of tokenized regression data or two-axis tensor representations as embedding inputs.

Further advancing their concepts, [3] also introduces the Stock2Vec algorithm, designed to train embeddings $\hat{e}_i$ specifically tailored for various c_i. In the financial ML domain, the strategy of representing c_i as contextualized embeddings is commonly employed to uncover correlations among assets, primarily to enhance risk minimization and portfolio optimization strategies. Contextualized embeddings for c_i work similar to NLP where contextualized word-token embeddings represent relationships and meanings of the word-tokens. Embeddings of c_i typically cluster stocks within the same industry close together or express similar relations trough similar distance vectors. In related literature such as [8], [9], [10], and [11] embedding training strategies for c_i are introduced. Non of these models use the embeddings for SMP/SPP downstream tasks.

The intercorrelation among stocks stands as a pivotal factor in forecasting future stocks prices, a notion widely acknowledged in contemporary research literature as for example in [12] or [2]. In our proposed approach the intricate interrelationships are encapsulated through e_i. Traditional methodologies commonly integrate correlation matrices directly into the modeling pipeline, as exemplified by [13], or leverage them within graph-based networks, as showcased in [14]. However, few studies opt for the utilization of c_i-specific vectors e_i, as demonstrated by [15] or [2].

The importance of including relational information in stock presentations was demonstrated, for example, by Kim et al. [16]. In their model, there are performance differences depending on the basis from which relationships between two stocks are modeled (e.g. "Industry-Product or material produced" or "Country of origin-Country"). The ablation study in [14] shows decreasing performance if relationship modeling is omitted.

An approach loosely related to the tokenization of stock data that involves inputting quantitative stock data (along fundamental data) as prompts into LLMs is discussed in [17] and in [18]. One of the relatively few instances of employing LLMs for non-linguistic data is exemplified by the ALBEF model developed by Li et al. [19]. This model utilizes parts of the pre-trained BERT architectures to process merged visual and textual input features.

SPP/SMP are generally regarded as difficult problems [20] and the performance of ML models is correspondingly low. Especially when only quantitative methods are used, as in the approaches presented here. In order to make our proposed approach comparable with the literature, we will look at the performance of some models that were also trained on US-American stocks. In their study, Qin et al. [21] demonstrated the efficacy of quantitative SPP models, achieving a root mean square error (RMSE) of 0.31 on National Association of Securities Dealers Automated Quotations (NASDAQ) data, in contrast to a higher RMSE of 0.96 observed in their Recurrent Neural Network [22] (RNN)-based baseline model. Similarly, Feng et al. [12] reported models (using non-quantitative external relationship information) that exhibited an RMSE of 0.015 for New York Stock Exchange (NYSE) data and 0.019 for NASDAQ data. Their Long short-term memory (LSTM) [23] model based baseline models recorded RMSE values of 0.019 on the NASDAQ dataset and 0.015 on the NYSE dataset, respectively. In the context of SMP, Ding et al. [24] proposed a model that attained an accuracy of 57.3% on NASDAQ data, surpassing the LSTM baseline's accuracy of 53.89%. Furthermore, the same model achieved an accuracy of 58.7% on China A-shares data, which also exceeded the LSTM baseline accuracy of 56.7%. The model proposed in [25] exhibited significant performance, attaining an SMP accuracy of 60.7% across diverse interday data for single c_i from the S&P 500 and Korea Composite Stock Price Index (KOSPI). The authors also introduced baseline models that achieved accuracies within the range of 51.49% to 57.36%.

III. Model

As delineated in Section I, we propose three distinct methodologies employing transformer-encoder based language models for the prediction of stock prices. These approaches, along with the specific components of the LLM pipeline they

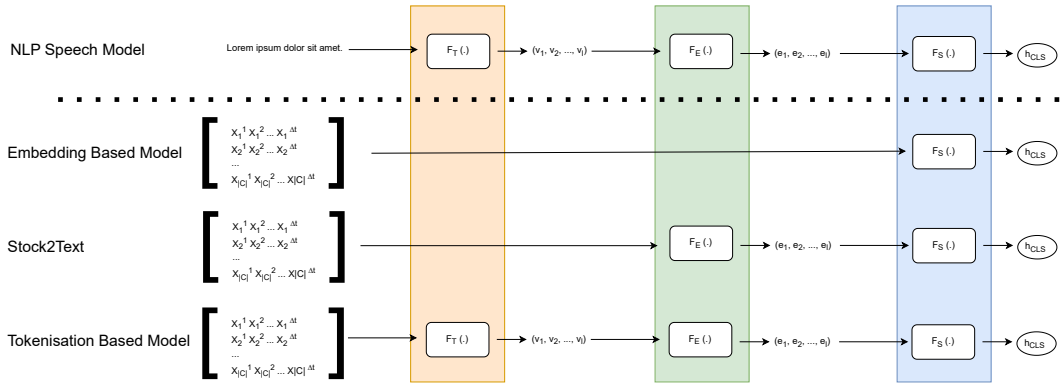

Fig. 1. Visualized comparison of the ASMs with classic NLP models.

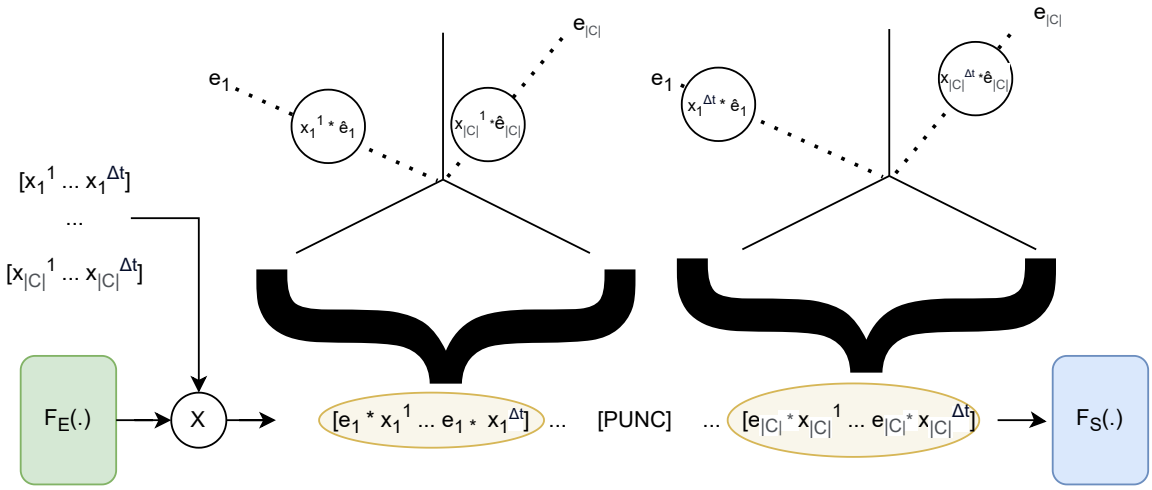

Fig. 2. Visualization of a scaled company embedding input after the conversion of a stock data time series input into a Stock2Sentence representation.

adapt, are illustrated in Figure 1. Each approach utilizes the CLS-token $h \in \mathbb{R}^\eta$, where η represents the size of the model. Subsequent to processing by the LLM, the position of this token is input into a linear layer, which is followed by a Sigmoid activation function to perform the model prediction $\hat{y}$. Depending on the specific application, either the Mean Squared Error (MSE) for SPP or Binary Cross Entropy (BCE) $\mathbb{H}(\hat{y}, y)$ for SMP is employed as the loss function. Similar to traditional predictive tasks in NLP, the CLS-token serves as a representative of the entire processed input sequence, with respect to the specified tasks.

As a language model we define the basic speech model (BERT) $F_S((\tilde{e}_i)_{1=i}^{l})$ taking embedded inputs in the form $(\tilde{e}_1, \tilde{e}_2, \dots \tilde{e}_l)$ with $\forall \tilde{e}_i \in \mathbb{R}^\eta$ and transforming it into h. In order to create the embedded word-tokens $(\tilde{e}_i)_{1=i}^{l}$ an embedding model $F_E((\hat{v}_i)_{1=i}^{l}, \tilde{E})$ which transforms tokenized text input in the form $(\tilde{v}_1, \tilde{v}_2, \dots \tilde{v}_l)$ (with $\forall \tilde{v}_i \in \tilde{V}$ and $\tilde{V} \subset \mathbb{N}$) is used. To tokenize the input text $\tilde{X}$ we use the tokenizer $F_T(\tilde{X}, \tilde{V})$.

A. Notation

In the following (with strong orientation to the notation of [3]) the price information of a specific company c_i at timestep t is denoted as $x_i^{(t)} \in \mathbb{R}^4$. Stock data is typically expressed using a 5-dimensional datapoint for a given time interval by specifying for each time step (e.g. the last 60 minutes), the Opening Price, the Highest Price, the Closing Price, the Lowest Price and the Trading Volume (OHCLV-features). In the approaches presented here, we do not use Trading Volume information. The price of all $c_i \in C$ at t is expressed as the "Market Snapshot" [3] $X^{(t)} \in \mathbb{R}^{4 \cdot |C|}$ stacking the price information of all companies. The concatenation of Market Snapshots over the whole observation horizon Δt is referenced as $X \in \mathbb{R}^{(4 \cdot |C|) \times \Delta t}$.

B. Embedding Based Approach

In alignment with the frameworks previously delineated in [26] or [24] for example, this approach employs a concatenated array of Δt Market Snapshots $X^{(t)} \in \mathbb{R}^{4 \cdot |C|}$ as embedding vectors, directly into $F_S(.)$, thereby circumventing the initial

embedding and tokenization procedure. Here we map $|C| \equiv \eta$ and $\Delta t \equiv l$.

The primary distinction between the application of a standard Transformer model, i.e the one described in [27], lies in the nuanced specificities of the modified language model $F_S(.)$ (in this case BERT).

Following the methodology introduced by Yoo et al., a linear transformation layer, with parameters uniformly shared across all c_i and incorporating a Rectified Linear Unit (ReLU) activation function, is applied to transform the raw price features X into a latent feature representation $\bar{X}$. A consequential aspect of this approach is the flexibility it affords in the input dimensionality of $F_S(.)$, thereby enabling the utilization of η, as specified by the original language model.

C. Stock2Sentence

The idea to map X into sentence-like structures is discussed in [3]. For this methodology, the procedure commences by "flattening" X, subsequently transposing the input at each respective timestep dimension, and then concatenating the flattened and transposed inputs sequentially. Each c_i is transformed into an embedding tensor $\hat{e_i}$ utilizing $F_{\mathrm{E}}(.)$ alongside the trainable embedding matrix E. The derivation and specifications of E are elaborated upon in Paragraph III-C0a.

To encode price information and construct the final embedding vector e_i, two distinct scaling methodologies are evaluated: $e_i = \hat{e_i} \bigstar x_i^{(t)}$ or $e_i = \hat{e_i} \bigstar F_{\hat{E}}(f_{\text{norm}}(x_i^{(t)}), \hat{E})$ with $\bigstar$ being defined as either a multiplicative or additive method. Each of the four OHCL-features represents one quarter of the η dimensions of e_i by stacking the vectors. In the approach utilizing $F_{\hat{E}}(.)$, $\hat{E}$ represents a learnable parameter matrix, and $f_{\text{norm}}(.)$ is a function designed to normalize the input values of each c_i and each OHCL-feature to an integer range between 0 and θ_{norm}. As delineated in [26], the technique of shifting embedding tokens within the vector space constitutes a prevalent strategy, employed, for instance, to encode positional information, as exemplified in [28].

In order to generate sentence like structures we assign

$$a^{(t)}[j, i] = e_i^{(t)}[j] \tag{1}$$

and

$$A = \begin{bmatrix} a^{(1)} & \text{[PUNC]} & a^{(2)} & \text{[PUNC]} & \dots & a^{(\Delta t)} \end{bmatrix} \tag{2}$$

with [PUNC] $\in \mathbb{R}^{\eta \times 1}$ as outlined in [3]. The punctuation token [PUNC] is utilized to aid the model in differentiating among various t. A schematic representation of this method is depicted in Figure 2.

a) Contextualized Stock Embedding Vectors: The embedding matrix E can be optimized through several methodologies, including Stock2Vec and other algorithms detailed in Section II. The aim of using E is to represent relationships between different c_i abstractly as high-dimensional vectors. Access was obtained to the pre-trained weights from the work of Dolphin et al. in [11] and [10], facilitating empirical testing. Alternatively, E can be initialized randomly and trained from scratch. For the Stock2Vec algorithm, we use a slightly adapted (compared with [26]) version in which, for example, time-dependent embeddings are used for the CBOS algorithm and the task is SMP instead of SPP.

One significant advantage of the Stock2Sentence approach is that it improves the information presentation to the model as each position in the "text" encapsulates a single data point from the second (l-dimensional) input tensor axis. In contrast, the method employing $X^{(t)}$, as discussed in Section III-B, incorporates $|C|$ information points per position. This approach endeavors to leverage the underlying principles and mechanisms of NLP models by aligning more closely with traditional NLP text structures where each input position includes one word-token. However, a notable drawback of this method is the resultant expansion in input length by a factor of $|C|$, which substantially increases computational demands. This is particularly challenging for transformer architectures, which are characterized by a space complexity of $\mathcal{O}(n^2 + n \cdot k)$ [27], thus rapidly depleting computational resources.

D. Reinterpretation as Text

The final methodology incorporates $F_E(.)$, $F_S(.)$, and $F_T(.)$, and applies tokenization to regression values. This tokenization parallels the discretization process, bearing resemblance to the embedding-scaling techniques, albeit its application within the domain of equities is notably unconventional. It has precedent in other ML contexts e.g. discretization can be analogized to the utilization of histograms for transforming continuous variables, as for example in [29].

This strategy fundamentally reinterprets X not as a tensor, but rather as textual data predominantly composed of numerical entries. Consequently, this technique facilitates a comprehensive reorientation of the problem from SPP to processes typical for NLP, marking a significant paradigm shift in the approach to data analysis.

To do this, we define the vocabulary V using the word-tokens from the set

$$C \cup \{\text{"-"}, 0, \text{"[PUNC]"}\} \cup \{n | n \in \mathbb{N}, n \leq \theta_V\} \ . \tag{3}$$

The input is represented as a sentence

$$b^{(j)} = \begin{bmatrix} c_1 \tilde{x}_1^{(j)} & c_2 \tilde{x}_2^{(j)} & c_{|C|} \tilde{x}_{|C|}^{(j)} \end{bmatrix} \tag{4}$$

and the whole input, as

$$B = \begin{bmatrix} b^{(1)} & \text{[PUNC]} & b^{(2)} & \text{[PUNC]} & \dots & b^{(\Delta t)} \end{bmatrix} \ . \tag{5}$$

For each c_i, stock ticker symbols (such as [APPL]) can be employed as single tokens, akin to the methodologies outlined in [8].

We define each transformed $x_i^{(t)}$ as $\tilde{x}_i^{(t)}$ by choosing only one price feature (i.e. the Closing Price) and implementing a digit reversal of the original values, subsequently multiplying by a scaling factor 10^{θ_x}, and rounding to θ_x decimal places. This reversal process restructures the price display such that each position aligns uniformly across different companies; for instance, the most significant digit represents the fourth decimal place, the next significant digit represents the third

decimal place, and so forth. This methodological adjustment has been found to enhance the stability of the training process.

An alternative method could involve employing the default $F_T(.)$ and V of the corresponding language model offering a different avenue for data preprocessing and analysis.

IV. Methods

In adherence to classical methodologies within the realm of LLMs, we delineate two principal phases: "pre-training" and "fine-tuning". Pre-training is employed in NLP to instill a foundational understanding of language in LLMs - a "generalized language understanding" [26]. Conversely, fine-tuning is dedicated to adapting the model for specific downstream tasks. As delineated in [26] and [3], the adaptation of popular techniques such as masked language modeling (MLM) [1] and next-sequence prediction (NSP) [1] can be similarly applied to stock prediction models. This approach teaches the models a generalized comprehension of stock data, elucidates intercorrelations among various c_i, and enhances the overall generalized performance. Pre-training stages are deferred to future investigative efforts.

In this research, our focus is primarily on the fine-tuning tasks. These tasks involve the already described SPP with the target being $y = X^{(t+o)}[D]$ and SMP with the target being $y = \mathbb{I}^{(t)}(X^{(t)} > X^{(t+o)})[D]$ (adhering to the notation established by Yoo et al in [30]) with D being the indices of the Closing Price datapoints. For the purposes of this study $o = 1$ holds, as is conventionally employed in related research, exemplified in studies such as those in [12] or [31].

V. Dataset and Experiment

The foundational data source for our dataset is the Alpha Vantage API[1], which provides us with extensive research access to stock data. Consistent with the methodology of Voigt et al., our analysis incorporates data from 309 companies listed on the S&P-500 index that had available records dating back to the year 2000. The temporal scope of our dataset extends from 2000 to 2023, with data aggregated at intraday 60-minute intervals. Following the approach adopted, we selected the Closing Price of each timestep interval as the predictive target y. We select the data from the year 2000 to the year 2021 as the training data, used to optimize the model parameters and the data from 2021 to 2022 for the validation set used to optimize the hyperparameters. For the final test set evaluation we use the data from 2022 to 2023. This data split is the most established approach in SPP/SMP and followed for example in [32], [33] or [34].

The phenomenon of absent data points is frequently observed in intraday datasets, particularly exacerbated with finer time granularity. Conversely, interday datasets typically exhibit minimal occurrences of missing $x_i^{(t)}$ values. To address this issue, in cases where data points are absent, we impute the value by padding from the most recent existing $x_i^{(t)}$ entry, as done in [3]. All values are min-max normalized for each c_i and OHCL-feature respectively as done in previous studies such as [35].

[1]https://www.alphavantage.co/

TABLE I
Results for embedding based approach.

Model	SMP			SPP		
Δt	4	8	16	4	8	16
$\text{BERT}_{\eta=64}$	50.5	50.6	50.9	3.8	3.6	3.7
BERT	53.2	53.4	52.9	2.7	2.8	2.8
$\text{BERT}_{\text{base}}$	52.0	52.2	52.1	**2.6**	2.8	2.9
$\text{BERT}_{\text{large}}$	54.1	54.3	**55.1**	3.0	3.2	3.1
$\text{BERT}_{\text{pre-trained}}$	50.6	50.7	50.9	3.5	3.8	3.3

TABLE II
Results for Stock2Sentence based approach. The subscript specifies how E was initialized. Here the "Stock Embeddings" are from [11] and the "Stock Embeddings-CBR" are from [10] (for c_i for which there were no weights, random initialization was used). The S2V approaches were taken from [3] and retrained. For all approaches, it was empirically determined that the use of $\hat{E}$, $\theta_{\text{NORM}} = 100$ and $\bigstar$ as an addition operation works best. To validate this assumption, two more approaches were trained.

Model	SMP			SPP		
Δt	4	5	6	4	5	6
BERT	54.1	53.7	53.9	2.6	2.2	2.5
$\text{BERT}_{\text{Sock Embeddings}}$	50.7	50.7	50.8	3.6	3.6	3.7
$\text{BERT}_{\text{Sock Embeddings- CBR}}$	50.9	51.2	51.3	2.9	3.1	3.1
$\text{BERT}_{\text{S2V- SG}}$	53.4	53.6	53.6	**1.9**	2.0	2.0
$\text{BERT}_{\text{S2V-CBOS}}$	**54.3**	54.1	53.9	2.5	2.4	2.5
$\text{BERT}_{\bigstar=\text{Multiplication}}$	50.3	50.9	51.2	3.8	3.9	3.6
$\text{BERT}_{\bigstar=\text{Addition}}$	51.2	51.4	52.5	3.0	3.2	2.9

TABLE III
Results for Tokenization based approach.

Model	SMP			SPP		
Δt	3	4	5	3	4	5
BERT_{V}	51.2	51.1	**51.4**	3.7	3.4	**2.9**
$\text{BERT}_{\text{Default-BERT-Vocab}}$	50.8	50.9	50.8	3.6	3.6	3.7
$\text{BERT}_{\text{Default-BERT-Vocab, Pre-trained}}$	50.3	50.8	50.9	3.5	3.5	3.8

VI. Results

In Table I, we delineate the outcomes for the embedding-based approach; in Table II, we present the results for the Stock2Sentence-based approach; and in Table III, the outcomes for the tokenization-based method are detailed. Due to the scaling of the input length by $|C|$ for the Stock2Sentence approach and by the factor $|C| \cdot v$ for the tokenization-based approach, Δt must be selected significantly smaller for these two approaches than for the embedding-based approach. The hope is that the detailed correlation and relationship modeling between the individual stocks will nevertheless enable a good performance to be achieved. We report the Accuracy for SMP and the RMSE for SPP scaled by the factor 100 on the min-max normalized values. Each experiment was repeated five times.

It was observed that there is considerable variation in the performance across different companies, which supports the hypothesis that predictability may vary significantly among

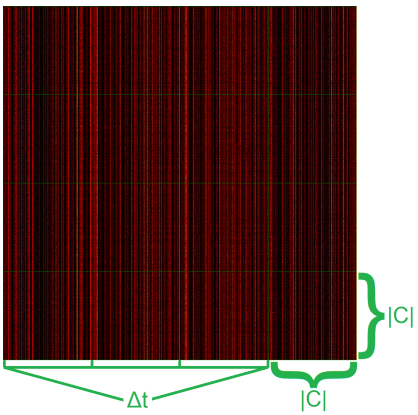

Fig. 3. Grad-CAM visualisation of attention scores for Stock2Sentence approach. The darker the color the higher the relevance (red) and the lighter (yellow) the lower. Please refer to the Appendix for detailed explanation of the used visualization algorithm.

stocks. This could have been one of the reasons why in [25] (see Section II) only single stocks were chosen for prediction. A real world application of ASMs could be based on a trading strategy in which investments are only made in stocks with high predictability / high accuracy values.

A. Investigating Time and Stock Relationships

As delineated in Paragraph III-C0a, the Stock2Sentence approach facilitates the representation of each of the $|C| \cdot \Delta t$ data points as an individual component of the input sequence. This methodological innovation permits a detailed examination of the internal model attention mechanisms with respect to each data point. Consequently, it becomes feasible to analyze the relative importance of each c_i at each t in relation to every other c_i at each t. A visualization is given in Figure 3.

In contrast to prior methodologies, which were restricted to investigating either the attention across different c_i as in [30], or across different t as in [36], the Stock2Sentence approach provides a more granular and interconnected analysis of attention dynamics, thereby enhancing our understanding of model behavior across both company and time dimensions.

VII. Discussion and Future Research

The embedding-based methodology aligns closely with conventional models, such as the approach detailed by Ding et al. in [24]. Consequently, the results obtained through this method are comparable, and in some instances, slightly inferior, as evidenced in the results presented in Table I. Notably, the effectiveness of the model appears to be positively correlated with its size. For instance, when η is set to 64, the model performance merely approaches the baseline threshold. Additionally, initializing the model with weights pre-trained for NLP tasks, (denoted as $\text{BERT}_{\text{pre-trained}}$) detrimentally impacts its performance. This reduction in efficacy is anticipated given the original training objectives of the pre-trained model, which differ significantly from the current application. Moreover, variations in Δt exert a minimal influence on the model's performance.

In the Stock2Sentence methodology, the initialization of E with pre-trained weights, from Dolphin et al. from [11] and [10], yields suboptimal results. This inefficacy is likely attributable to the small values of η used during the training of these models ($\eta = 16$ or $\eta = 20$). Future investigations could reconsider these approaches by re-training the models from [10] and [11] and potentially employing larger dimensions E. Additionally, strategies that do not incorporate $\hat{E}$ also demonstrate inadequate performance. A plausible explanation for this phenomenon is the scaling of $\hat{e}$, particularly when operations such as $\bigstar$ as multiplication are applied. This scaling may cause price information to excessively overlap with the semantic content encapsulated in $\hat{e}$, to the extent that the model becomes incapable of accurately understanding $\hat{e}$.

Tokenization-based methodologies exhibit generally poor performance. One potential issue is the increased input length that results from tokenizing regression values. This approach is especially ineffective when applied to pre-trained BERT models and utilizing the BERT vocabulary. A deficiency which may be attributed to the significant divergence of the tokenized stock data from the natural language corpus originally used to train the BERT models.

As previously emphasized, the primary goal of this research is to elucidate the intrinsic properties of these models and their capability to obviate the necessity for human analysis or domain-specific expertise. The preliminary results demonstrate the efficacy of the outlined models, which, in many instances, surpass the established baseline and exhibit substantial potential for generating profits. Given the innovative nature of these methodologies, there is considerable scope for enhancing their performance through further investigation, particularly concerning the tokenization-based approach. Prior studies, such as for example [17] and [18], suggest that incorporating trend and histogram inputs in LLMs may yield more effective results. Additionally, employing more intricate technical indicators for defining e or including fundamental data could further augment model performance.

One of the primary advantages of the Stock2Sentence (as well as for the tokenization approach) approach is its capability to eschew a fixed embedding dimension position for each c_i, unlike the majority of existing state-of-the-art SPP/SMP models. This flexibility enables the model to accommodate new entities, such as emerging companies, by dynamically integrating or excluding specific $x_i^{(t)}$ values, for instance during periods of zero trading or when data is missing. Additionally, the approach allows for the incorporation of contextual fundamental data, such as data derived from processing current news and social media. Notably, this last methodology warrants further exploration, as incorporating fundamental data often enhances model performance by embedding additional contextual information that may not be readily apparent from mere time series data alone.

Future research should prioritise the integration of further, publicly available, LLMs, such as GPT-2 [37], TransformerXL [38], T5 [39], and LLaMA [40], within the models used for the Stock2Sentence or tokenization methodologies. Moreover,

it is essential to adapt pre-training techniques such as MLM and NSP. These methodologies could facilitate the models' ability to discern inter-correlations among different c_i, which is regarded as a crucial strategy for enhancing the accuracy of SMP/SPP.

Currently, our research prioritizes predictive models, which are common in the domains of SPP and SMP, due to the complexity associated with generative approaches that forecast over extended future periods. However, generative models can still be employed as suggested by [3], where predictions are selectively generated for specific c_i (or t), i.e. $\hat{y}^{(t)} \notin \mathbb{R}^{|C|}$. This approach enables the model to restrict its predictions to prices or price movements where it possesses a significant degree of confidence and incorporate the idea of the different predictability of individual stocks.

VIII. Summary

In conclusion, we have delineated three methodologies to replace specific components of a conventional NLP model pipeline of LLMs tailored for regression data, exemplified in the context of SPP/SMP. We have expounded upon the benefits of our models and validated our methodologies through empirical testing of the BERT model. Furthermore, we have suggested avenues for future research focusing on adaptations of generative speech models and of the incorporation of textual data.

Acknowledgment

The stock data used for the models presented in this research was collected courtesy of research access kindly provided by Alpha Vantage [1].
We used the chatGPT AI [2] to improve the text in all sections of this work.
We thank Rian Dolphin who provided us with his trained weights for the initialization of E from his publications [11] and [10].

Appendix

To visualize the Stock2Sentence approach we adapt and modify the Grad-CAM visualization from [41]. The relevance map $R \in \mathbb{R}^{(|C|\cdot\Delta t)\times(|C|\cdot\Delta t)}$ visualizes the relevance of each of the $|C| \cdot \Delta t$ input points with respect to each other. It is calculated as

$$R = \bar{A} \text{ and } \nabla A := \frac{\partial \ (\frac{1}{|C|} \cdot \sum_{c_i \in C} \mathbb{H}(\hat{y}_i, y_i))}{\partial A} \qquad (6)$$

following the notation of [41].

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019.

[2] X. Zhang, Y. Zhang, S. Wang, Y. Yao, B. Fang, and P. S. Yu, "Improving stock market prediction via heterogeneous information fusion," *Knowledge-Based Systems*, vol. 143, p. 236–247, Mar. 2018.

[3] F. Voigt, K. Von Luck, and P. Stelldinger, "Assessment of the Applicability of Large Language Models for Quantitative Stock Price Prediction," in *Proceedings of the 17th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '24, (New York, NY, USA), p. 293–302, Association for Computing Machinery, 2024.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[5] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013.

[6] R. DeFusco, D. McLeavey, J. Pinto, and D. Runkle, *Quantitative Investment Analysis*. 2015. . John Wiley Sons. (Cited on pages 1 and 3).

[7] A. S. Wafi, H. Hassan, and A. Mabrouk, "Fundamental analysis models in financial markets – review study," *Procedia Economics and Finance*, vol. 30, pp. 939–947, 2015. IISES 3rd and 4th Economics and Finance Conference.

[8] X. Gabaix, R. Koijen, and M. Yogo, "Asset embeddings," *SSRN Electronic Journal*, 01 2023.

[9] B. Sarmah, N. Nair, D. Mehta, and S. Pasquali, "Learning embedded representation of the stock correlation matrix using graph machine learning," 2022.

[10] R. Dolphin, B. Smyth, and R. Dong, "Industry classification using a novel financial time-series case representation," 2023.

[11] R. Dolphin, B. Smyth, and R. Dong, "Stock embeddings: Learning distributed representations for financial assets," 2022.

[12] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Trans. Inf. Syst.*, vol. 37, mar 2019.

[13] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, "Modeling the stock relation with graph network for overnight stock movement prediction," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 4541–4547, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Special Track on AI in FinTech.

[14] G. Ang and E.-P. Lim, "Guided attention multimodal multitask financial forecasting with inter-company relationships and global and local news," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (S. Muresan, P. Nakov, and A. Villavicencio, eds.), (Dublin, Ireland), pp. 6313–6326, Association for Computational Linguistics, May 2022.

[15] X. Du and K. Tanaka-Ishii, "Stock embeddings acquired from news articles and price history, and an application to portfolio optimization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 3353–3363, Association for Computational Linguistics, July 2020.

[16] R. Kim, C. H. So, M. Jeong, S. Lee, J. Kim, and J. Kang, "Hats: A hierarchical graph attention network for stock movement prediction," 2019.

[17] X. Li, X. Shen, Y. Zeng, X. Xing, and J. Xu, "Finreport: Explainable stock earnings forecasting via news factor analyzing model," 2024.

[18] X. Yu, Z. Chen, and Y. Lu, "Harnessing LLMs for temporal data - a study on explainable financial time series forecasting," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track* (M. Wang and I. Zitouni, eds.), (Singapore), pp. 739–753, Association for Computational Linguistics, Dec. 2023.

[19] J. Li, R. Selvaraju, A. D. Gotmare, S. Joty, C. Xiong, and S. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," 2021.

[20] A. Pagliaro, "Forecasting significant stock market price changes using machine learning: Extra trees classifier leads," *Electronics*, vol. 12, no. 21, 2023.

[21] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *CoRR*, vol. abs/1704.02971, 2017.

[22] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, 1990.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[2] https://chat.openai.com/

[24] Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo, "Hierarchical multi-scale gaussian transformer for stock movement prediction," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 4640–4646, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Special Track on AI in FinTech.

[25] T.-T. Nguyen and S. Yoon, "A novel approach to short-term stock price movement prediction using transfer learning," *Applied Sciences*, vol. 9, no. 22, 2019.

[26] F. Voigt, "Adapting natural language processing strategies for stock price prediction." DC@KI2023: Proceedings of Doctoral Consortium at KI 2023, 2023.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[28] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018.

[29] S. Oymak, M. Mahdavi, and J. Chen, "Learning feature nonlinearities with regularized binned regression," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1452–1456, 2019.

[30] J. Yoo, Y. Soun, Y.-c. Park, and U. Kang, "Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, (New York, NY, USA), p. 2037–2045, Association for Computing Machinery, 2021.

[31] J. Liu, H. Lin, X. Liu, B. Xu, Y. Ren, Y. Diao, and L. Yang, "Transformer-based capsule network for stock movement prediction," in *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, (Macao, China), pp. 66–73, Aug. 2019.

[32] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2823–2824, 2015.

[33] E. Ramos-Pérez, P. J. Alonso-González, and J. J. Núñez-Velázquez, "Multi-transformer: A new neural network-based architecture for forecasting samp;p volatility," *Mathematics*, vol. 9, no. 15, 2021.

[34] W. Lu, J. Li, J. Wang, and L. Qin, "A cnn-bilstm-am method for stock price prediction," *Neural Computing and Applications*, vol. 33, pp. 4741–4753, May 2021.

[35] J. M.-T. Wu, Z. Li, N. Herencsar, B. Vo, and J. C.-W. Lin, "A graph-based cnn-lstm stock price prediction algorithm with leading indicators," *Multimedia Systems*, vol. 29, pp. 1751–1770, Jun 2023.

[36] J. Chen, T. Chen, M. Shen, Y. Shi, D. Wang, and X. Zhang, "Gated three-tower transformer for text-driven stock market prediction," *Multimedia Tools and Applications*, vol. 81, pp. 30093–30119, Sep 2022.

[37] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[38] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Annual Meeting of the Association for Computational Linguistics*, 2019.

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[40] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[41] H. Chefer, S. Gur, and L. Wolf, "Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 387–396, 2021.