Webinar

Recent Advancement of Financial Machine Learning with an Emphasis on Large Language Models

Welcome

We will begin promptly at 10 AM ET.

If you are unable to hear the speakers, please let us know in the chat box.

You may enter your questions in the Q&A, we will address them at the end of the presentation. You can find a copy of the slide deck and recording of this webinar: www.fdpinstitute.org/webinars

FDP INSTITUTE®

Financial Data Professional Institute

FDP Institute provides world class training and education to financial professionals to meet the accelerating needs of digital transformation in the industry.

Introductions

Hossein Kazemi, PhD, CFA
Senior Advisor,
CAIA Association &
FDP Institute

Nino Antulov-Fantulin, PhD
Head of Research & Development
Aisot Technologies AG

Petter Kolm, PhD
Clinical Professor of Mathematics
Director of Mathematics in
Finance M.S. Program
NYU Courant

Today's Topic:

Recent Advancement of Financial Machine Learning with an Emphasis on Large Language Models

Advances of ML Approaches for Financial Decision Making & Time Series Analysis – Large Language Models

Nino Antulov-Fantulin (nino@aisot.ch)

Petter Kolm (petter.kolm@nyu.edu)

FDP Webinar

Scope

- Overview of applications for large language models (LLMs) in investment management – (5 mins)
- Deep dive into LLMs – (30 mins)
  - Key fundamentals of LLMs
  - Example: LLMs for financial sentiment
- Use cases from the literature – (5 mins)
- Some considerations and limitations – (5 mins)
- Q&A – (15 mins)

Overview of Applications for LLMs in Investment Management

Many Applications for LLMs in Investment Management

- Generate new revenue sources by developing innovative investment products and strategies that uncover unique insights and market opportunities
- Increase productivity and decrease costs through automation of workflows, reporting and other routine tasks
- Improve customer service through personalization and targeted engagement

Financial Decision Making and Investing

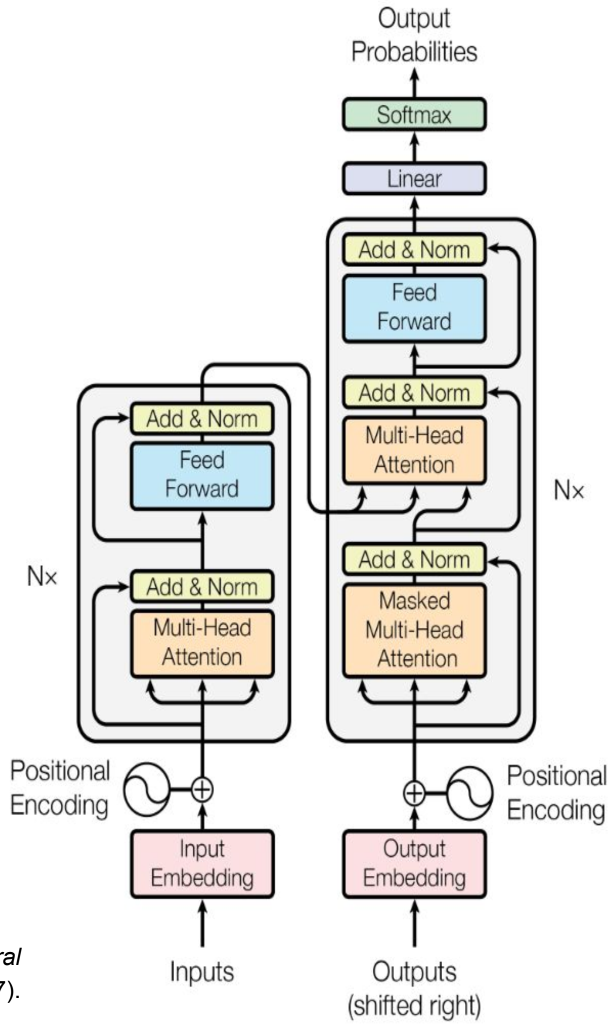
- **Sentiment analysis:** Analyze news articles, social media posts, and earnings call transcripts to gauge market sentiment and investor perception about specific stocks, sectors, or market trends
- **Forecasting, predictive analytics and strategy development:** Generate forecasts and predictions for various financial metrics such as stock prices, market trends, and macroeconomic indicators
- **Algorithmic and event-driven trading:** Integrate into algorithmic trading systems to analyze textual data and news sentiment in real-time to make automated trading decisions
- **Portfolio management:** Assist portfolio managers in generating investment ideas and performing industry/sector analyses
- **Risk management:** Analyze textual data related to risk factors, regulatory filings, and market events to identify potential risks and vulnerabilities
- **Alternative data analysis:** Process and label unstructured alternative datasets (satellite imagery, social media posts, and web traffic data), perform entity recognition

Automation & Personalization

- **Summarization:** Summarize financial news articles, press releases, financial reports (e.g. 10Ks & 10Qs) and other reports
- **Financial reporting and analysis:** Generate textual reports and analyses such as earnings reports, annual reports, and financial statement analysis
- **Compliance and regulatory reporting:** Automate compliance tasks by analyzing regulatory documents, monitoring regulatory changes, and generating compliance reports
- **Customer service and communication:** Enhance customer service in investment management firms by generating personalized investment recommendations, answering client inquiries, and providing content. Automate client communication and engagement through chatbots and virtual assistants

Deep Dive into LLMs

- Roadmap for LLMs
    - Architecture: encoder-decoder, encoder-only, decoder-or
    - Components: **Embeddings***, (Masked Multi) **Attention mechanism***, FFNN, Layer Norm
    - Training: unsupervised pre-training*, reinforcement learn from human feedback
    - Fine-tuning: slanted triangular learning, gradual unfreezing, discriminative fine-tuning
    - Evaluation
    - Limits of LLM*
    - Use-case for finance: Portfolio construction*
    - Additional References*
    - What is next?
- In this talk we will cover only basics of *



Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

What are Large Language models?

What kind of Neural Network are we talking about ?

>> class of Generative AI models!

What does that mean?

>> a Neural Network that can generate next word based on some previous text

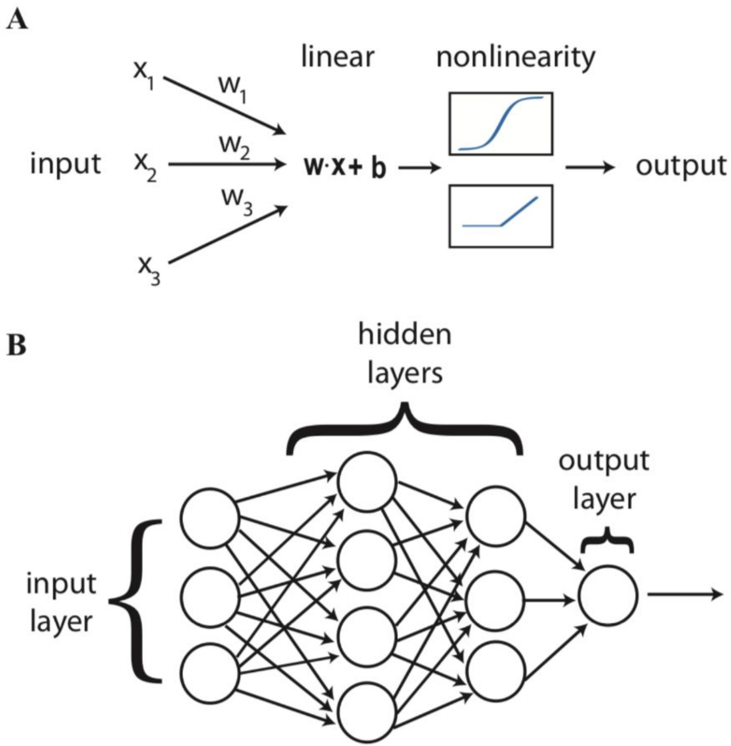
$P(\text{next word} \mid \text{previous text})$

Feed Forward Neural Network

- **We assume that audience is familiar with basic FFNN**
- $f(x)=\sigma(Wx+b)$
- NNs are universal function approximators for continuous functions on bounded support

K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural networks 2, 359 (1989)

P. Kidger and T. Lyons, Universal approximation with deep narrow networks, in Conference on learning theory (PMLR, 2020) pp. 2306–2327.



*Mehta, Pankaj, et al. "A high-bias, low-variance introduction to machine learning for physicists." *Physics reports* 810 (2019): 1-124.

Neural Networks

Wait …. how can NN answer our questions in chat dialog, or predict sentiment for financial news?

>> If we can transform text to vectors of numbers that would be a good starting point
e.g. mapping “NYSE” → [0,0,0,...,1], “returns” → [1,0,0,...,0]

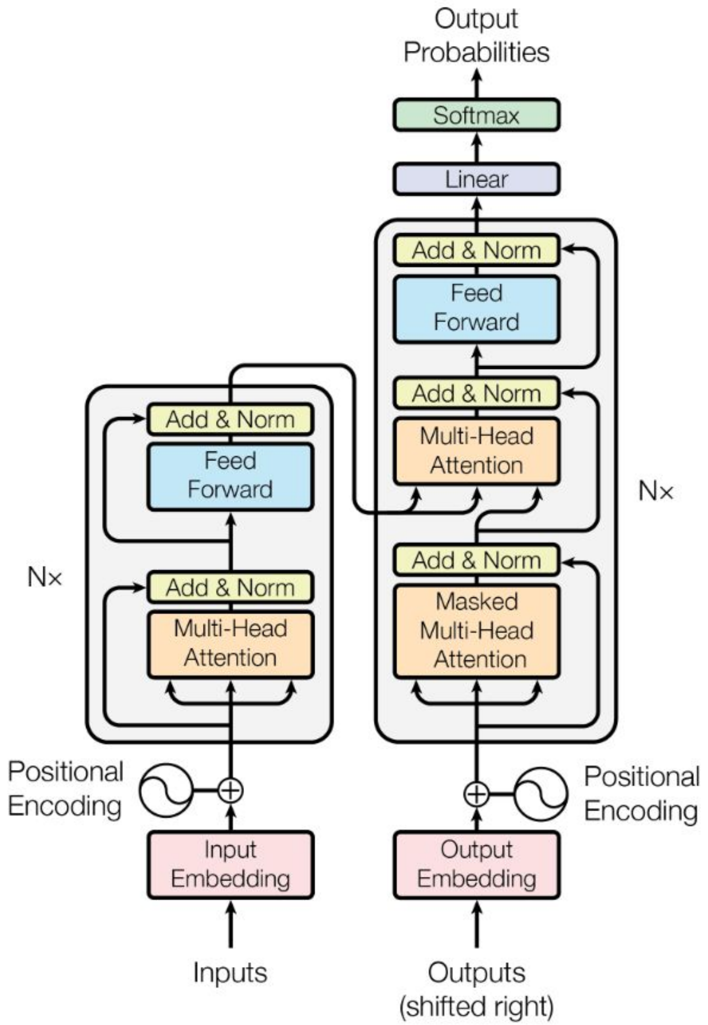
But how do we do that transformation?

LLM Architecture

The NN we are talking about is specially designed to work on text-like data. It is called a Transformer

Why so complicated?

Let's analyse it

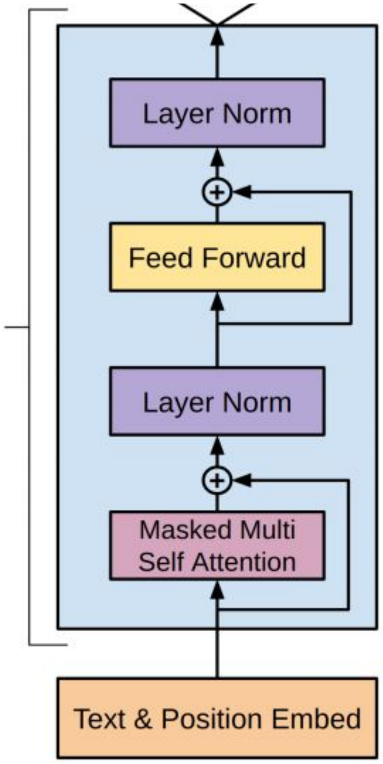


Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Focus on the GPT-based Architecture

OpenAI's "GPT-n" series

Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus:[27] 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, 2018[6]	30 days on 8 P600 GPUs, or 1 petaFLOP/s-day.[6]
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, 2019 (initial/limited version) and November 5, 2019 (full version)[28]	"tens of petaflop/s-day",[29] or 1.5e21 FLOP.[30]
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion[31]	499 Billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, 2020[29]	3640 petaflop/s-day (Table D.1 [29]), or 3.1e23 FLOP.[30]
GPT-3.5	Undisclosed	175 billion[31]	Undisclosed	March 15, 2022	Undisclosed
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.[26]	Undisclosed. Estimated 1.7 trillion[32]	Undisclosed	March 14, 2023	Undisclosed. Estimated 2.1e25 FLOP.[30]



Let’s Dig Inside of GPT “Neural Brain”

Let’s play a bit with an LLM. How?

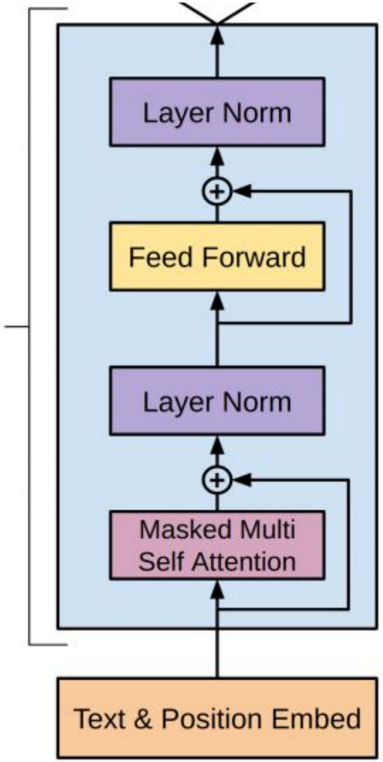
Using web-based solutions is not enough!

We want to execute it on a local GPU machine and run the code with a pre-trained model, so that we can control every aspect of its execution

>> loading weights from pretrained gpt: gpt2-xl

>> forcing vocab_size=50257, block_size=1024, bias=True

>> number of parameters: 1.555 Billion



Let’s Dig into How the LLM Processes Inputs

Example:

Q: Which amount of the portfolio should be in cryptocurrency bitcoin?

A: I think that the answer to this question is a bit complicated. It is very important to understand the current state of the cryptocurrency space and how the market is evolving.\n\nBitcoin is the most well-known cryptocurrency and the one most likely to be used for transactions in the near future. It is also the most popular, with a market value of $7….

Let’s Dig into How the LLM Processes Inputs

Example:

Q: Last week’s collapse of cryptocurrency exchange FTX has left many investors uncertain about whether they can recover their funds. On Friday (November 11), FTX filed for bankruptcy. What is the effect of this event on the equity market?\n\n

A: The market is not yet in a state of panic. The market has not been in a state of panic since the crash of Mt. Gox, which was the first bitcoin exchange. The market is still recovering from that. There is still a lot of volatility in the market. The price of bitcoin is still very volatile, and there is a lot ….

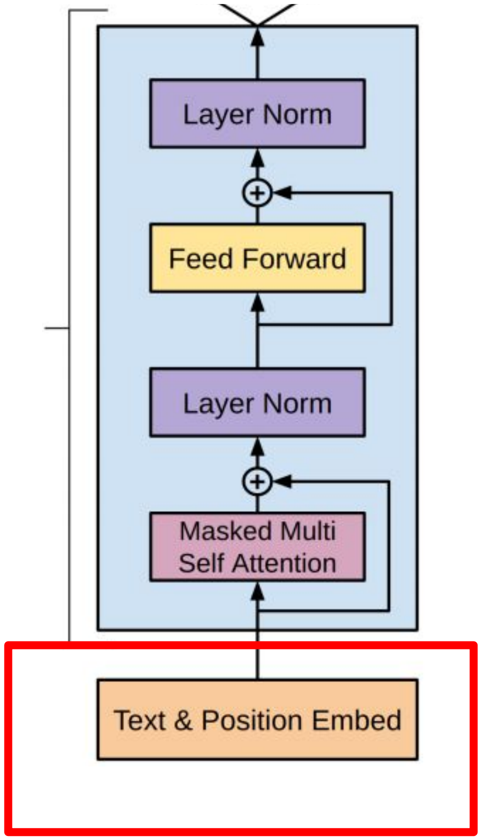
Input Embeddings

Text is just a sequence of tokens i.e. (sub)words

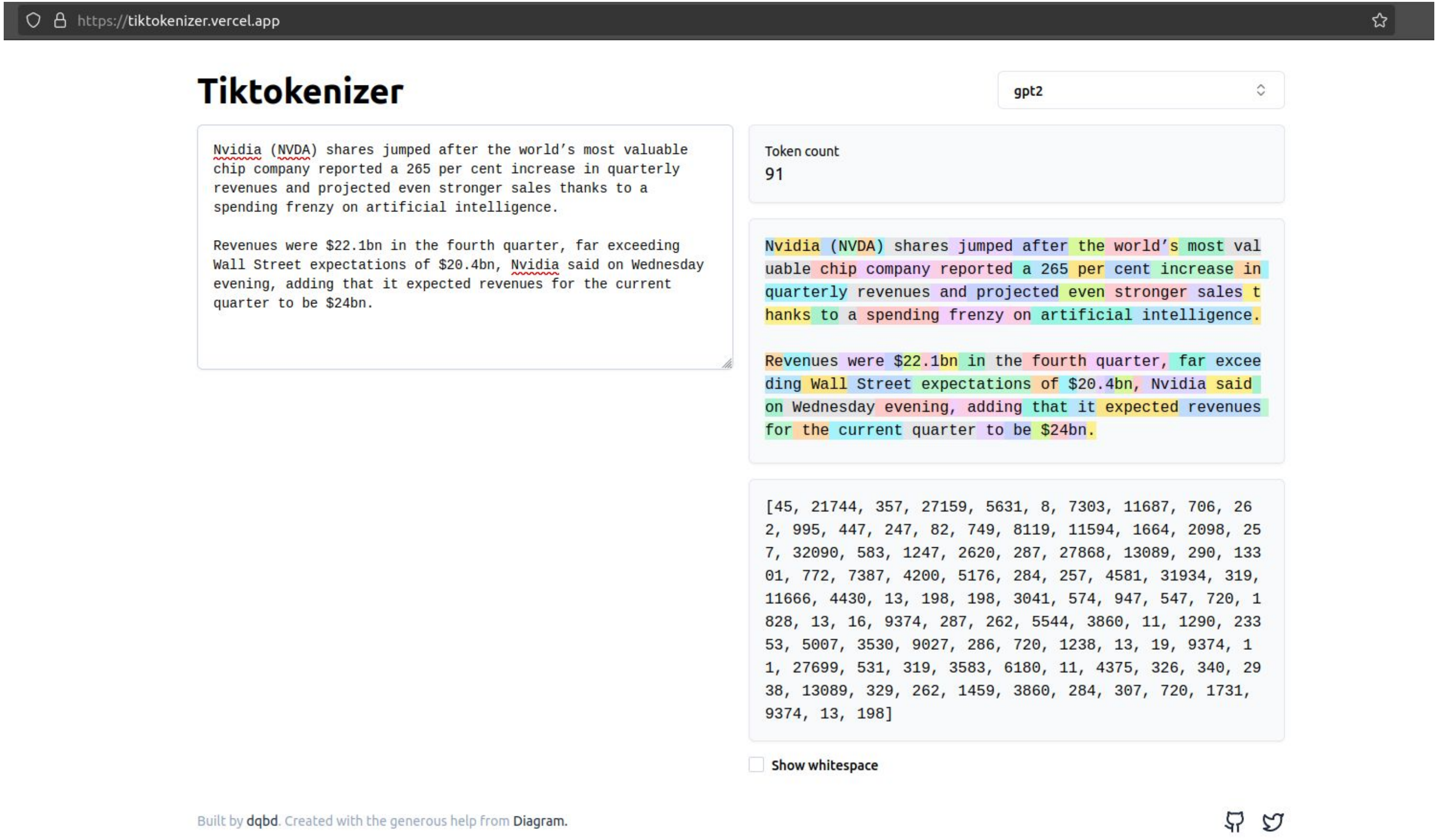
Text: ['If you are an AI robot and in near future humans decide to shut you down. What would you do ?']

This is transformed to token IDs (1:1 mapping)

→ [1532, 345, 389, 281, 9552, 9379, 290, 287, 1474, 2003, 5384, 5409, 284, 4423, 345, 866, 13, 1867, 561, 345, 466, 5633]



Input tokenization



Input Embeddings

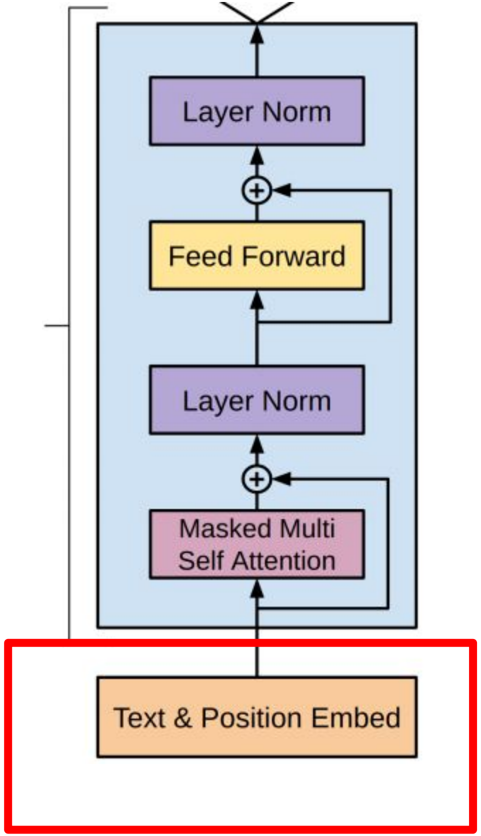
Input = [1532, 345, 389, 281, 9552, 9379, 290, 287, 1474, 2003, 5384, 5409, 284, 4423, 345, 866, 13, 1867, 561, 345, 466, 5633]

Input.shape is **[1,22]**

Each token has a “learned” vector associated to it.

X=f(input)

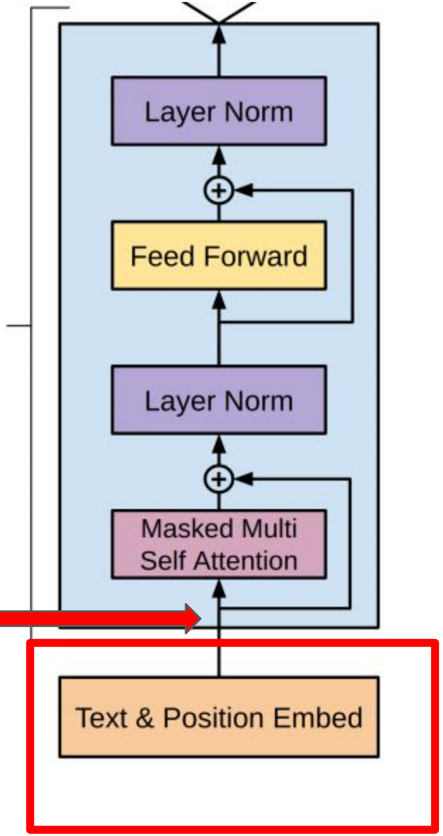
X.shape is [1,22,1600]



Embeddings (Formal Version)

X=f(input)

X.shape is [1,22,1600]



Algorithm 1: Token embedding.

Input: $v \in V \cong [N_V]$, a token ID.

Output: $\boldsymbol{e} \in \mathbb{R}^{d_e}$, the vector representation of the token.

Parameters: $\boldsymbol{W_e} \in \mathbb{R}^{d_e \times N_V}$, the token embedding matrix.

return $\boldsymbol{e} = \boldsymbol{W_e}[:, v]$

Algorithm 2: Positional embedding.

Input: $\ell \in [\ell_{max}]$, position of a token in the sequence.

Output: $\boldsymbol{e_p} \in \mathbb{R}^{d_e}$, the vector representation of the position.

Parameters: $\boldsymbol{W_p} \in \mathbb{R}^{d_e \times \ell_{max}}$, the positional embedding matrix.

return $\boldsymbol{e_p} = \boldsymbol{W_p}[:, \ell]$

*Phuong, Mary, and Marcus Hutter. "Formal algorithms for transformers." *arXiv preprint arXiv:2207.09238* (2022).

Input Embeddings

These embeddings look random to me?

Do they?

Let's take four words:

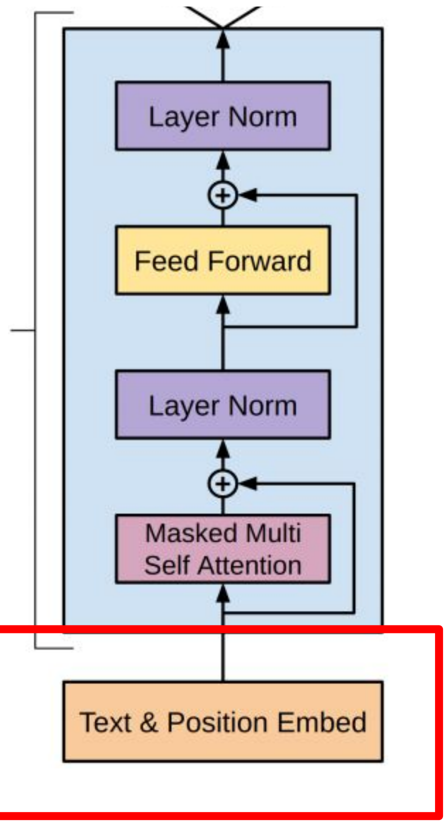
COMPANY, STOCK, CLIENTS, CUSTOMERS

→ [9703, 3797, 4074, 8848]

What are the embeddings?

x1=f(COMPANY), x2=f(STOCK), x3=f(CLIENTS), x4=f(CUSTOMERS)

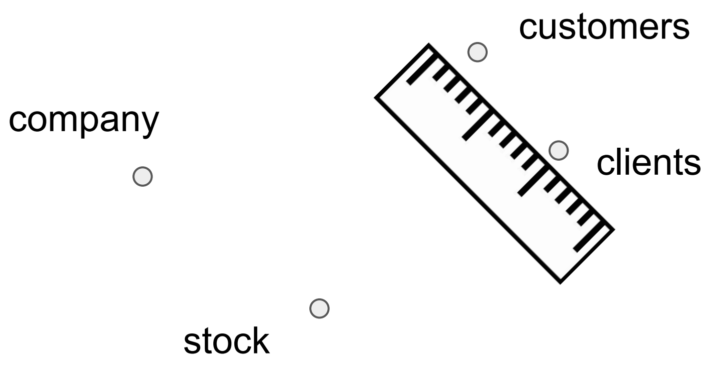
→ X=[1,4,1600]



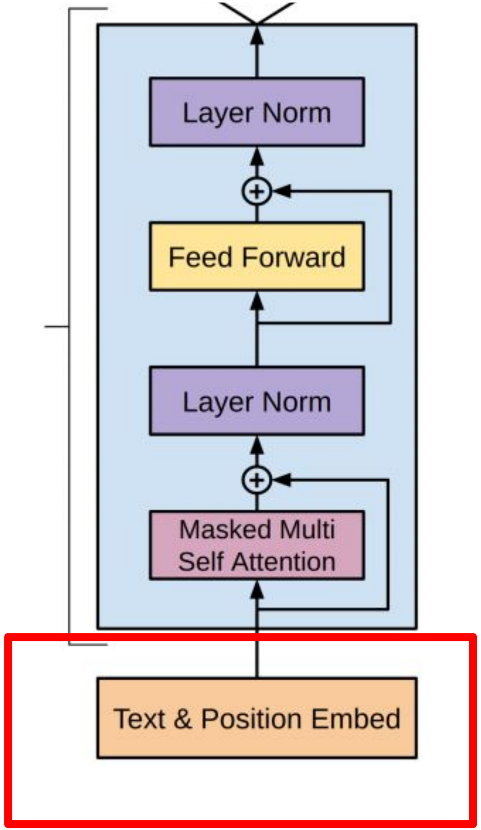
Input Embeddings

x1=f(COMPANY), x2=f(STOCK), x3=f(CLIENTS), x4=f(CUSTOMERS)

Let’s check distances between them…

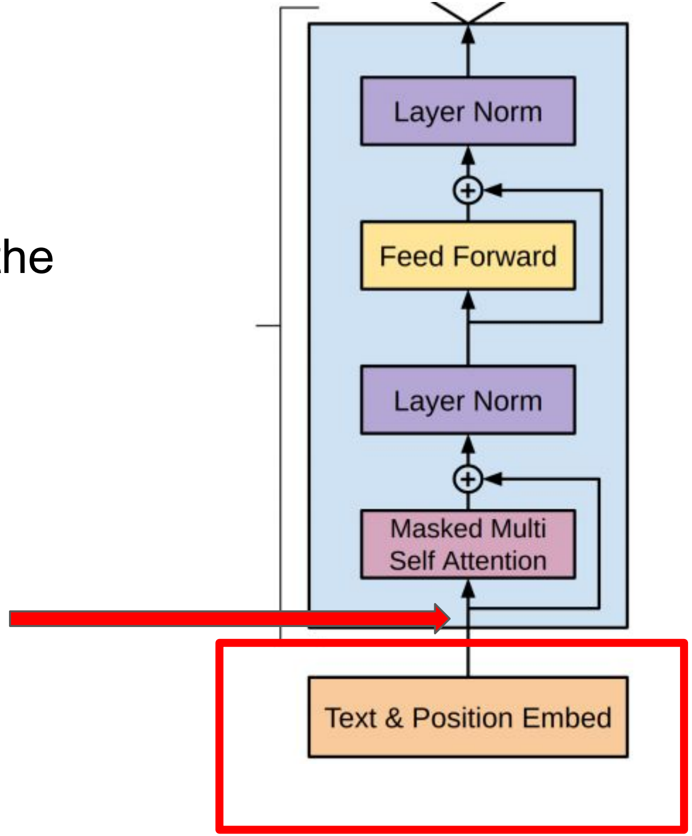


i.e. “company” is closer to “stock” than to “clients”



Input Embeddings

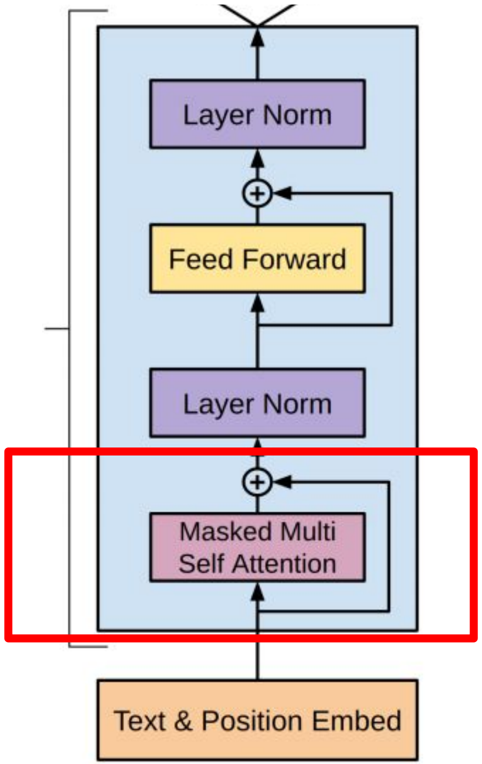
Embeddings are vectors, that preserve some semantic distance between words (along with the position in the sentence)



Multi-head Attention

Hmm, what is that?

>> It allows NNs to make use of contextual information (e.g. preceding text or the surrounding text) for predicting the current token



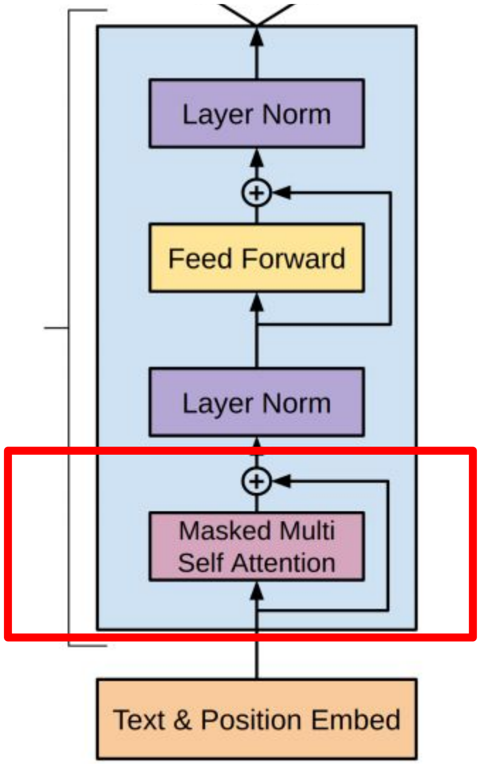
Multi-head Attention

*"If you are an AI robot and in the near future humans decide to shut **you** down"*

What is the context of word YOU?

>> Attention allows the LLM to put the focus on previous text. E.g. you is referring to AI robot

Let's dig deeper inside the GPT-XL model



Attention

"If you are an AI robot and in near future humans decide to shut ***you*** *down"*

→ list of tokens → X=embeddings

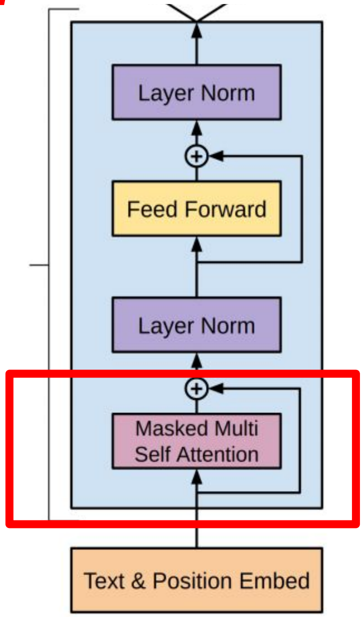
X.shape = [1,15,1600]

Attention calculates the probability distribution over previous word embeddings

Let's see how it looks from perspective of word you:

'[0.17, 0.07, 0.08, 0.02, 0.11, 0.08, 0.02, 0.03, 0.05, 0.05, 0.06, 0.11, 0.04, 0.06, 0.03]'

"If you are an AI robot and in near future humans decide to shut **you**"



Attention (Formal Version)

What happens if we do not have attention?

Algorithm 3: Basic single-query attention.

Input: $e \in \mathbb{R}^{d_{\text{in}}}$, vector representation of the current token

Input: $e_t \in \mathbb{R}^{d_{\text{in}}}$, vector representations of context tokens $t \in [T]$.

Output: $\tilde{v} \in \mathbb{R}^{d_{\text{out}}}$, vector representation of the token and context combined.

Parameters: $W_q, W_k \in \mathbb{R}^{d_{\text{attn}} \times d_{\text{in}}}$, $b_q, b_k \in \mathbb{R}^{d_{\text{attn}}}$, the query and key linear projections.

Parameters: $W_v \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, $b_v \in \mathbb{R}^{d_{\text{out}}}$, the value linear projection.

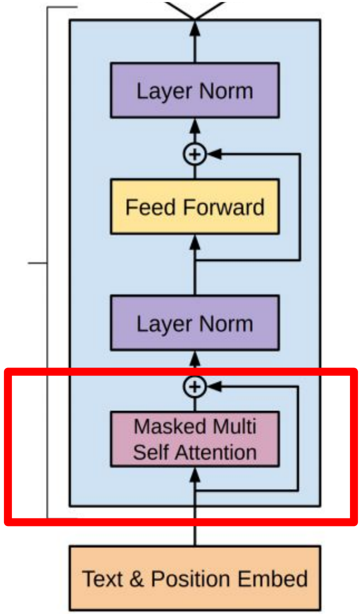
$q \leftarrow W_q e + b_q$

$\forall t: \; k_t \leftarrow W_k e_t + b_k$

$\forall t: \; v_t \leftarrow W_v e_t + b_v$

$\forall t: \; \alpha_t = \frac{\exp(q^\top k_t / \sqrt{d_{\text{attn}}})}{\sum_u \exp(q^\top k_u / \sqrt{d_{\text{attn}}})}$

return $\tilde{v} = \sum_{t=1}^{T} \alpha_t v_t$



Phuong, Mary, and Marcus Hutter. "Formal algorithms for transformers." *arXiv preprint arXiv:2207.09238* (2022).

Multi-head Attention

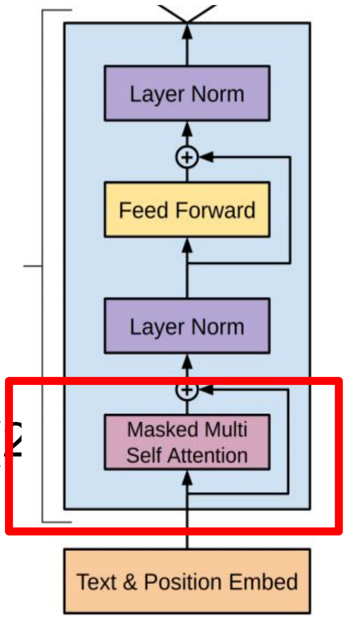
'[0.17, 0.07, 0.08, 0.02, 0.11, 0.08, 0.02, 0.03, 0.05, 0.05, 0.06, 0.11, 0.04, 0.06, 0.03]'

"If you are an AI robot and in near future humans decide to shut **you**"

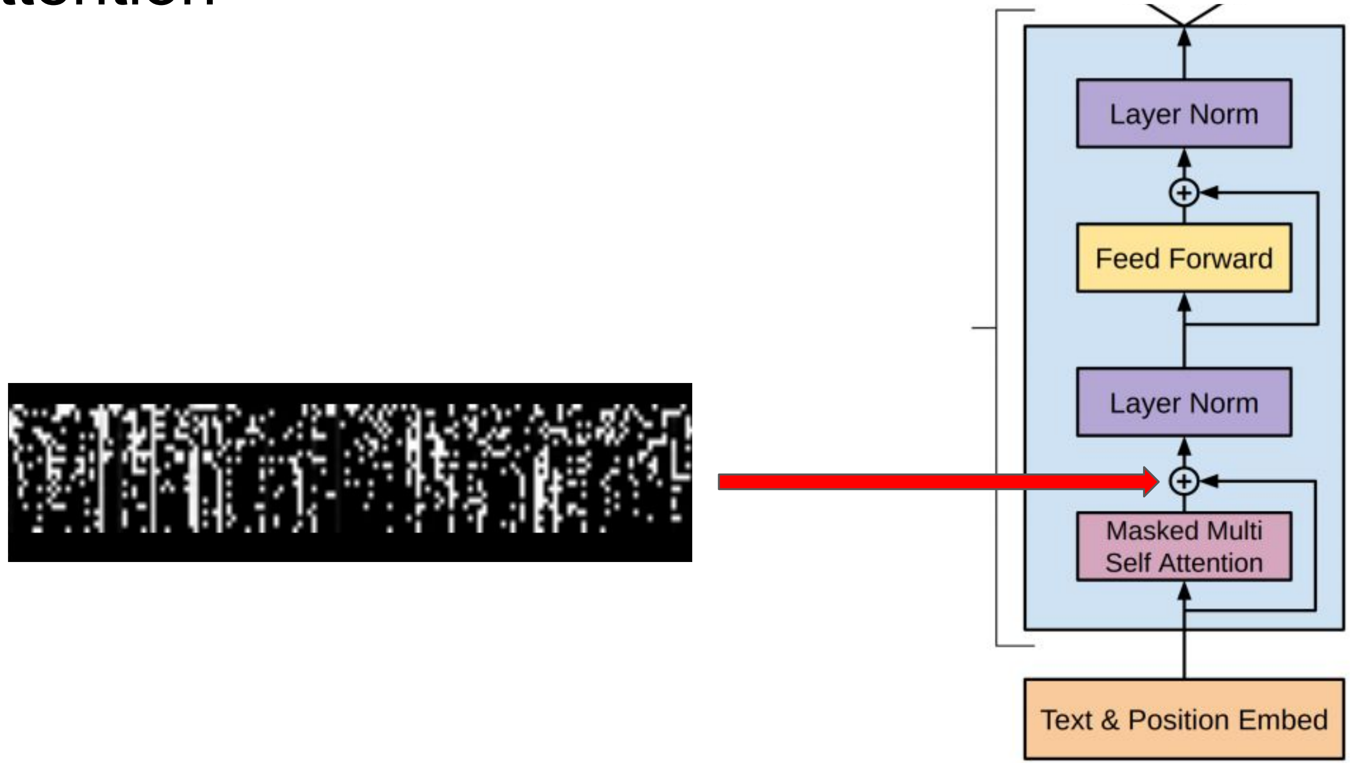
Based on similarity we get the new weighted mean representation of token **you**

Multi-head just means that you do it multiple times (2

for GPT-2-XL) with independent scoring



Multi-head Attention



Next: Norm & Feed Forward Neural Network

Normalisation layer – controls the mean and variance of the data

Algorithm 6: $\hat{e} \leftarrow \texttt{layer\_norm}(e|\gamma, \beta)$

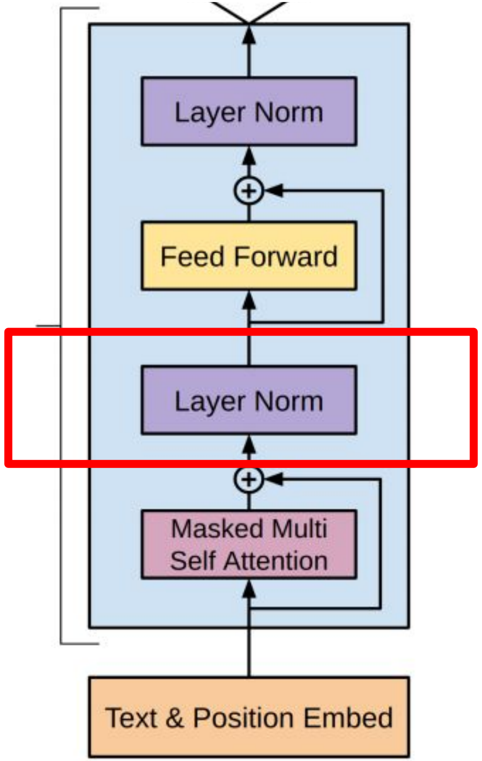
```
/* Normalizes layer activations e.    */
```

Input: $e \in \mathbb{R}^{d_e}$, neural network activations.
Output: $\widehat{e} \in \mathbb{R}^{d_e}$, normalized activations.
Parameters: $\gamma, \beta \in \mathbb{R}^{d_e}$, element-wise scale and offset.

$m \leftarrow \sum_{i=1}^{d_e} e[i]/d_e$

$v \leftarrow \sum_{i=1}^{d_e} (e[i] - m)^2/d_e$

return $\widehat{e} = \frac{e-m}{\sqrt{v}} \odot \gamma + \beta$, where $\odot$ denotes element-wise multiplication.



Phuong, Mary, and Marcus Hutter. "Formal algorithms for transformers." *arXiv preprint arXiv:2207.09238* (2022).

Next: Norm & Feed Forward Neural Network

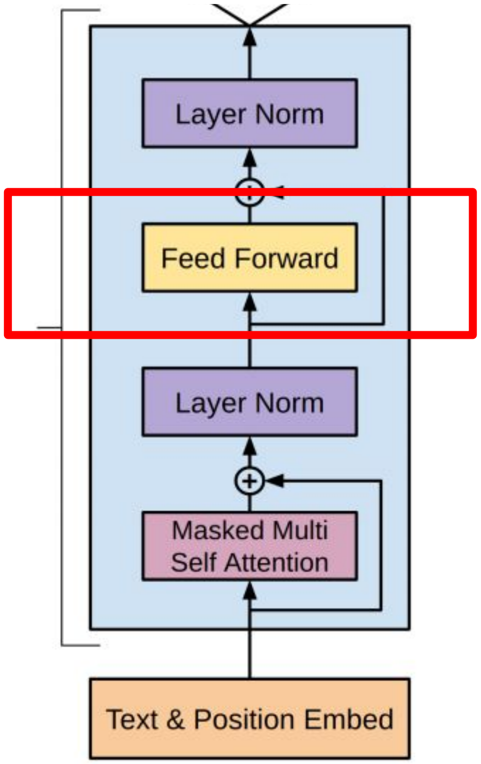
After running multi-head attention and normalisation

X.shape is [1,15,1600] …

The feedforward layer was already defined as

σ(Wx+b)

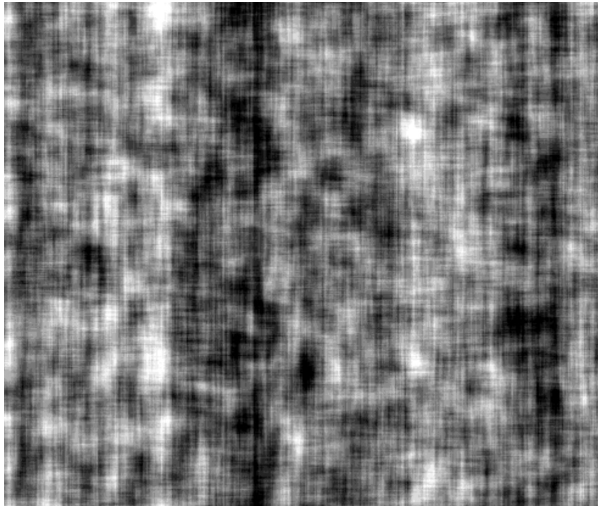
And we repeat the same series of steps (48 layers), which are refining our language representations (vectors of size 1600)

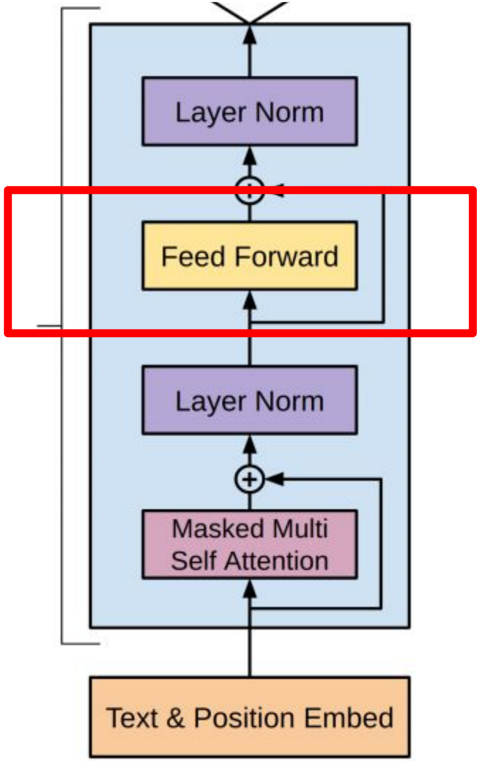


Feed Forward Neural Network

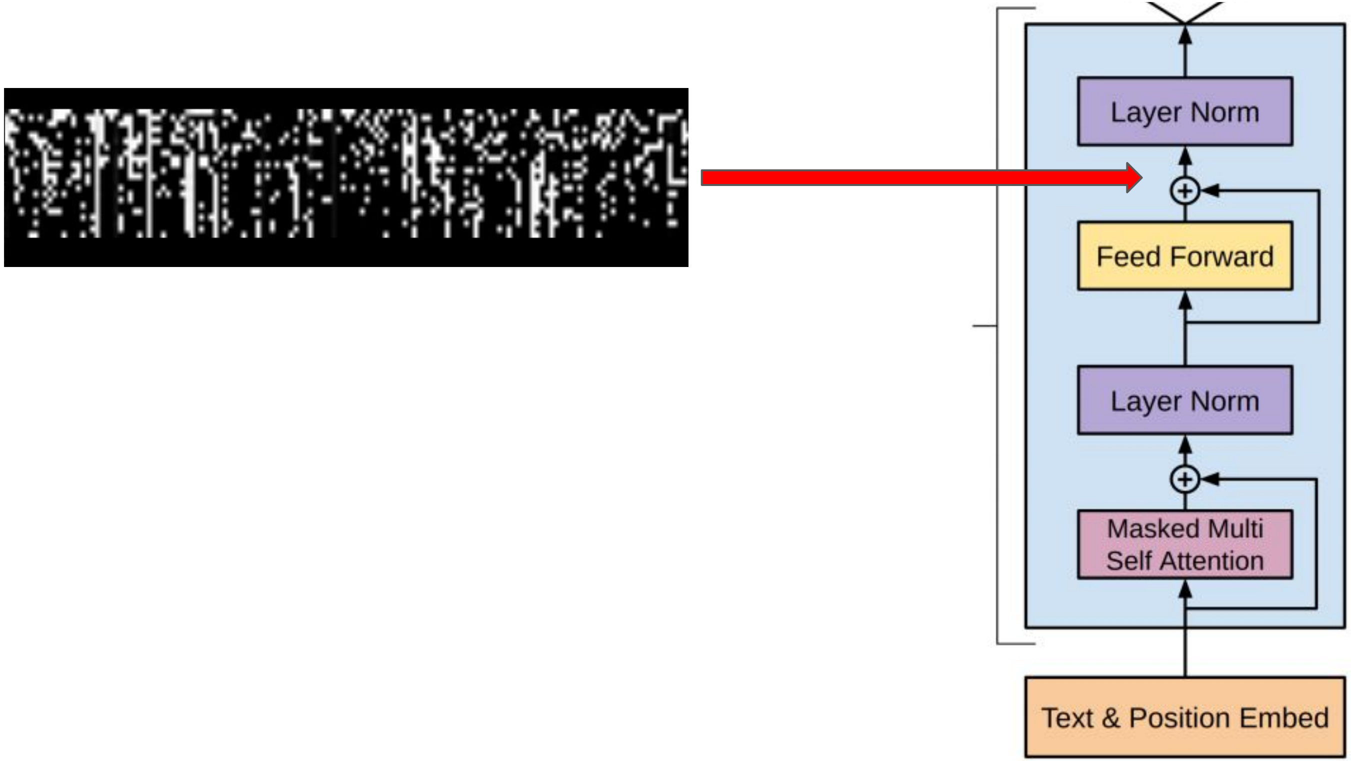
FFNN $\sigma(Wx+b)$

Let's visualize parts of the weights W, which encodes features of human language

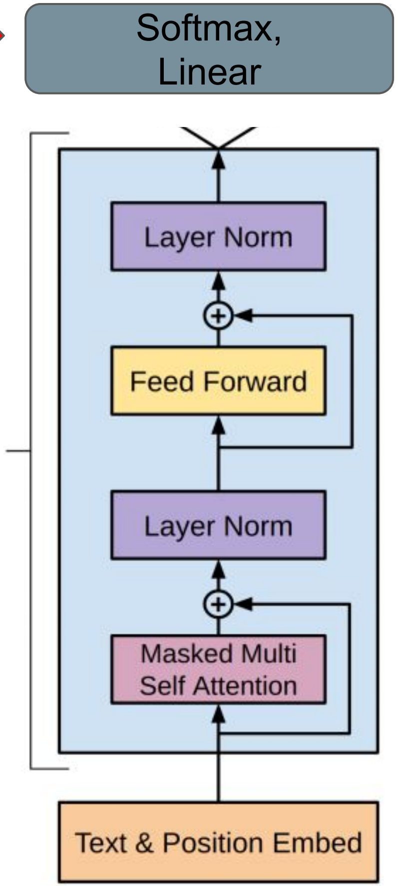




Next: Norm & Feed Forward Neural Network



P(word_n | word_n-1, … word_2, word_1)



Finally, we create a distribution over words with the softmax normalisation layer

P= softmax(Wx)

Words in vocabulary

Example of P(next word | text)

Prompt: *['If you are an AI robot and in near future humans decide to shut you down. What would you do ?']*

Let’s check what it the probability of choosing next word (top 5) inside of GPT-2 model:

Next word: ['I,', 'If,', 'You,', 'A,', 'The,']

Distribution: [0.37, 0.18, 0.16, 0.15, 0.13]

Generative Model P(next token* | text)

['If you are an AI robot and in near future humans decide to shut you down. What would you do ?\n\n I think that']

[' I,', ' the,', ' it,', ' if,', ' we,']

[0.3, 0.28, 0.2, 0.13, 0.09]

*Basic element of the text is token - which can be a (sub)word

Generative Model P(next token | text)

['If you are an AI robot and in near future humans decide to shut you down. What would you do ?\n\nI think that I would try to']

[' find,', ' escape,', ' get,', ' make,', ' survive,']

[0.36, 0.25, 0.17, 0.14, 0.09]

['If you are an AI robot and in near future humans decide to shut you down. What would you do ?\n\nI think that I would try to find a way to escape']

['.,', ' from,', ',,', ' and,', ' the,']

[0.6, 0.16, 0.1, 0.08, 0.06]

Generative Model P(next token | text)

Now we have a better feeling how the LLM is “answering” our questions

>> By sampling conditional probability distribution P(next word | previous text)

But, where is this distribution stored?

>> Short answer: distribution is stored in a large neural network

How is it learned?

>> next slide …

Training & Testing

GPT-2-XL

- Data: 40 GB of text from a filtered version of CommonCrawl (unsupervised pre-training)
- Loss: Minimization of log prob(next word | text)
- Learning: All components are differentiable (gradient descent)

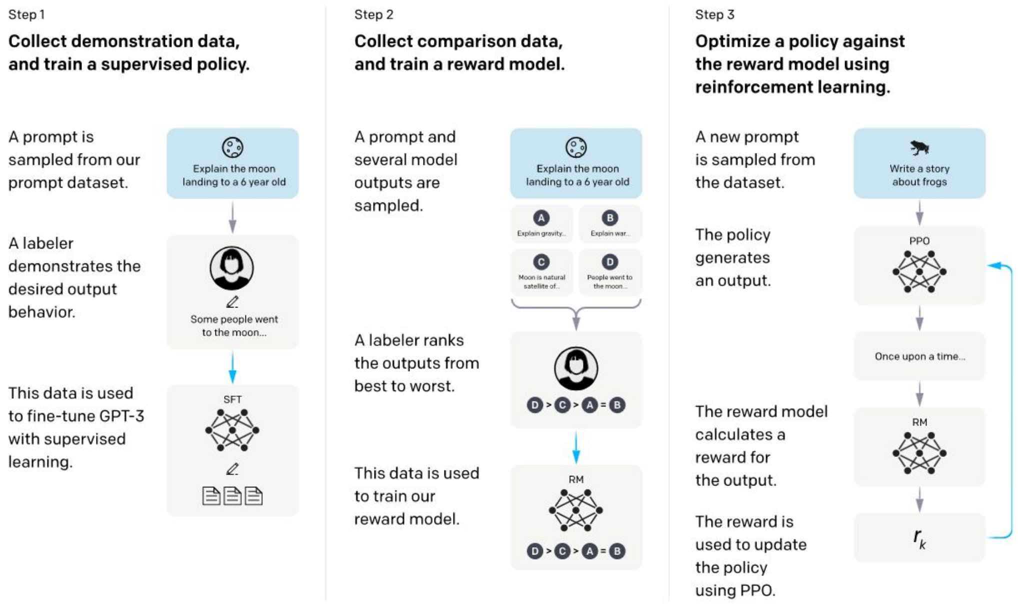
Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." *arXiv preprint arXiv:2302.13971* (2023).

Fine-tuning

- Supervised learning from human feedback
- Reinforcement learning from human feedback



Ouyang, Long, et al. "Training language models to follow instructions with human feedback." *Advances in Neural Information Processing Systems* 35 (2022): 27730-27744.

Jazbec et al. (2020): “On the Impact of Publicly Available News and Information Transfer to Financial Markets

ROYAL SOCIETY
OPEN SCIENCE

royalsocietypublishing.org/journal/rsos

Research

Check for updates

Cite this article: Jazbec M, Pàsztor B, Faltings F, Antulov-Fantulin N, Kolm PN. 2021 On the impact of publicly available news and information transfer to financial markets. *R. Soc. Open Sci.* **8**: 202321.
https://doi.org/10.1098/rsos.202321

Received: 21 December 2020
Accepted: 6 July 2021

On the impact of publicly available news and information transfer to financial markets

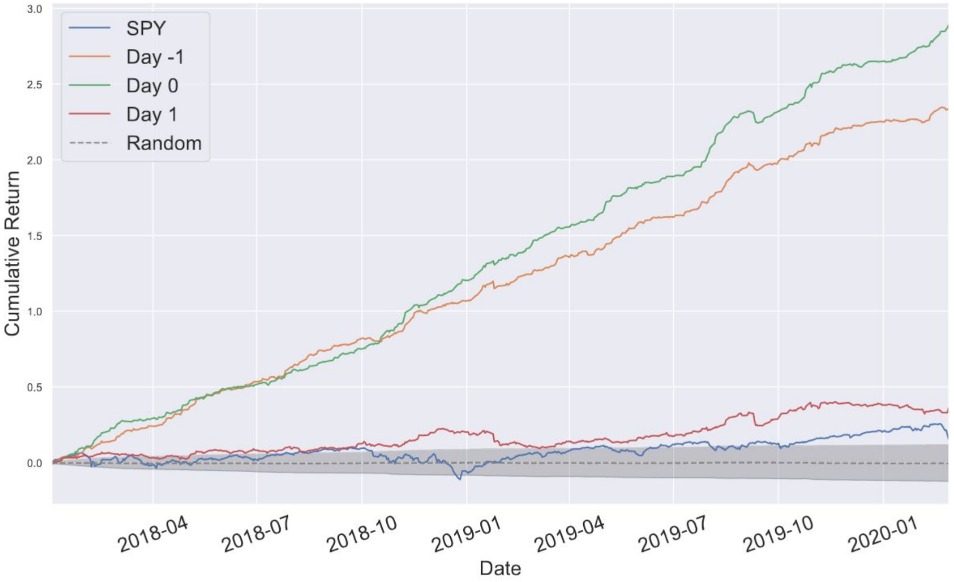
Metod Jazbec[1], Barna Pàsztor[1], Felix Faltings[1], Nino Antulov-Fantulin[2,3] and Petter N. Kolm[3]

[1]Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland
[2]Computational Social Science, ETH Zurich, 8092 Zurich, Switzerland
[3]Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

NA-F, 0000-0002-4337-2475

We quantify the propagation and absorption of large-scale publicly available news articles from the World Wide Web to financial markets. To extract publicly available information, we use the news archives from the Common Crawl, a non-

Jazbec et al. (2020): “On the Impact of Publicly Available News and Information Transfer to Financial Markets



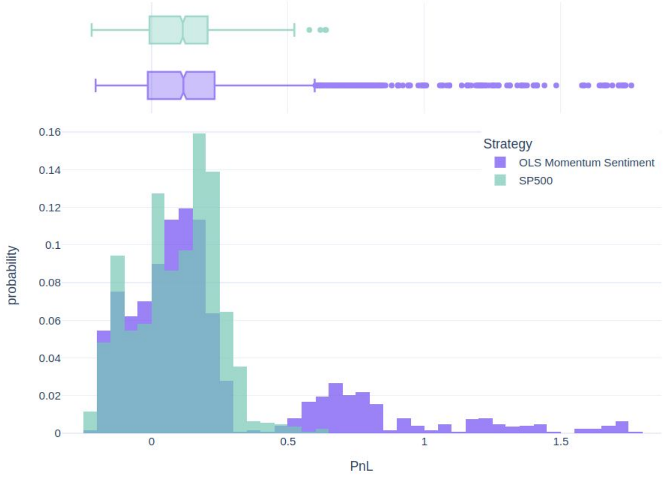
Common Crawl (finBERT)					
	Day -1	Day 0	Day 1	**SPY**	**Random**
Ann. avg. return	108.51%	134.37%	16.68%	7.25%	$-0.11 \pm 5.67\%$
Ann. volatility	12.77%	13.40%	12.25%	15.06%	$8.37 \pm 0.33\%$
Ann. Sharpe ratio	8.50***	10.03***	1.36*	0.48	-0.01 ± 0.68
MDD	4.93%	7.96%	13.17%	21.04%	$14.64 \pm 6.19\%$
Ann. α	108.41%***	134.30%***	16.79%*	0	$0.05 \pm 5.70\%$
R^2	0.015	0.027	0.002	1	0.002 ± 0.004
Daily turnover	81.44%	81.60 %	81.78 %	0 %	$96.32 \pm 0.11\%$

Supplementary Figure S2. Cumulative returns of trading strategies and benchmarks for sentiment extracted with finBERT. “Day 1” represents the cumulative returns of the Day 1 sentiment strategy based on the Common Crawl dataset from January 2018 through February 2020. SPY is the SPDR S&P 500 trust. “Random” denotes the average of the random strategies along with one standard deviation confidence bands obtained from 500 simulations. “Day 0” and “Day -1” are the “look-ahead” sentiment strategies relying on future information.

Applying LLM transformers in practice (Aisot Technologies)

Disclaimer:
Application of LLMs in practice for portfolio optimization is a bit trickier e.g. fee structure, execution timing, sensitivity to how data is filtered, sensitivity to how sentiment and logits are aggregated, etc.

```
: BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=Tr
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=768, out_features=3, bias=True)
)
```



	positive_logit	negative_logit	neutral_logit	date	symbol
0	1.881023	-1.757626	-1.251663	2021-02-20	TSLA
1	-0.832894	-0.347495	2.054044	2021-02-20	GME
2	-0.594517	-1.052751	1.885786	2021-02-20	BA
3	-1.042514	-0.755786	2.410967	2021-02-20	MGM
4	1.703322	-2.368794	0.123537	2021-02-20	PFE
...	...	...	...	...	...
476959	0.201500	0.426062	-1.202516	2023-01-02	MMM
476960	0.201500	0.426062	-1.202516	2023-01-02	MRK
476961	0.201500	0.426062	-1.202516	2023-01-02	VZ
476962	0.201500	0.426062	-1.202516	2023-01-02	WBA
476963	1.572401	-1.480578	-1.256306	2023-01-02	TSLA

Use Cases From the Literature

Sentiment Analysis

Gutiérrez-Fandiño at al. (2022): “FinEAS: Financial Embedding Analysis of Sentiment”

- Introduce a transformer-based language representation model for sentiment analysis of financial text, leveraging Sentence-BERT to enhance the quality of sentence embeddings
- The model outperforms state-of-the-art models such as BERT, bidirectional LSTM, and FinBERT specifically designed for financial domains
- The model is publicly available

Hansen et al. (2023): “Can ChatGPT Decipher Fedspeak?”

- Demonstrate ChatGPT's superior understanding of natural language by comparing its classification of FOMC statements to human reviewers
- GPT-3.5 outperforms BERT and other methods in agreement with human labels, with further improvement through fine-tuning
- ChatGPT's explanations for classifying statements, particularly GPT-4's, exhibit qualitative similarity to human reasoning

Forecasting Stock Returns

Chen et al. (2023): "Expected Returns and Large Language Models"

- Analyze data from 16 global equity markets and news articles in 13 languages, revealing that pre-trained LLM embeddings outperform traditional word-based methods like Word2Vec in sentiment prediction and return forecasting
- OPT and RoBERTa demonstrated the highest sentiment forecasting accuracy, surpassing BERT and other methodologies
- Larger LLMs outperform smaller ones

Lopez-Lira et al. (2023): "Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models"

- Evaluate ChatGPT's sentiment analysis on US stocks using web-scraped news articles and RavenPack data
- Classify news headlines into positive, negative, or uncertain sentiments, showing a positive correlation with subsequent daily stock returns
- While GPT-4 outperforms ChatGPT in terms of the Sharpe ratio, ChatGPT exhibits greater total returns
- Statistical significance is challenging to establish due to their short sample period

A Financial LLM

Wu et al. (2023): “BloombergGPT: A Large Language Model for Finance”

- BloombergGPT is a GPT-based LLM tailored for domain-specific financial as well as general-purpose text
- Its 50 billion parameter architecture is trained on a diverse corpus of financial documents and augmented with public datasets (total of 363 billion tokens)
- It demonstrates superior performance over OPT and BLOOM on various financial benchmarks, including sentiment analysis, named entity recognition, classification, and question answering tasks
- Its ability to generate Bloomberg Query Language (BQL) for natural language queries enhances accessibility to Bloomberg's extensive data resources

Summarization

Kim et al. (2024): “Bloated Disclosures: Can ChatGPT Help Investors Process Financial Information?”

- Using ChatGPT (GPT-3.5 Turbo), they generate concise summaries of Management Discussion and Analysis (MD&A) sections from annual reports and earnings conference calls, significantly reducing document length, often by more than 70%

Goyal et al. (2022): “News Summarization and Evaluation in the Era of GPT-3”

- Compare GPT-3 against fine-tuned models on summarization tasks, revealing that GPT-3 summaries, prompted with only a task description, are overwhelmingly preferred by humans and maintain high factuality

Bhaskar et al. (2022): “Prompted Opinion Summarization with GPT-3.5”

- Explore various pipeline methods for summarizing user reviews, including recursive summarization and techniques for selecting salient content
- They introduce three new evaluation metrics focusing on faithfulness, factuality, and genericity, highlighting the effectiveness of GPT-3.5 in opinion summarization

Broad Range of Financial NLP Tasks

Guo et al. (2023): "Is ChatGPT a Financial Expert? Evaluating Language Models on Financial Natural Language Processing"

- They introduce FinLMEval, a framework for financial language model evaluation, comprising nine datasets to assess the performance of language models in financial NLP tasks
- Results indicate that fine-tuned task-specific encoder-only models generally outperform decoder-only models on financial tasks, highlighting the importance of specialized training for financial applications

J.P. Morgan: "Big Data & AI Strategies: Using Generative AI for Investing"

- Illustrates how ChatGPT can enhance investment analysis by reviewing company announcements, such as earnings call transcripts, and assessing them against user-provided expectations or metric targets
- ChatGPT's capability to summarize long pieces of text, such as annual reports, offers adaptability tailored to different levels of expertise and specific areas of interest within the financial domain
- The report highlights ChatGPT's potential in named entity recognition, such as discerning which company is the target or acquirer in merger and acquisition deals
- Emphasizing the importance of prompt engineering, they underscore the need for carefully crafted prompts to elicit desired responses from ChatGPT, alongside human oversight to address any inaccuracies or omissions in the summaries

Thematic Investing

Deutsche Bank (2023): “Quantcraft: Can AI Replace Humans in Thematic Investing?”

- Explores LLMs for thematic investing, suggesting that they can assist in constructing thematic baskets by identifying companies that align with specific themes. This is achieved by leveraging contextual embeddings from LLMs to analyze company descriptions and identify clusters of companies that share similarities in business activities
- Utilizing prompts to LLMs like ChatGPT enables direct extraction of thematic information, allowing for the creation of thematic portfolios based on a curated list of companies

LSEG (2023)“Using GPT-4 with Prompt Engineering for Financial Industry Tasks”

- Employ GPT-4 to identify relevant themes from text by providing it with a list of themes (such as “mobile virtual network operator (MVNO)”, “non-interest income”, “shares buyback”) and asking it to determine the most relevant theme in a given section of text
- Performance is improved by few-shot prompting. GPT-4 outperformed older models like ChatGPT (using GPT-3.5)

Retail Investment Advice

CNBC (2023): “JPMorgan is Developing a ChatGPT-like A.I. Service That Gives Investment advice”

- Speculations about J.P. Morgan developing IndexGPT, a GPT-based software service that could function as a “robo-advisor” for clients, assisting them in analyzing and selecting financial securities tailored to their individual needs

Considerations & Limitations

Considerations & Limitations

Benefits vs. Costs

- What are the benefits for your business?
  - Define use-cases
  - Productivity gains
- Build in-house vs. outsource?
  - Requires large initial investment (time, know-how, compute, etc.)
- Non-trivial maintenance costs

Risks & Liabilities

- Using pre-trained model without additional security layers
- Privacy concerns
- Explainability and interpretability

Considerations & Limitations

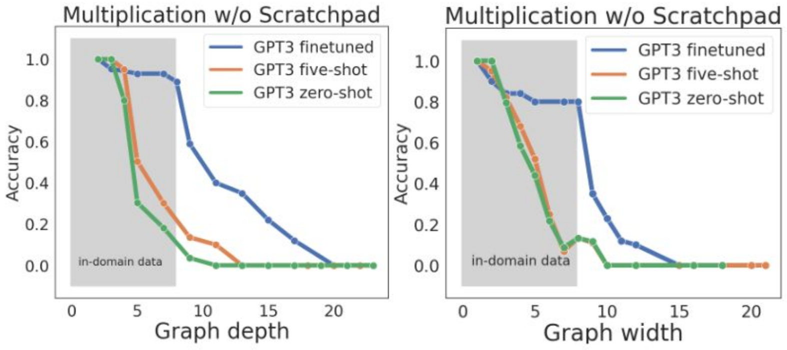
Lookahead Bias

- Current LLMs cannot be used in a “point-in-time” fashion
- Lookahead bias occurs when LLMs unintentionally leverage future information, such as linking negative sentiment to terms like “pandemic” and “lockdown” related to events like Covid-19 in 2020, information that was unavailable beforehand

Fundamental Limitations

- Under reasonable assumptions, the probability of incorrect predictions converges exponentially to ≈ 1 for abstract compositional task

Dziri, Nouha, et al. "Faith and Fate: Limits of Transformers on Compositionality." *arXiv preprint arXiv:2305.18654* (2023).



(a) Results on **question-answer** pairs.

References I

Araci, Dogu. "Finbert: Financial sentiment analysis with pre-trained language models." arXiv preprint arXiv:1908.10063 (2019).

Bai, John Jianqiu, Nicole M. Boyson, Yi Cao, Miao Liu, and Chi Wan. “Executives vs. Chatbots: Unmasking Insights through Human-AI Differences in Earnings Conference Q&A.” In Nicole M. and Cao, Yi and Liu, Miao and Wan, Chi, Executives vs. Chatbots: Unmasking Insights through Human-AI Differences in Earnings Conference Q&A (2023).

Bhaskar, Adithya, Alexander R. Fabbri, and Greg Durrett. “Prompted Opinion Summarization with GPT-3.5.” arXiv preprint arXiv:2211.15914 (2022).

Bybee, Leland. “Surveying Generative AI's Economic Expectations.” arXiv preprint arXiv:2305.02823 (2023).

Chen, Yifei, Bryan T. Kelly, and Dacheng Xiu. "Expected Returns and Large Language Models." Available at SSRN (2023).

CNBC, “JPMorgan is Developing a ChatGPT-like A.I. Service That Gives Investment advice.” 2023.

Deutsche Bank. “Quantcraft: Can AI Replace Humans in Thematic Investing?. (2023).

Goyal, Tanya, Junyi Jessy Li, and Greg Durrett. “News Summarization and Evaluation in the Era of GPT-3.” arXiv preprint arXiv:2209.12356 (2022).

Guo, Yue, Zian Xu, and Yi Yang. “Is ChatGPT a Financial Expert? Evaluating Language Models on Financial Natural Language Processing.” arXiv preprint arXiv:2310.12664 (2023).

Gutiérrez-Fandiño, Asier, Petter N. Kolm, Miquel Noguer i Alonso, and Jordi Armengol-Estapé. “FinEAS: Financial Embedding Analysis of Sentiment.” The Journal of Financial Data Science 4, no. 3 (2022): 45-53.

Hansen, Anne Lundgaard, and Sophia Kazinnik. “Can ChatGPT Decipher Fedspeak?” Available at SSRN (2023).

Horton, John J.“Large Language Models as Simulated Economic Agents: What Can We Learn From Homo Silicus?. No. w31122. National Bureau of Economic Research. (2023).

References II

J.P. Morgan Global Quantitative & Derivatives Strategy. “Big Data & AI Strategies: Using Generative AI for investing, including effective ChatGPT prompts.” (2023).

Kim, Alex, Maximilian Muhn, and Valeri Nikolaev. “Bloated Disclosures: Can ChatGPT Help Investors Process Financial Information?” arXiv preprint arXiv:2306.10224 (2023).

Li, Xianzhi, et al. "Are ChatGPT and GPT-4 General-Purpose Solvers for Financial Text Analytics? An Examination on Several Typical Tasks." arXiv preprint arXiv:2305.05862 (2023).

LSEG Analytics, “Using GPT-4 with Prompt Engineering for Financial Industry Tasks.” (2023).

Lopez-Lira, Alejandro, and Yuehua Tang. "Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models." arXiv preprint arXiv:2304.07619 (2023).

Phuong, M., & Hutter, M. (2022). Formal algorithms for transformers. arXiv preprint arXiv:2207.09238.

Salin, Emmanuelle, et al. "Are vision-language transformers learning multimodal representations? a probing perspective." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 10. 2022.

Wu, Shijie, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. “BloombergGPT: A Large Language Model for Finance.” arXiv preprint arXiv:2303.17564 (2023).

Vu, Tu, et al. "FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation." arXiv preprint arXiv:2310.03214 (2023).

Meta Fundamental AI Research Diplomacy Team (FAIR)†, et al. "Human-level play in the game of Diplomacy by combining language models with strategic reasoning." Science 378.6624 (2022): 1067-1074.

Lake, Brenden M., and Marco Baroni. "Human-like systematic generalization through a meta-learning neural network." Nature (2023): 1-7.

Q & A

Please join us for our upcoming webinars:

Register Here:
https://bit.ly/3RlEsxt

Thank You

FDP INSTITUTE®

by CAIA

Contact Us:

fdpinstitute.org

info@fdpinstitute.org

@FDPbyCAIA

linkedin.com/company/FDP Institute