MEANT: Multimodal Encoder for Antecedent Information

Benjamin Irving
Northeastern University
irving.b@northeastern.edu

Abstract

Multi-modal information plays an increasingly important role in the development of machine learning. It has been shown to positively impact performance on both uni-modal and multi-modal tasks. But information can do more then exist across modes— it can exist across time. How should we attend to temporal data that consists of multiple information types? This work introduces the MEANT model, which stands for Multi-modal Encoder for Antecedent information. The model is designed to jointly process image and text over a lag period of some number of days. The experiments were run on stock market data, a rich well of temporal information that can be split across modalities. The inputs used in the following paper consist of price, Tweet, and graphical data, employing a dual-stream encoder architecture followed by a temporal attention mechanism. The code is available at https://github.com/biirving/meant.

1 Introduction

Transformer architectures have shown to be performant across a wide range of language and image tasks [1][2], such as classification [2], machine translation [1], and object detection [4]. Attention based models have been shown to scale well to these problem spaces, allowing for competitive accuracy with lower compute requirements then other state of the art models [3].

The subsequent area of interest has been jointly processing these two modalities. Animals process the world through many senses, and large generative architectures should be primed to do the same. The expansion of processing different types of information together creates new possibilities, in solving new, novel problems, while also tackling preexisting challenges in innovative ways [10]. In recent months, multi-modal models have garnered serious momentum, with the release of large pretrained architectures such as Kosmos-1 [5] from Microsoft and GPT-4 [6] from OpenAI. These are the latest in a long line of models. Multi-modal processing refers mainly to data fusion[1]. How should text and image be processed together? Image and audio? Audio and text? Should the inputs be concatenated? Interleaved? Multi-modal models raise many questions, which have led to various methods being developed and deployed. One prominent strategy is early fusion, used by the ViLT [7], VL-BERT [8] and the Kosmos-1 [5] models, which all have achieved SOTA results on various datasets [7][8][5]. Early fusion refers to the idea of concatenating or interleaving embedded inputs before feeding them to an encoder [9]. Adversely, late fusion occurs when the two modalities are processed in separate input streams, and the two results are jointly processed by an MLP head or some other mechanism [9]. In the paper below, the MEANT model employs a hybrid strategy, where two input streams are used, but an encoder acts upon the concatenated information. Section (fill in) describes this in more detail.

Multi-modal information processing has already shown tremendous potential. Images are able to capture long-range dependencies from their inputs in a way that LLMs aren't, largely due to the

[1] Multimodal strategies have also targeted the attention component of models, through cross-attention and other methods. See [11]

Preprint. Under review.

way that patch embeddings work [12]. LLMs generally read their inputs with an attention mask, where the words are read from left to right or both backwards and forwards [1]. Generally, vision transformers don't employ the use of an attention mask, instead processing the "patches" in tandem with one another [2]. This method allows object detection models to classify things which may stretch from one corner of the image to the other [4]. Patch embeddings do bear some limitations that LLMs do not. The linear positional embeddings that language models employ allow for a richer understanding of immediate sequential relationships, how each word in a sentence builds on the ones around it to form a concrete, readable structure. These two principles can be combined in a powerful fashion. The MEANT model uses a vision transformer architecture find relationships in longer range information (i.e a graph of stock prices over a month) while it employs a language model to pick up more immediate trends (tweets pertaining to stock prices over a 5 day period). The stock market was chosen as a medium to explore the potential of temporal multi-modal information processing because it is a sequential problem, in which an individual can take advantage of short and long range information.

Stock movement prediction and analysis have drawn active interest in the research community for some time, largely due to the potential financial benefits [15]. Initially, research focused on time-series coupled with historic price data [15]. The problems of price prediction and trend identification proved to be enormously complex, seemingly erratic, the sum of the decisions of millions of people, made for a plethora of reasons, some consciously and others not. A human agent is not necessarily rational. Often, stocks are purchased in response to the change or trend in a technical indicator. On the other hand, buyers and sellers commonly make trades for reasons beyond the mathematical, among which include superstition, random impulse, or a growing trend in an online forum such as Twitter [17].

To comprehend such fluctuations, several studies have employed natural language techniques to financial markets, giving birth to the field of natural language-based financial forecasting (NLFF). Many of these studies have focused on public news [17]. Social media presents more time-sensitive information from active investors. Thus, for short term analysis, many researchers have begun to focus on Tweets for feature extraction [13], through which some have combined NLP techniques with traditional analysis on price data. Since Tweets often correspond to events as they happen in real time, such data is better suited for smaller windows, a 5 or 10 day period of price information. Combining the features extracted through NLP methods with price data has shown promising results. However, it is ineffective to feed the concatenated information to the model without encoding temporal dependencies. It is important to know when the tweets and prices occurred in relation to each other to avoid losing their expressive power. Closing prices and tweets which occur shortly before the target day will have a greater effect then the other auxiliary values.

Thus a key component in the model is the temporal attention mechanism. Each input contains a target day, preceded by a 5 day lag period, to account for the lack of long term foresight presented by tweets. The feature extraction of the Tweets themselves is processed by a traditional Transformer encoder [1]. However, when focusing on such a short time span, longer range information that may be of use when thinking about the viability of an asset are lost [17]. A meme stock may jump by some abhorrent amount, and subsequently blow up on Twitter or vice versa. This might lead the model to create a positive buy signal, only for the stock to crash back to earth in the next few days. Such a purchase might have been avoided if the model encoded long range information on top of the short range dependencies from the lag period. To account for this, MEANT encodes graph images which contain price information over a longer range, 26 days to be exact (the data set revolves around the MACD indicator, which is described in detail in Section 2).

The paper is organized as follows: Section 2 describes the data set and gives a clearer picture of the problem, which is followed by a description of the model itself in Section 3. Section 4 discusses the experiments and the results, followed by a conclusion which discusses challenges with the current model, and some directions that future research could take.

2 Data

The performance of the stock market is largely gauged through indexes [18]. One of the most popular of these is the Standard and Poor's Index (S&P) 500, which tracks 500 of the largest companies listed on exchanges in the United States. Many believe the S&P 500 indicates the strength of the market,

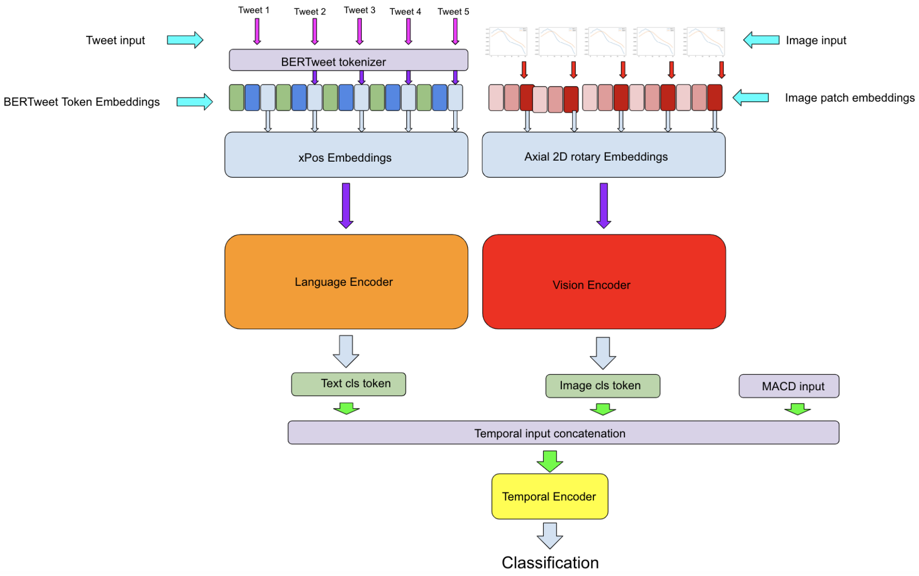

Figure 1: The MEANT model

and even the economy [18]. The companies listed on this index are generally stable, following trends[2] which are less erratic than smaller companies. In a sense, this makes their price fluctuations more relevant because they can be traced more easily. These pseudo-patterns give automated trading models a more significant chance to extract dependencies and form functional maps which may be of use in predicting future trends [16][17][18]. Thus the data-set used in the experiments below is comprised of information pertaining only to companies listed on the S&P 500. Momentum indicators are more interesting than basic price information for similar reasons. Whether by virtue of people believing in them more, or from actual mathematical relevance, they are less noisy, and thus tend towards more statistically significant patterns [18]. The data-set follows the Moving Average Convergence-Divergence (MACD) indicator, which is built on the back of Exponential Moving Average (EMA). The EMA is defined as follows:

$$EMA_t = (1 - \alpha) \cdot EMA_{t-1} + \alpha \cdot y_t \tag{1}$$

where t represents the day of EMA and y_t represents the closing price on that day, or in the case of the signal line, the MACD value on that day. α represents the degree of decrease; $\alpha = \frac{2}{t+1}$. Higher values, it can be observed, decrease more rapidly. The MACD consists of an MACD line, which is the difference between the fast EMA and the slow EMA (which are commonly set to 12 days and 26 days respectively), a signal line, which is the EMA of the MACD line itself (usally over a 9 day period) and a histogram, which is the difference between the MACD and the signal line. The MACD indicator was chosen because it has been shown to perform well against other indicators in terms of making accurate assertions about price directions [19]. The MACD for each S&P 500 company was taken over a year period from 4/10/2022 - 4/10/2023, along with some number of Tweets mentioning that company for each day in that period. These values were then clustered into 5 day lag period, so that each datapoint processed by the model consisted of 5 MACD vectors, 5 days of Tweets, and a graph of the MACD indicator over the long period from each of those days (5 images containing graphs of the MACD indicator over 26 days leading up to said day). A example of the graph inputs can be seen in Figure 2. These data points were classified as positive if the below equation held for the target day (the last day in the lag period):

$$M_{t-1} < S_{t-1} \wedge M_t > S_t \wedge M_t > 0 \tag{2}$$

The data set labeled values as 1 if the MACD was above 0 on the target day and crossed the signal line, and the price experienced an upwards trend in the succeeding week (higher lows). Otherwise

[2] Supposedly. The truth is that no ticker follows a stalwart, mathematically sound path. A chaos of too many factors. Yet larger companies typically remain less effected by random noise, due to the sheer number of individuals who believe in the strength of said asset, and thus buy and sell less frequently.

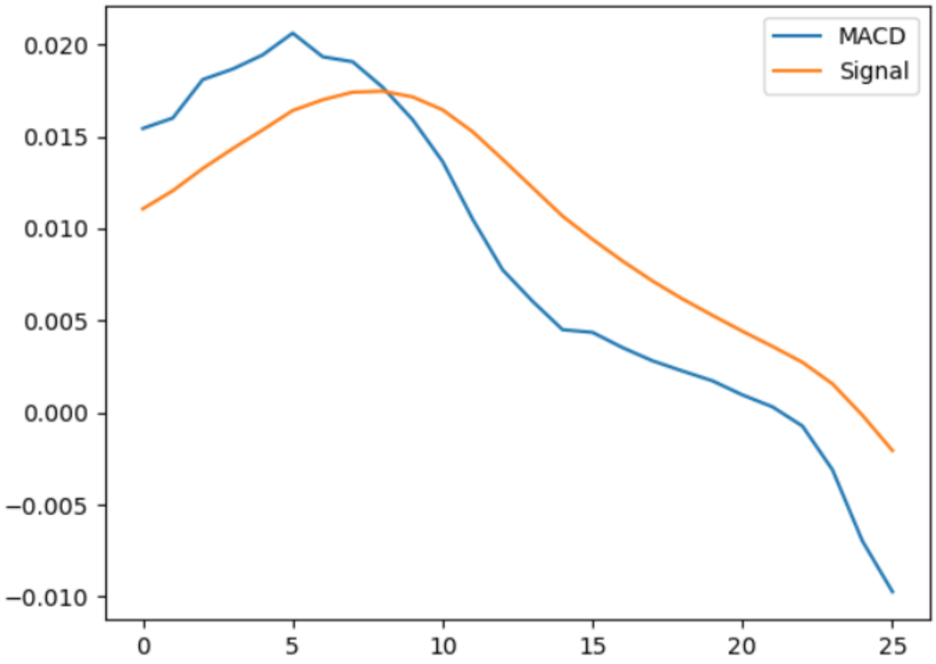

Figure 2: Example of an MACD long range graph, for Apple on 5/8/2023. It displays the signal and MACD lines over a 26 day period.

they were labeled as 0. This is a buy signal, meaning that 1 labels should be thought of stocks to buy, while 0 labels are stocks to leave alone.

3 MEANT

MEANT combines the advantages of image and language processing with temporal attention, in order to extract dependencies from multi-modal, sequential information. In this experiment, that data is the MACD indicator over a lag period, accompanied by Tweets and graphical data. Figure 1 displays the architecture. MEANT, similarly to most SOTA multi-modal models [10], is built atop the Transformer architecture.

3.1 The Transformer

The Transformer is an attention based architecture introduced by the Google team in 2017 [1]. The aim of the Transformer was to extract detailed features from sequential data without the use of recurrence, instead relying completely on attention. The main advantages of the Transformer model is that it can create long range dependencies between disparate parts of a given input, while also being resistant to severe effects from anomalous data. This is important to consider regarding the stock market, where patterns could be found across many days, or even weeks. Furthermore, the architecture is extremely parallizable, and can utilize distributed computation over many GPUs to fine-tune its weights at a much faster pace. Of primary interest, regarding MEANT, is the attention mechanism.

3.2 Attention

The attention mechanism takes its biological inspiration from the fovea, the human field of vision is centered. In the mathematical context, it can be described as mapping a set of key-value pairs to an output, where the query, key, and values are also represented by vectors. Specifically, I use the mechanism known as scaled-dot product attention, which can be defined as follows. Let Q, K, and V be the query, key, and value matrices, respectively, where each matrix has dimensions $n \times d$, and d is the dimension of the query, key, and value vectors, and n represents the number of input tokens to be processed. The scaled dot product attention function is then defined as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{3}$$

In this equation, the dot product of the query matrix and the transpose of the key matrix is calculated, and the result is divided by the square root of the dimensionality of the query and key vectors ($\sqrt{d}$). This scaled dot product is then passed through the softmax function, which normalizes the values to sum to 1, creating a set of weights that can be used to scale the values in the value matrix. Finally, the scaled values from the value matrix are multiplied by the weights to produce the final output of the attention function.
However, with a single attention function, we are limited in the parts of the input that the model can build specific dependencies for. Thus, we employ multiple scaled dot-product attention mechanisms in parallel, in a process known as multihead attention, which is defined as:

$$Multihead(Q, K, V) = concat(head_1, \dots, head_h)W^O \tag{4}$$

where $head_i$ is the output of the i-th attention head, which is defined as:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

Multihead attention creates Q, K, and V matrices which become specialized at processing vectors from different parts of the input. MEANT uses 5 attention heads to process the Tweet information. The dimension of the embedded Tweets is $\mathbb{R}^{768}$, so to capture detailed feature information the multi-head attention mechanism uses a dimension of $\mathbb{R}^{768}$ as well.

3.3 Transformer Encoder

In the original transformer paper, the model is built on top of an encoder-decoder architecture [1]. MEANT is an encoder-only model, similar to BERT [21]. The transformer stacks the attention mechanism with linear layers to extract relevant features from the input. Between the 2 parts of the encoder, and before the output, there is a residual connection, meaning that the input to that portion of the architecture is fed through added with the original input. This is done to alleviate the vanishing gradient problem [23]. The encoder structure employed by both the language and vision pipelines is inspired by the Magneto model [22] from Microsoft which makes use of sub layer normalization, meaning that a layer norm is interleaved between the attention and linear layer components of the encoder. This architecture was chosen because it has been shown to be successful on a wide variety of uni-modal and multi-modal problems [5][22]. The general encoder structure can be seen in Figure 3.

3.4 Token and Patch Embeddings

Before being fed to the attention mechanism, the two input types have to be prepared for processing using two different embedding strategies. In many LLMs, text is tokenized before being fed into the embeddings. Tokenization refers to mapping each word, or piece of a word, to some encoded vector in a pretrained codex. These tokens are then processed by an embedding matrix, which projects the token vectors into some high dimensional space where attention can extract further information. The Tweets in MEANT are first tokenized using a pre-trained BERTweet tokenizer [20], which is just a BERT [21] tokenizer that has been finetuned on a Tweet dataset. The word embeddings used in this model also come from BERTweet [20].

Images are not processed with tokenization. The raw pngs are first transformed into tensors of rgb values and reshaped to a manageable size. MEANT uses image sizes of 4 x 224 x 224, where 4 represents the number of channels and the subsequent dimensions are the height and width respectively.

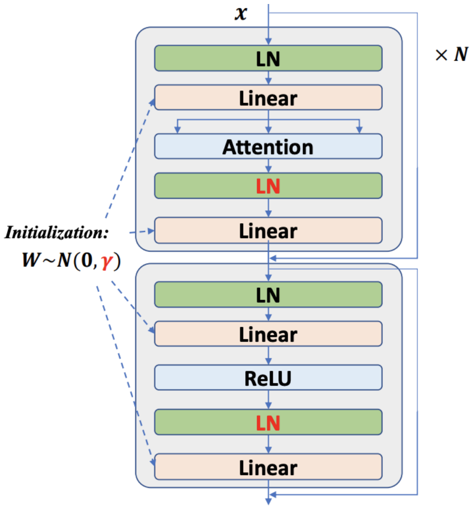

Figure 3: The encoder structure from Magneto [22]

These vectors are then broken down using a patch embedding strategy, from the original vision transformer [2]. A patch embedding just breaks down an image into patches, somewhat similar to the way in which CNNs process an image. The image $x \in \mathbb{R}^{C \times H \times W}$ becomes $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where C is the number of channels, N is the number of patches, and P is the patch dimension. This input is then passed to the vision encoder. Both of the encoders use a [cls] token, which is simply a tensor concatenated to each input before being fed to the attention mechanism. The [cls] token learns a representation of the entire input, and this tensor is what is eventually evaluated in the Temporal attention step. The [cls] token is meant to reduce the computational burden of the model, and allow for a better evaluation of the entire input. Both the language and vision encoders train a [cls] token in dimension $\mathbb{R}^{L*D}$, where L represents the lag period and D represents text or image dimension.

3.5 Positional Encoding

While both the image and language encoders draw from the same general structure, there is an important difference in their positional embedding strategy. Positional encodings inject information about the relative and absolute positions of the tokens in the inputs [1]. They allow the model to understand where certain words typically fall in relation to one another, or how objects in images relate to one another according to their location. In the original transformer paper, they used cosine functions at different frequencies to produce positional information that they just added on top of the embedded inputs, before feeding them into the attention mechanism. The MEANT model uses rotary positional embeddings [24], which are applied directly to the query and key matrices in the attention mechanism. Instead of layering the positional information on top, a rotation matrix is multiplied with the key and query matrices to create relative positional information. Rotary embeddings encode the absolution position with a rotation matrix. This rotation matrix then multiplies key and value matrices of every attention layer with it to inject relative positional information at encoder layer in the model. When encoding relative positional information into the inner product of the j-th key and the i-th query, we would like to formulate the function in a way that the inner product indicates only the relative position. Rotary Position Embedding (RoPE) [24] utilizes the rotation operation in Euclidean space and describes the relative position embedding as a rotation of the feature matrix, where the angle of rotation is proportional to its position index. All of this may seem a bit complex, but one can walk through it using the equations below.

To rotate a 2-D vector c counterclockwise by some θ, c can be multiplied by the following matrix:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{6}$$

Now consider the query and key matrices, represented by $f_q(x_m, m)$ and $f_k(x_n, n)$ respectively. These can be jointly defined by the following equation, where the W matrices are the weights of the queries and keys, and the x values represent the input to the attention mechanism. The general intuition is that one can rotate the affine-transformed embedding vector by some amount of angle multiples of its position index.

$$f_{q,k}(x_m, m) = \begin{pmatrix} \cos(m\theta) & -\sin(m\theta) \\ \sin(m\theta) & \cos(m\theta) \end{pmatrix} \begin{pmatrix} W_{q,k}^{(11)} & W_{q,k}^{(12)} \\ W_{q,k}^{(21)} & W_{q,k}^{(22)} \end{pmatrix} \begin{pmatrix} x_m^{(1)} \\ x_m^{(2)} \end{pmatrix} \tag{7}$$

This rotation can be generalized to some arbitrary number of dimensions, by expanding the rotational matrix as follows.

$$R^d_{\Theta,i} = \begin{bmatrix} \cos i\theta_1 & -\sin i\theta_1 & 0 & 0 & \dots & 0 & 0 \\ \sin i\theta_1 & \cos i\theta_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \cos i\theta_2 & -\sin i\theta_2 & \dots & 0 & 0 \\ 0 & 0 & \sin i\theta_1 & \cos i\theta_1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \cos i\theta_{d/2} & -\sin i\theta_{d/2} \\ 0 & 0 & 0 & 0 & \dots & \sin i\theta_{d/2} & \cos i\theta_{d/2} \end{bmatrix} \tag{8}$$

As this pertains to the MEANT model, the language and vision encoder use different θ values to rotate the respective inputs. The language encoder uses the xPos variant of the rotary embeddings, so that the thetas are assigned according to the following equation.

$$g_\zeta[n] = \frac{1}{d/2} \sum_{i=0}^{d/2} \cos(n\theta_i)\zeta^n \tag{9}$$

ζ_n can be defined as follows:

$$\zeta_n = \frac{i}{\frac{d}{2} + \gamma} (1 + \gamma), \quad \gamma \in [0, 1] \tag{10}$$

d represents the dimension of the input to the model. The vision encoder uses 2D axial rotary embeddings, which simply means that θ is set according to the following equation:

$$\theta_i = i * floor(d/2) * pi \tag{11}$$

where d is again the dimension of the input.

3.5.1 Temporal Attention

After both the Tweets and the graphs have been processed, [cls] tokens of each modality are then concatenated to the MACD information from that 5 day lag period.

$$t_i = [w, i, m] \in \mathbb{R}^{lxdim_t} \tag{12}$$

l is the lag period, while dim_t is the temporal dimension, which is the sum of the language, image, and MACD dimensions. In the vanilla implementation of the MEANT model, the temporal dimension was 1540. The [cls] tokens are trained to become reasonable representations of the entire input over time, and are thus more easily processed by the classification head then the entire input [2]. In the case of MEANT, the outputs are not directly fed into a classification head, but are instead passed to a temporal mechanism. At this point in the pipeline, relevant image and text features have been extracted for each trading day in relation to themselves, not to one another. The temporal attention mechanism focuses on the target day, and its relationship to the preceding days. The purpose of the model is to extract a pattern from the preceding days, to identify future MACD crossings which may result in a profitable push. MEANT does this in an incredibly simple manner, to great effect. The query matrix in the attention mechanism only acts upon the target day, so that all of the keys and values are processed in relation to the target day.

$$Attention(Q, K, V) = softmax\left(\frac{Q_t K^T}{\sqrt{d}}\right) V \tag{13}$$

The [cls] tokens have done the majority of the work in extracting relevant features, and the temporal mechanism only need process a simple computation to find a meaningful temporal pattern. The temporal encoder is structured identically to the image and language encoders in all other aspects. There are positional temporal embeddings layered on top, but these are simply a learned parameter vector, not rotary embeddings. The output of the temporal encoder is then processed by the MLP head, which produces a classification.

4 Experiments

The experimental goal with MEANT was to measure its ability to extract meaningful information from multiple modalities over time. MEANT was ran on the data-set described above, with some variations to test the importance of different components of the model.

4.1 Setup

The data set comprised of the MACD values of S&P 500 companies from 4/10/2022 - 4/10/2023, along with Tweets and graph images for each of those days. The data was split 70/30, into training and test categories. The model was built using Pytorch [26], and ran on a V100-pcie GPU. Many papers have previously used the Matthews Correlation Coefficient (MCC), which can be defined as follows

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{14}$$

MCC takes true positives, false positives, true negatives, and false negatives into account. It measures the quality of binary predictions, returning a value between -1 and 1. -1 represents a completely incorrect prediction, 0 and random prediction, and 1 a correct prediction. MCC is more accurate then basic accuracy, because it takes false negatives and false positives into account. The model was measured in MCC, accuracy, and AUROC. In terms of the base-model specifics, MEANT uses 5 stacked text encoders, 5 stacked image encoders, and one temporal attention mechanism. The deeper-model had 30 stacked text encoders and 30 stacked image encoders. The model-without Tweets used 5 stacked image encoders, while the model without images used 5 stacked language encoders. Each experiment ran for 50 epochs, besides the large model which was ran for 100 epochs.

4.2 Performance

Table 1: Experiment Results

	Accuracy %	MCC	AUROC
MEANT-base	94.37	0.68	0.72
MEANT-large	98.57	0.83	0.85
MEANT-noTweet-base	62.37	0.09	0.53
MEANT-noTweet-large	65.38	0.12	0.57
MEANT-noVision	84.29	0.37	0.64

Looking at the table above, the most performant model was the large MEANT with both the language and image encoder. It achieved an accuracy of 98.57 %, an MCC of 0.83, and an AUROC score of 0.85. The MEANT-base achieved 94.37 %, with an MCC of 0.68 and an AUROC of 0.72. Figure 4 shows the loss curve, which has some interesting micro-spikes, which are likely caused due to the erratic nature of the Tweets. They were scraped at random, the only relation to one another being that they mentioned an S&P 500 company. Such method most certainly produced some random injections, which caused small amounts of noise to appear in the loss curve. There was a notable performance drop off without using Tweets, belying the importance of the short-range information. The MEANT-base model with no language input only achieved an accuracy of 62.37 % and a AUROC score of 0.53, which is close to random guessing. Looking at Figure 5, the loss function was all over the place, with large spikes in both the large and base models. This discrepancy emphasizes the importance of the semantic data in the effectiveness of the model. The Tweets play a huge role, appearing to be more important to the overall performance then the image data, which encodes the long range price information. This is an extremely surprising result, contrary to the assumptions of the traditional technical analyst. It indicates that the sentiment found on Twitter is more of an accurate guide to short-term momentum classification than the MACD values.

The model was then compared to other relevant Multi-modal architectures. Looking at Table 2, MEANT outperformed them all. Their disadvantage was the lack of temporal support. This meant that they were only able to process the target days in the lag period, and not extract dependencies in the days leading up to the final classification. The most comparative performance to the MEANT-base model was ViLT, which achieved 85.94% classification accuracy. The AUROC of this model was only 0.61, which isn't much better than random guessing. These results suggest the importance

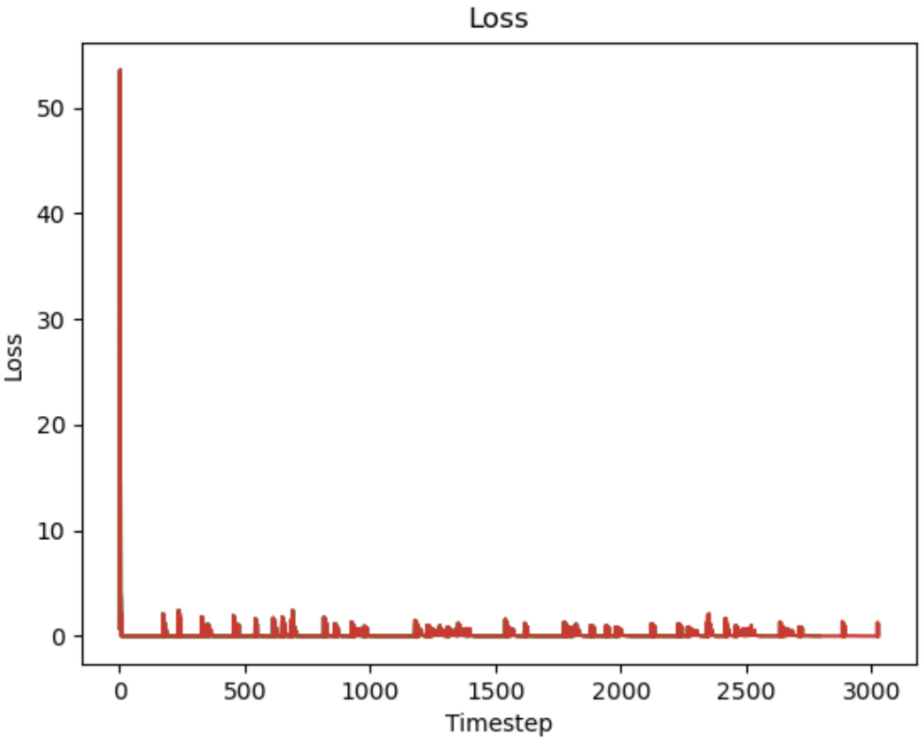

Figure 4: A performant run.

of the temporal component of the architecture, and how much it contributed to the success of the MEANT-base model.

Table 2: Experiment Results

	Accuracy %	MCC	AUROC
MEANT-base	94.37	0.68	0.72
VL-BERT	82.37	0.53	0.58
ViLT	85.94	0.59	0.61
ViLBERT	81.28	0.57	0.60

5 Conlusion

MEANT proved to be a highly successful architecture, which outperformed comparable architectures on my data set. The Tweet information proved to be the most significant contributor to accuracy. There could be many reasons for this, one of which could be that the casual investor looks to social media when making a decision. Another could be that many quantitative algorithms performing large amounts of trades take Tweets into account. Whatever the case, semantic feature extraction is clearly a potent tool in the analysis of the stock market.

Temporal encoding also proved to be very important. The other comparative models failed to stack up against MEANT, with the difference being that they did not process days in the lag period. This could be because of less semantic information from the Tweets, or because of patterns that the model learned from lag period.

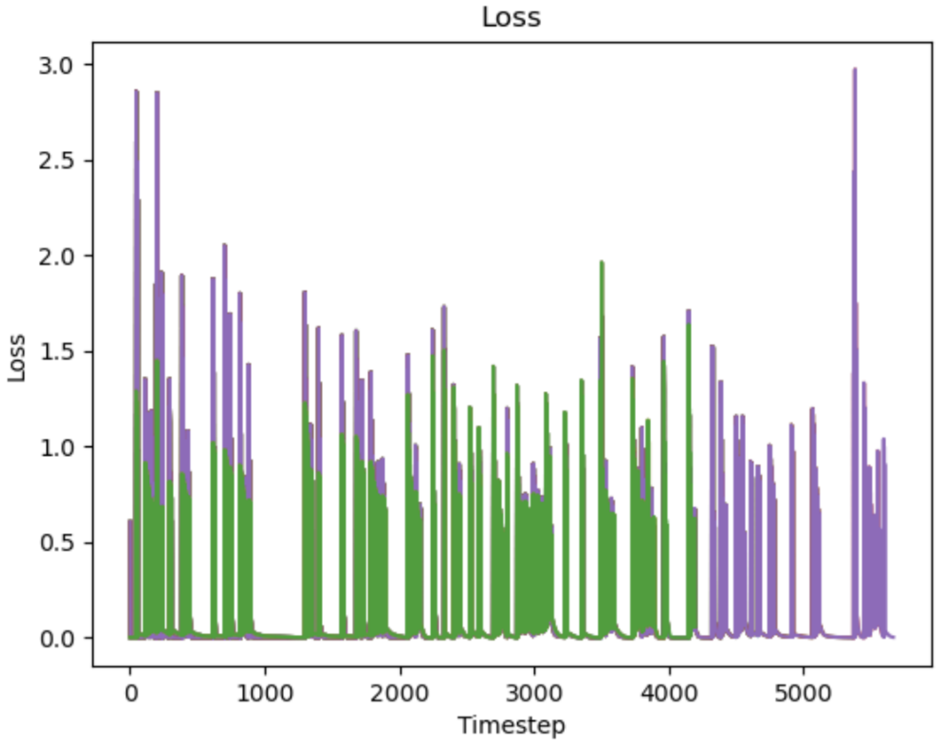

Figure 5: A non-performant run. The green spikes represent the loss for the large model, while the purple spikes show the loss for the base model.

5.1 Challenges

The biggest challenge in the creation of the MEANT model was the temporal architecture. There were previous iterations that did not allow the loss function to stabilize. What finally allowed success was the incorporation of attention to extract temporal dependencies.

The data set also posed some problems. As alluded to, scraping the Tweets in an uncontrolled fashion resulted a high degree of chaos and noise to be injected. Multiple companies are often mentioned in Tweets, tacked on to the end to enter into as many recommended streams as possible for high visibility. Furthermore, it was highly unbalanced, with a huge discrepancy between the positive and negative examples. The data needs to be cleaned before the model is pretrained for practical deployment.

5.2 Extensions

MEANT has shown to perform significantly better than comparative models on multi modal momentum stock data. Future steps would be to explore performance in other areas, such as medical data or more complex problem spaces such as robotics, where multi modal processing has serious potential [10]. In terms of the stock market space, MEANT was trained to be a buy-side classifier, focusing on identifying stocks for purchase. To complete the system, the next step would be to train a sell-side classifier, and deploy it on the markets with money to trade. A policy gradient mechanism could be layered over the top, to weight the success of decisions to buy and sell and securities. Such a complete system could prove to be a formidable force in automated trading.

References

[1] Ashish Vaswani & Noam Shazeer & Niki Parmar & Jakob Uszkoreit & Llion Jones & Aidan N. Gomez & Lukasz Kaiser & Illia Polosukhin (2017). Attention Is All You Need. In *NIPS* 2017

[2] Alexey Dosovitskiy & Lucas Beyer & Alexander Kolesnikov & Dirk Weissenborn & Xiaohua Zhai & Thomas Unterthiner & Mostafa Dehghani & Matthias Minderer & Georg Heigold & Sylvain Gelly & Jakob Uszkoreit &Neil Houlsby (2021). An image is worth 16x16 words: Transformers for image recognition at scale. ICLR, 2021.

[3] Surafel Melaku Lakew & Mauro Cettolo & Marcello Federico (2018). A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation. CoRR, 2018.

[4] Hangbo Bao &Li Dong &Furu Wei (2021). BEiT: BERT Pre-Training of Image Transformers. CoRR, 2021.

[5] Shaohan Huang & Li Dong & Wenhui Wang & Yaru Hao & Saksham Singhal & Shuming Ma & Tengchao Lv & Lei Cui & Owais Khan Mohammed & Barun Patra & Qiang Liu & Kriti Aggarwal & Zewen Chi & Johan Bjorck & Vishrav Chaudhary & Subhojit Som & Xia Song & Furu Wei. Language Is Not All You Need: Aligning Perception with Language Models. arXiv:2302.14045 (2023)

[6] OpenAI. GPT-4 Technical Report. arXiv:2303.08774 (2023)

[7] Wonjae Kim & Bokyung Son & Ildoo Kim. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. arXiv:2102.03334 (2021)

[8] Weijie Su & Xizhou Zhu & Yue Cao & Bin Li & Lewei Lu & Furu Wei & Jifeng Dai. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. arXiv:1908.08530 (2020)

[9] Gadzicki & Konrad & Khamsehashari &Razieh & Zetzsche & Christoph. Early vs Late Fusion in Multimodal Convolutional Neural Networks. 2020 IEEE 23rd International Conference on Information Fusion (FUSION)

[10] Paul Pu Liang & Yiwei Lyu & Xiang Fan & Zetian Wu & Yun Cheng & Jason Wu & Leslie Chen & Peter Wu & Michelle A. Lee & Yuke Zhu & Ruslan Salakhutdinov & Louis-Philippe Morency. MultiBench: Multiscale Benchmarks for Multimodal Representation Learning. arXiv:2107.07502 (2021)

[11] Jiasen Lu & Dhruv Batra & Devi Parikh & Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. arXiv:1908.02265 (2019)

[12] Namuk Park & Songkuk Kim. How do Vision Transformers Work? arXiv: 2202.06709 (2022)

[13] Araci. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv:1908.10063 (2019)

[14] Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. arXiv: 1712.02136 (2017)

[15] Hirotugu Akaike. Fitting autoregressive models for prediction. Annals of the Institute of Statistical Mathematics, volume 21, pages 243-247.

[16] Huynh &Huy &Dang &L. Minh &Duong &Duc. A New Model for Stock Price Movements Prediction Using Deep Neural Network. (2017)

[17] Wu, Huizhe & Zhang, Wei & Shen, Weiwei & Wang, Jun. Hybrid Deep Sequential Modeling for Social Text-Driven Stock Prediction. Association for Computing Machinery. (2018)

[18] Goetzmann & William N. & Massimo Massa. Index Funds and Stock Market Growth. The Journal of Business, vol. 76, no. 1, 2003, pp. 1–28.

[19] Pat Chiong. A comparative study of the MACD-base trading strategies: evidence from the US stock market. arXiv:2206.12282 (2022)

[20] Dat Quoc Nguyen & Thanh Vu & Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. arXiv:2005.10200 (2020)

[21] Jacob Devlin & Ming-Wei Chang & Kenton Lee & Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv: 1810.04805 (2019)

[22] Hongyu Wang & Shuming Ma & Shaohan Huang & Li Dong & Wenhui Wang & Zhiliang Peng & Yu Wu & Payal Bajaj & Saksham Singhal & Alon Benhaim & Barun Patra & Zhun Liu & Vishrav Chaudhary & Xia Song & Furu Wei. Foundation Transformers. arXiv: 2210.06423 (2022)

[23] Razvan Pascanu & Tomas Mikolov & Yoshua Bengio. On the difficulty of training Recurrent Neural Networks. arXiv: 1211.5063 (2012).

[24] Jianlin Su and Yu Lu & Shengfeng Pan & Bo Wen & Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv: 2104.09864 (2021)

[25]Yutao Sun & Li Dong & Barun Patra & Shuming Ma & Shaohan Huang & Alon Benhaim & Vishrav Chaudhary & Xia Song & Furu Wei. A Length-Extrapolatable Transformer. arXiv: 2212.10554 (2022)

[26] Paszke, Adam & Gross, Sam & Chintala, Soumith & Chanan, Gregory & Yang, Edward & DeVito, Zachary & Lin, Zeming & Desmaison, Alban & Antiga, Luca & Lerer, Adam. Automatic differentiation in PyTorch (2017). In *NIPS* 2017.