Large language models in finance

By Dr Svetlana Borovkova
(VU Amsterdam and Probability & Partners)

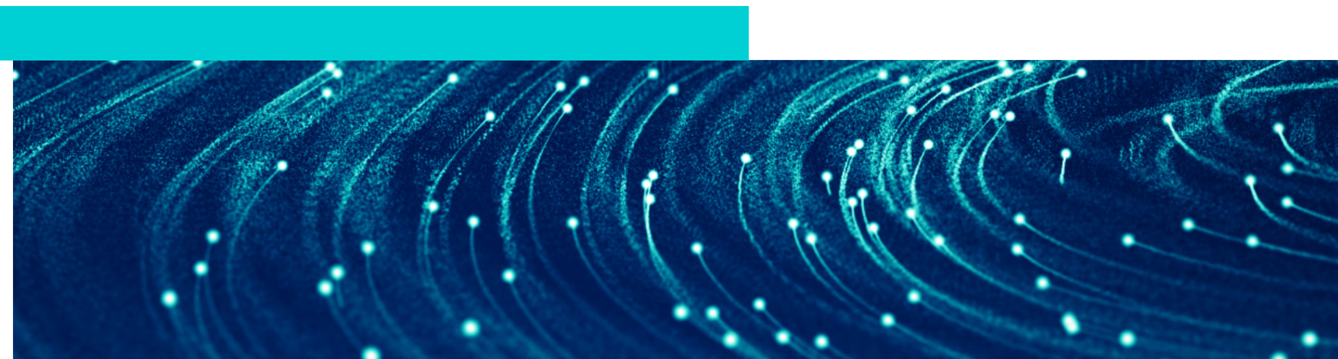
PROBABILITY & PARTNERS

LSEG DATA & ANALYTICS

Contents

Introduction **3**
What is a large language model? **4**
Some history: symbolic and ML-based NLP 5
Transformers 5
BERT vs GPT 6
Generative LLMs applications in financial sector **7**
Current and impending applications 7
Sentiment analysis and further applications 9
Adapting a generative LLM to your task **10**
Prompt engineering 11
Parameter-efficient fine tuning (PEFT): LoRA and QLoRA 12
Logit processing 13
LSEG datasets for LLM applications **14**
Pitfalls of using LLMs and mitigation strategies **16**
Timeliness 16
Hallucinations 17
Problem of back-testing 17
Model risk, explainability and fairness 18
Other issues **20**
Sustainability and power consumption 20
Commercial solutions for GPU, cloud computing and re-training LLMs: in-house vs outsourcing debate 20
Conclusion **21**

In this paper, we give a bird’s-eye view of large language models (LLMs) and outline the most significant issues related to their applications in financial services. We will discuss potential use cases, LLMs’ limitations and the challenges associated with their applications. The objective is to provide the reader with understanding of various aspects of LLMs, placing them in the context of financial institutions. Additionally, we will discuss ways of implementing LLMs in finance-related areas, outline potential dangers and pitfalls and explore emerging strategies for overcoming these challenges.



Introduction

The emergence of generative AI and LLMs such as ChatGPT is the most significant technological development of the past year. The speed with which ChatGPT has been adopted by the public is unprecedented: within just five days of its public release on 30 November 2022, ChatGPT amassed over one million users – something that took Spotify five months and Netflix five years. Over the past year, ChatGPT has remained the talk of the town in financial institutions' hallways and boardrooms.

Following OpenAI's ChatGPT release, other companies accelerated the development of their own comparable LLMs. The most significant one is Meta's LLaMA, which is a family of LLMs released in February 2023. LLaMA, being open source rather than proprietary like ChatGPT, currently serves as the foundation model for fine-tuned variants of LLMs.

In the weeks and months after the ChatGPT release there was a prevailing sense of euphoria mixed with awe and fear at its human-like intelligence. Nearly a year later we are still expanding our understanding of the capabilities and limitations of LLMs. The conversation about LLMs' application in various industries, including finance, is becoming more nuanced and informed.

While many emerging IT and consulting firms are developing business cases with LLMs, more established players – particularly in financial services – are entering the game. Financial quantitative specialists (quants) are identifying areas in which LLMs can be effectively applied. Overall, especially in finance, there is less excitement about LLMs than there was a year ago, the sentiment replaced by measured and informed perspectives. Nonetheless, in day-to-day conversations there still exists a considerable amount of sometimes-uninformed enthusiasm often coupled with an assumption that LLMs will replace many human functions and insights. This is frequently accompanied by a lack of deep understanding of the potential dangers and pitfalls of using LLMs, as well as general unawareness of constructive ways forward.

In this paper, we will give an overview of LLMs and outline the most significant issues related to their applications in financial services. We will discuss potential use cases, LLMs' limitations and the challenges associated applying them. It will become evident that large, reliable and use case-relevant historical datasets and domain-specific knowledge are crucial to the application of these ground-breaking models in finance. Using several examples of LSEG datasets, we will illustrate how such data can be used to achieve improvements and design effective variants of LLMs tailored for specific financial applications.

What is a large language model (LLM)?

A large language model is a machine learning model capable of ‘understanding’ and generating human-like text across a wide variety of contexts. These models are fundamentally artificial neural networks consisting of a gigantic number of nodes, layers and connections, resulting in a huge number of parameters (weights). Due to their vast scale, they are trained on very large textual datasets (such as the entire corpora of Wikipedia). Their primary goal is to convert text into ‘meaning’, also known as **embedding**. The embedding, situated in the latent space of those networks, represents a very rich set of different language aspects – or rather, of meanings. This includes such elements as style, substance, structure, semantics, syntax, formatting and various others.

For LLMs, the fundamental ‘units’ they operate on (i.e., their ‘alphabet’) are the so-called ‘**tokens**’, rather than ‘letters’ or ‘words’. A token can be a single character or a whole word and some often-used words are themselves single tokens. Language is converted into a sequence of tokens by a process called **tokenisation**, which is the initial step in any LLM. Subsequently, these sequences of tokens are combined with their positions into vectors, which are then fed into the neural network. The neural network learns to compress or embed them into its latent space, representing the meaning of the original text. Different facets of this meaning can be converted into desired characteristics of the original text based on the specific task the LLM was trained for. For instance, an LLM can generate a sentiment score for the input text or identify the main subject of the text, a task known as named-entity recognition.

Generative LLMs, such as ChatGPT (also called **generative AI**), belong to the category of sequential or **autoregressive completion models**. When given a prompt, these models can generate human-like text by auto-completing previous text (for example, a user’s input or prompt), with the most likely continuation. This process involves initial pre-training of the model on an extensive textual dataset, creating a rich, versatile and densely populated embedding space.

The model incrementally and probabilistically samples from this embedding space to find the ‘meaning’ most likely to follow the meaning of the given prompt or the preceding output. Neural networks, being excellent interpolators, can generate meaning not seen in their training corpus, which makes them powerful tools for generating diverse and context-relevant text.

LLMs are primarily unsupervised machine learning algorithms: they require no tagged data to pre-train them. These models are trained on a broad corpus of textual data and the resulting trained LLMs, like ChatGPT, are able to handle a wide variety of contexts and tasks but lack any deep subject-specific knowledge. Herein lies both their main strength and their greatest weakness. However, LLMs can be adapted to specific tasks by re-training or fine-tuning them based on domain-specific and tagged data. This allows LLMs to acquire specialised knowledge – a topic we will explore in more detail later on.

Some history: symbolic and ML-based NLP

Interpretation of human language by computers traces back to the rule-based techniques of natural language processing (NLP) rooted in computational linguistics. These techniques involve lexical, semantic and syntax analysis (parsing) and combine our understanding of the rules and structure of text with specific dictionaries.

Statistical NLP emerged in the late 1990s and by the early 2000s, machine learning-based NLP started to develop, eventually replacing the statistical approach around 2015. Machine learning-based NLP relies on artificial neural networks to understand the structure and meaning of text. This is achieved by training these networks on large quantities of textual data, which became increasingly available with the growth of the Internet.

Deep learning involves neural networks with many hidden layers and various network configurations (e.g., recurrent, convolutional or long-short term memory networks) which have been trained to interpret text and perform specific tasks. Over the years, there have been steady improvements in typical NLP applications such as machine translation, name entity recognition and sentiment analysis. However, natural language generation, the task of creating human-like text, remained an elusive and difficult task for a long time, even with the most advanced neural networks – until recently.

Transformers

The year 2017 marked a truly revolutionary development in machine learning with the introduction of the Transformer architecture. This development, introduced by Google Brain researchers Vaswani et al. in their seminal paper 'Attention is all you need', is the foundation upon which are built large-scale models capable of understanding and generating human-like text across a diverse range of tasks.

Unlike traditional neural networks that process words (or, rather, tokens) in an input sentence sequentially, a Transformer processes all words in an input sequence **simultaneously** in parallel. This parallel processing is done by the so-called **attention** mechanism, which assigns soft (changing) attention weights to all words in a sequence simultaneously, based on their 'importance' or relevance to a query (reference) word. This intrinsic parallelism allows for much faster training compared to sequential networks, enabling the training of significantly larger models (with huge numbers of parameters) on much larger datasets. Transformers played the key role in the development of pre-trained LLMs: all LLMs such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) are powered by this architecture.

Transformers come in two types: encoders and decoders. Both share the fundamental principle of compressing text into the latent space of its 'meaning'. However, encoders analyse sequences of input text as a whole (so they know the beginning, middle and end of a sequence simultaneously), while decoders reveal tokens in a sequence from left to right. At each point in time, a decoder only knows the past tokens and uses this information to predict the next token in the sequence, with all subsequent tokens 'masked'. Consequently, decoders are often called **autoregressive text generators** – In simpler terms, encoders are bidirectional information processors, while decoders are unidirectional.

The original transformers combined both an encoder and a decoder. However, it is possible to build a language model using only one of them. An example of an encoder-only language model is BERT, whereas the GPT model family is a decoder-only language model.

BERT vs GPT

BERT, developed in 2018 by Google researchers, is a family of language models with an encoder transformer architecture. It comes in two sizes: BASE, with 110 million parameters, and LARGE, with 340 million parameters. By modern standards, these are both relatively small language models.

By now, BERT has become an established model widely used by researchers and businesses globally. Pre-trained BERT, available as open source, can be fine-tuned on smaller, labelled and domain-specific datasets to perform specific tasks. It is widely used in financial services, for example, for sentiment analysis of financial news, analysing analyst call transcripts, M&A deal sheets and more. An example of a successful application is LSEG StarMine M&A Target Model. This model uses fundamentals and pre-trained BERT, fine-tuned on LSEG News Archive and historical M&A deals, to score companies' likelihood of being acquired in an M&A deal in the next 12 months.

In tasks which are specific and exact – and when relevant labelled training data is available – BERT and other encoder models often exhibit excellent performance. The reason for this is twofold: first, BERT's bidirectional encoder architecture allows it to construct versatile text embeddings since it has knowledge of the entire sequence and hence of the context. Second, the pre-trained BERT can easily be customised and fine-tuned for various precise tasks. For classification, for instance, we only need to add a simple one-hidden-layer neural network (classifier) on top of a pre-trained BERT and train it on a modest-sized labelled dataset to achieve excellent performance.

However, an encoder architecture means that BERT is unable to generate new text as it lacks the decoder part. The latest generation of LLMs: GPT (generative pre-trained transformers) – overcomes this limitation.

The most famous GPT is OpenAI's ChatGPT, released in its 3.5 version in November 2022 as an open chatbot. OpenAI has been pioneering GPT models since 2018. In March 2023, the latest model GPT4 was released as a paid version, ChatGPT Plus. GPT models are much larger than encoder models: it is purported that GPT4 has 1.76 trillion parameters but OpenAI has never revealed the exact size of the model.

Generative LLMs such as ChatGPT are trained on huge, often undisclosed data corpora, first in an unsupervised way and later in a semi-supervised way using what is called **reinforcement learning with human feedback**. This training approach makes them versatile and proficient across a great variety of tasks, such as summarising documents, writing poetry or having a human-like conversation on practically any topic. However, this also makes them too general and hence they lack any significant domain-specific expertise. Generative LLMs also come with other drawbacks, such as lack of timeliness, tendency for incorrect answers and others, which will be addressed further. Despite these challenges, these models are technological marvels, beyond what anyone thought possible just a year ago.

ChatGPT can be used through a chatbot interface or via the paid API but it is a proprietary model that cannot be modified. The key for successful applications of generative LLMs is the ability to modify them for specific tasks, similar to what can be done with BERT. The most useful development in generative LLMs for finance applications is the emergence of **open-source GPT models** such as LLaMA by Meta AI. LLaMA is a family of models with different sizes (7B, 13B, 32B and 65B parameters). LLaMA 2, the next generation of LLaMA, is also available in various sizes (7B, 13B and 70B parameters) and represents a significant improvement over the original. Other open-source models are offered several companies including EleutherAI and Cerebras.

These models, referred to as so-called **foundation models**, are pre-trained on huge textual datasets and available for download and free use by researchers and businesses. The main advantage of these open- source foundation models lies in the ability to modify them for different applications (more about this later). Various modifications of LLaMA (e.g., Alpaca, Vicuna) are rapidly emerging.

Applications for generative LLMs in the financial sector

Current and upcoming applications

Generative LLMs are rapidly finding their way into business and finance. While some applications are already in use, others are actively being explored and developed by companies and service providers. Below we will outline both existing and emerging applications, without speculating about possible applications that are currently beyond reach.

Pre-trained LLMs are good at summarisation and information retrieval but show less capability in intelligence generation. So most present and imminent applications of LLMs in the financial sector focus on areas of low materiality but high efficiency gains. This includes **summarisation of** lengthy and verbose **documents** like company reports or analyst calls. Recently, financial firms have been thinking of using LLMs in **ESG applications**, i.e., for summarising and interpreting companies' sustainability reports to extract E, S and G information for internal ESG ratings.

Another application is summarisation, clean-up, and **generation of financial instrument documentation**, such as term sheets for derivatives and mortgage/loan offerings, as well as checking for contracts' compliance with existing regulation. However, for LLMs to be effective in generating such documents, they need to be fine-tuned on similar documents. Fortunately, this is feasible as large public datasets of such documents are available. For example, ISDA Create platform contains various derivatives master agreements, loan agreements, initial and variation margin term sheets and other relevant documents.

Another promising class of applications is in the area of **model governance**. Fine-tuned LLMs can help **create model documentation, model validation** templates and **reports** and assist with various other tasks related to model inventory and model risk. Once again, fine-tuning LLMs on related, relevant data is crucial for these applications.

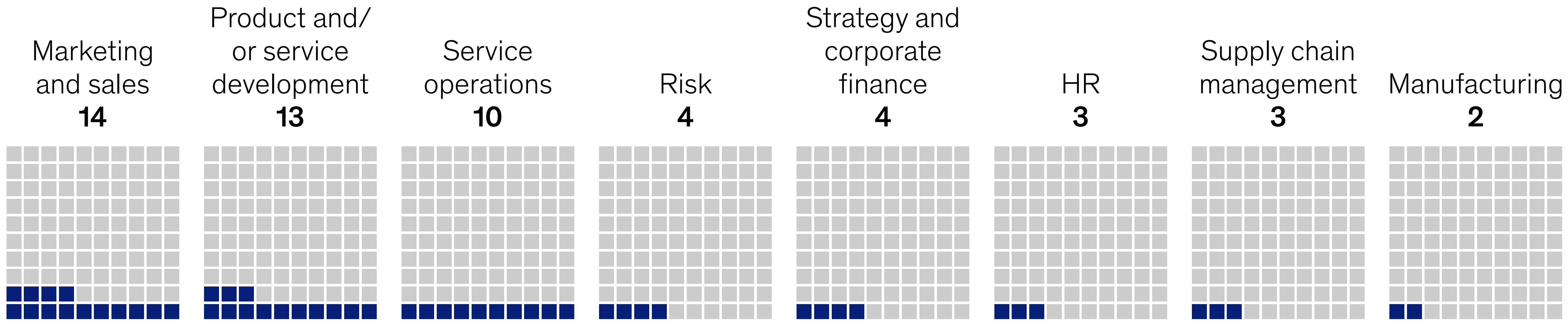
An area in which LLMs are making a significant difference is in coding. While it is not yet feasible to let an LLM generate a trustworthy computer code on its own, it can be of great assistance in **debugging, improving and commenting on or critiquing computer code.** Quantitative modellers are already using generative LLMs to **translate between coding languages** or to identify and link existing subroutines to complete code for a specific task.

Applications that are relevant for – but not limited to – finance fall into areas such as marketing, customer experience and customer care. A recent release by Salesforce, Einstein GPT (which is powered by same the OpenAI technology that powers ChatGPT), is precisely such a LLM, specifically fine-tuned to customer relation management applications.

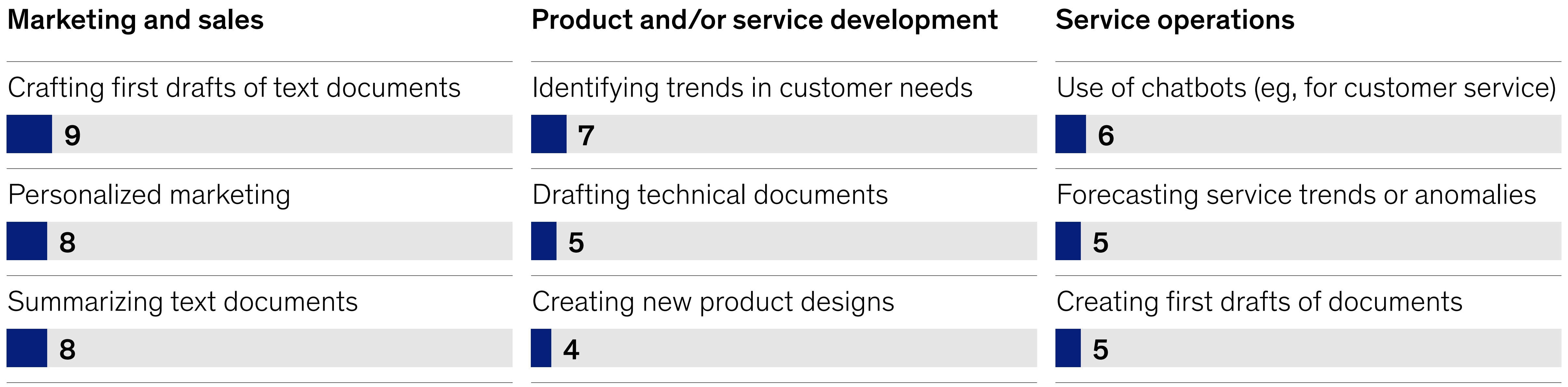
Currently, generative LLMs are often perceived as a 'co-pilots' for quants or finance professionals. Their main advantage appears to lie in relieving humans of tedious, repetitive and time-consuming tasks. This, in turn, frees up valuable time for more meaningful and complex efforts.

A recently released McKinsey report, 'The state of AI in 2023: Generative AI's breakout year' (August 2023), confirms this observation and demonstrates that current generative AI application areas do have low materiality but high efficiency gain (see overleaf).

Share of respondents reporting that their organisation is regularly using generative Ai in given function, %[1]



Most regularly reported generative AI use cases within function, % of respondents



Marketing and sales		Product and/or service development		Service operations	
Crafting first drafts of text documents	9	Identifying trends in customer needs	7	Use of chatbots (eg, for customer service)	6
Personalized marketing	8	Drafting technical documents	5	Forecasting service trends or anomalies	5
Summarizing text documents	8	Creating new product designs	4	Creating first drafts of documents	5

[1] Questions were asked of respondents who said their organizations have adopted AI in at least 1 business function. The data shown were rebased to represent all respondents.
Source: McKinsey Global Survey on AI, 1,684 participants at all levels of the organization, April 11–21, 2023

Sentiment analysis and further applications

Non-generative LLMs, like BERT, have proven highly successful in NLP tasks, such as **sentiment analysis of financial news** and other financial documents and communications. This application is less clear when it comes to generative LLMs. While sentiment analysis of nonspecific-domain text is certainly a strength of generative LLMs, challenges arise when domain expertise is required.

For instance, various studies have shown that ChatGPT's sentiment analysis of financial documents is less accurate than that of BERT, which is retrained on specific financial data, i.e., on similar documents, labelled for sentiment by humans or traditional NLP techniques. This is particularly evident when using 'naïve' prompts, such as: 'Score this document for sentiment, in terms of positive, negative or neutral.' ChatGPT's performance can be enhanced somewhat by using more creative or specific prompts or providing it with examples first. Such interventions are called prompt engineering, a topic we will discuss in more detail shortly.

In one example, the recent LSEG paper "Using GPT-4 with prompt engineering for financial industry tasks" (2023) shows that when provided with a few examples in its prompts, ChatGPT's ability to accurately score financial documents for sentiment improves slightly. However, other studies indicate that the success of this approach is somewhat limited. For instance, a recent study by a renowned sentiment provider revealed that no amount of clever prompt engineering for sentiment scoring of analysts' calls could elevate ChatGPT's scoring accuracy to the level of BERT's, which had been retrained on labelled dataset of such calls.

The studies mentioned above investigated sentiment classification by LLMs of financial documents related to specific companies, such as company reports or analyst call transcripts. The situation becomes significantly more complex for ChatGPT when dealing with other, more intricate asset classes such as foreign exchange rates, interest rates or commodities.

Sentiment analysis for commodities is notoriously difficult, as sentiment must correlate with supply and demand. In news related to commodity markets, what might seem positive (e.g., news about stricter child labour regulation in the Democratic Republic of Congo (DRC)) can have negative implications for a specific commodity (e.g., cobalt) and vice versa. Our own recent small-scale study on an extract of LSEG Machine Readable News (MRN), in combination with News Analytics, indicated that ChatGPT sometimes struggles to discern whether a piece of news indicates upward or downward pressure on a particular commodity market. To succeed in such nuanced tasks, an LLM must be retrained or fine-tuned on specific data, such as LSEG MRN.

Another exciting financial application of LLMs in the near future is **risk analysis and development of early warning risk systems**. The concept behind such systems is similar to sentiment analysis but on a more global scale, encompassing markets, indices, sectors or countries. Again, fine-tuning an LLM for this task will require a database of content which has direct relevance for heightened investment risk across diverse asset classes. Proprietary datasets designed for this purpose exist and we will elaborate on some of them below.

Using generative LLMs for **fraud detection in documents and transactions** is another promising yet highly sensitive application. Crucial considerations such as fairness and bias will play prominent roles in the design of such applications. We will elaborate on these issues and outline some recent bias mitigation strategies in the last section.

Looking further into the future, we come to the 'holy grail' of investment management – areas like **alpha generation and the creation of profitable investment strategies**. However, these applications appear less ripe for implementation than initially anticipated. They involve intelligence generation and LLMs, primarily designed for language generation rather than abstract reasoning and intelligence, currently face challenges with such tasks. Additionally, there are fundamental obstacles to overcome, which will be further discussed below.

But next we will address a crucial consideration when applying generative LLMs in finance: how to effectively retrain and fine-tune them for specific applications.

Adapting a generative LLM to your task

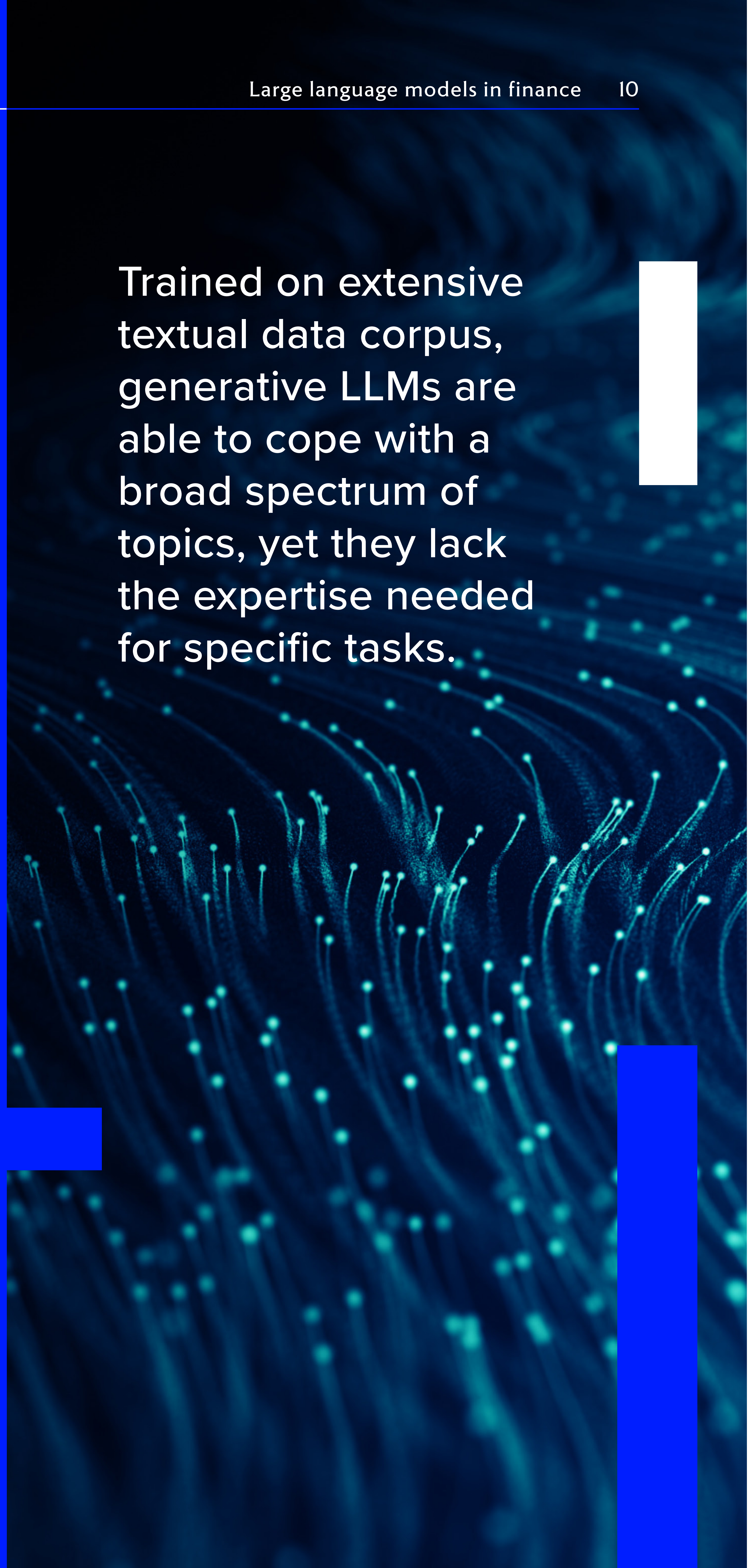
A central challenge in using LLMs for financial applications, including those mentioned above, lies in their generality and the absence of domain-specific knowledge. Trained on extensive textual data corpora, generative LLMs are able to cope with a broad spectrum of topics, yet they lack the expertise needed for specific tasks.

So how to deal with this major drawback? One apparent 'brute force' solution is to develop a bespoke large language model from scratch and train it on an extensive yet domain-specific textual corpus. Bloomberg GPT recently unveiled a model it has developed using this strategy, which generated high expectations in the finance community. While such an LLM is expected to outperform ChatGPT in financial tasks, it still needed a massive training dataset. Hence, it still grapples with the challenge of being too general and will likely lack domain expertise for highly specific tasks, such as sentiment analysis for a particular asset class (e.g., commodities) or generating regulatory-compliant model validation reports. Moreover, creating and training a new LLM is a hugely complex and costly task, with expenses ranging in the tens if not the hundreds of millions of USD.

An alternative approach involves taking an open-source LLM such as LLaMA and retraining or fine-tuning it using a domain-specific data corpus – like the approach taken with pre-trained BERT. This modification 'injects' the model with much-needed domain expertise. However, as discussed earlier, generative LLMs have several orders of magnitude more parameters than BERT. The retraining process, requiring modifications to all the model's weights (e.g., 7 billion for the small LLaMA), remains too CPU- and memory-intensive and thus expensive. At least, this was the prevailing wisdom among LLM enthusiasts until a revolutionary development, the introduction of low rank adapters (LoRA) and quantised LoRA (QLoRA), swept the rug from under that view. These ground-breaking techniques now allow for the efficient retraining or fine-tuning of a foundational LLM (such as LLaMA or Falcon) in a massively efficient way, in terms of both compute power and memory, making it feasible to achieve the much-needed domain knowledge at a relatively low cost. LoRA and QLoRA are such game changers in the fine-tuning of LLMs that we will explain them in more detail below.

Technologically, the advent of LoRA and QLoRA makes retraining and fine-tuning of LLMs economically viable and efficient. However, this still requires a large volume of high-quality, reliable and domain-specific data. The crucial role of the retraining data corpus cannot be overstated. Companies that have proprietary data or access to such data will benefit, as the verifiable and relevant data corpus becomes the main distinguishing factor among various generative LLMs and their modifications. Now more than ever, data is the new gold – and its necessity for adapting LLMs for finance applications will greatly amplify its value.

In a subsequent section we will present examples of LSEG datasets that appear especially valuable for LLM retraining and explore some potential applications of generative LLMs fine-tuned with these datasets. But first, we will provide a more in-depth explanation of fine-tuning LLMs.



Trained on extensive textual data corpus, generative LLMs are able to cope with a broad spectrum of topics, yet they lack the expertise needed for specific tasks.

Prompt engineering

The output of a large language model is highly sensitive to the prompts it receives. Therefore, we can craft prompts that will elicit the desired response. This approach does not entail any model modifications but relies on providing precise instructions to the LLM. Prompts should be engineered to steer the model toward the intended outcome, whether it is a particular format or style. Various techniques can be used to generate effective prompts.

Instruction-based prompts

In this approach, the goal is to provide a detailed and specific set of instructions. For instance, in sentiment analysis, an instruction-based prompt might look like this: ‘Evaluate the sentiment of this financial news (positive, negative, or neutral). Disregard non-financial information and focus on the influence of this news item on the company’s share price.’ The more specific and detailed the prompt, the higher the likelihood of an accurate response.

Role assignment

In this method, an LLM is asked to assume a specific professional role, like that of a trader or stock analyst. For instance: ‘Summarise this company’s quarterly report from the perspective of a stock analyst. Use only information within the document and do not exceed 100 words’ (this prompt is essentially a combination of role assignment and instruction-based methods).

One-shot vs. few-shot prompting

In this approach, a few-shot prompt includes initial examples of question-answer pairs (one-shot prompt is the prompt without examples). The examples serve as demonstrations of what is expected from the LLM. For instance, a few-shot prompt could be:

‘Toyota was fined for violating EU safety regulations.
A: Negative.

Analysts expect a modest impact of the EU’s fine.
A: Positive.

Toyota’s share price declined at the end of yesterday’s trading.
A: Negative.

EU safety fines may affect other sectors and companies.
A: Xxxxxxx.’

Chain-of-thought (CoT)

This technique requires the LLM to provide not only an answer to a question but also to explain the intermediate reasoning steps leading to that answer. It can be implemented with a one-shot prompt where a question is followed by the instruction ‘Let’s think step by step’, or with a few-shot prompt providing an example to demonstrate what is desired.

Lastly, there exist even more imaginative (or perhaps unconventional) methods to entice the LLM to do what you want. These can include flattery (as even LLMs are not immune to it!) or by employing a more sinister tone and resorting to tactics like blackmail or threats (even of shutting it down!).

Parameter-efficient fine tuning (PEFT): LoRA and QLoRA

LoRA and QLoRA are remarkably simple yet powerful techniques for the efficient and cost-effective fine-tuning of foundational LLMs for diverse specific tasks. LoRA was introduced by Microsoft researchers at the end of 2021 and the original paper describing it won the prestigious IEEE 2022 award for best paper. It is not surprising that, since recently, LoRA and QLoRA are recognised as the powerhouse behind LLM adaptation and fine-tuning, playing the key role in incorporating domain-specific knowledge into these models.

The elegance of the two main concepts of LoRA is truly remarkable. First, LoRA is based on a fundamental technique in matrix calculus: singular value decomposition. This decomposition is applied to the colossal matrix of weights (or more precisely, weight change matrix) to reduce the number of parameters that require adjustment by up to 10,000 times. Second, the LoRA approach involves collecting these smaller matrices of LLM parameter offsets corresponding to each specific task. These matrices are then added to the original model weights one by one to generate numerous variants of newly retrained LLM in one single step. Essentially, this allows you to preserve multiple sets of parameter changes, each tailored to a specific task. Consequently, you can switch between tasks by adding the corresponding parameter offset matrix to a foundational LLM, much like shifting gears on a fancy bike.

Let me explain both ideas in a more detail.

An LLM's parameters are contained in very large matrices; for instance, GPT-3 has 176 billion weights. When an LLM is trained, during each training epoch i, the weights are iteratively adjusted using gradient descent to bring the actual output closer to the desired output: Wi=W(i-1)+ΔW(i). This process is repeated many times until convergence is achieved. This training is very time consuming. One idea behind LoRA is to accumulate all ΔWi into a cumulative ΔWand adjust the weights in one collective step: Wnew=W(old)+ΔW. In this way we can store multiple ΔW sets, each tailored to a specific task (e.g., sentiment analysis of commodity news, rates, FX) and switch between them by adding the relevant ΔW to the foundational model's weights W. This concept is vividly illustrated by the hugginface.co image generation tool: LoRA The Explorer. It is powered by a single image generation foundational model, yet it allows for generation of any desired image in various styles, such as Ghibli or other animation styles, pixel art, Lego style, manga, watercolour and many more. Each distinct style is nothing more than a specific parameter offset ΔW and the foundational model switches between these offsets to create images in different styles.

But there is a problem with storing several massive matrices ΔW (theoretically, each of them has the same size as the original model weight matrix W): it is as memory-intensive as storing the original model. This is where LoRA's brilliance shines through: it decomposes the full matrix ΔW into two 'thin' matrices A and B: ΔW=A*B. For example, if W is 1000x1000, A could be 1000x3 and B is 3x1000. The lower dimension of A and B (in this example, 3) corresponds to the rank of ΔW, representing the number of 'independent' columns in it - uncorrelated 'main' factors, each carrying unique information about the original matrix not contained in other factors. This is the essence of information compression: the columns of ΔW are typically dependent, i.e., contain a lot of redundant information, which we compress into just a few independent factors. Such compression reduces the redundant information but can introduce some error in the 'recovered' matrix A*B. The original LoRA paper shows that the information loss is limited. Moreover, if a task is very precise and specific, one can allow for more factors, while less precise tasks can be dealt with using fewer factors (and hence, higher compression). The same paper show that the number of independent factors in a typical ΔW matrix for a GPT-3 model is 10,000 times lower than the number of original columns.

The QLoRA extension, introduced in May 2023 by hugginface.co, uses quantisation to further limit storage and compute time of LoRA without significant performance loss.

The open-source code for LoRA and QLoRA allows anyone to use these techniques in combination with an open-source foundational model such as LLaMA to create their own fine-tuned generative LLM. However, a high-quality training dataset specific to the task is still crucial. In one of the following sections we will review some of such datasets.

Logit processing

Chatbot-based LLMs such as ChatGPT are designed to sustain a free-flowing conversation, which is what makes it so appealing to the general public. However, this feature can be a major drawback in business applications, where the desired answer is often numerical rather than textual. For instance, in sentiment analysis, the goal is to obtain sentiment scores or probabilities along with a confidence measure. Another example is extracting numerical information (e.g., maturity date) from a large text document, such as a derivatives term sheet. However, generative LLMs are naturally verbose, making it difficult to restrict them to providing concise numerical answers without unnecessary text. This is also known as **text-to-data** issue.

Clever and iterative prompt engineering, involving a series of specific instructions to constrain the output format, works for some LLMs like ChatGPT but not for others like LLaMA or Falcon. Moreover, obtaining the correctly formatted answer in a single step is unreliable, as it significantly restricts the LLM's output. In such cases, a relatively new technique called **logit processing** becomes helpful. This approach, used in combination with prompt engineering at the first step, dynamically adjusts token probabilities to constrain grammar and achieve the desired format conversion. As a result, it supresses verbose answers and ensures the requested formatting of the LLM's output.

Logit processing involves tweaking the LLM's parameters to influence the probabilities of output tokens. While the model determines the most likely output (token) given the context, these outcomes can be guided towards the desired output format. The parameters that can be tweaked to adhere to a prespecified format include:

- **Parameters that influence when to stop generating text.** These set the number of tokens to generate or stop words.
- **Parameters that limit LLM creativity and make it more 'predictable'.** These parameters include temperature, top-K, top-P and beam search width. A higher temperature parameter ensures more high-probability tokens are used in the output, restricting the LLM's 'creativity'. The top-K parameter is the limit on the number of most probable options for the next token, so the lower it is, the more predictable the output. The top-P parameter, used in combination with top-K, allows the LLM to randomly choose a set of the most likely tokens whose cumulative probability is equal to or exceeds top-P. Top-P parameter mitigates the restrictiveness of the top-K parameter. The beam search width determines the number of options considered as candidates at each step, thereby liming the diversity of the outputs.
- **Parameters that reduce repetition in the output**, such as repetition penalty. It penalises repeated tokens in the output, restricting the verbosity of the answer.

LSEG datasets for LLM applications

We have repeatedly stressed the crucial value of reliable and relevant data for adapting LLMs for specific applications. Many such relevant datasets are available, such as on the ISDA Create platform mentioned previously. However, the distinguishing feature of re-trained and fine-tuned models will lie in the use of proprietary training data. Some firms – particularly data providers – are uniquely positioned to leverage such data.

Datasets needed for successful LLM applications in finance can be categorised into three types based on their nature and utilisation in LLM applications:

- **Textual data** sourced from specific financial domains. These may include, e.g., analyst call transcripts, financial news, company quarterly reports and filings, financial contracts specifications, model validation reports and other finance-related texts.
- **Data derived from models and analytics**, which are often labelled datasets. These datasets assign specific characteristics to individual data points using models like classifiers. For instance, a dataset of financial news might have each piece of news tagged for a specific company and labelled for sentiment (positive, negative or neutral) by an NLP algorithm or a machine-learning model. Another example could be a dataset of companies along with their credit ratings, determined through company fundamentals, financial ratios and related data, all encapsulated within the dataset itself.
- **Quantitative and financial data** encompassing firm fundamentals, financial ratios, earnings and other structured, usually numerical, tagged data.

The utilisation of these three data types in LLM applications varies significantly. The first type is valuable for unsupervised learning or fine-tuning an LLM, enabling it to 'learn the financial lingo', i.e., to interpret intricate financial language which is, often distinct from other types of text on which the model was initially trained. Having learned this specific financial jargon, a fine-tuned LLM becomes more adept at comprehending or even generating similar texts upon request, such as crafting a new model validation report or derivative term sheet.

The second data type is the labelled data, crucial for training any ML model to perform a precise task. It serves as a means to train an LLM to perform exactly the same task as the original model or a classifier – like labelling financial news for sentiment or assessing relevance to a particular entity or assigning a credit rating to a company based on its fundamentals and ratios. In any ML application, a large and reliable labelled dataset holds immense value and in LLM applications are no exception.

The final data type functions as reference databases, addressing issues such as timeliness and the hallucination tendencies of LLMs (discussed further below). In short, timeliness pertains to LLMs having been trained on data only up to a specific historical point, making them unable to incorporate the most up-to-date information in their responses. Hallucinations refer to the tendency of LLMs to produce seemingly plausible yet inaccurate information. Both challenges can be managed through a technique known as Retrieval-Augmented Generation (RAG). Here, an LLM's answer is cross-checked with an external and credible database. It enables the model to access the most current and reliable information outside its training dataset. To illustrate this, think of RAG as akin to an open book vs closed book exam: in RAG, the model searches through an 'external book' (external database) for answers, unlike the traditional approach where it tries to generate answers from memory, with all possible unfortunate consequences (such as not knowing the answer and making it up). Access to an external database also allows for the verification of accuracy in LLM-generated answers and ensures trust. An added advantage of RAG is that it does not require an LLM to be constantly retrained on newly available data, avoiding associated high costs and computational time.

LSEG, being the world's largest financial data provider, has a multitude of datasets, some combined with advanced analytics, which cover all three categories and are excellently suited for fine-tuning and adapting LLMs for various financial industry tasks. On the next page I will give some examples of such datasets for each of the three abovementioned categories.

Textual data

LSEG's vast filings database comprises corporate disclosures from more than 65,000 publicly traded companies and contains an extensive collection of textual resources. The latter currently exceeds 2.8 million documents and over 20,000 new ones are added daily.

Another, similar textual data resource is the StreetEvents dataset, which includes corporate disclosures and brokerage events from over 18,000 public companies across 90 countries. It contains a wide array of content including conference call transcripts and summaries, SEC filings and various corporate event summaries.

Data combined with analytics examples

Machine Readable News and News Analytics

LSEG Machine Readable News is a unique historical and real-time database of all news alerts, stories and updates that come over the Reuters newswire for companies, commodities, macroeconomic news and markets, including foreign exchange, fixed income, futures and equities. It includes up to 10,000 stories a day from 131 countries as well as corporate and regulatory announcements from over 40 third-party newswires. Stories include a wealth of metadata, such as topic codes, of which over 2,300 are used. Each news item has a unique identifier, enabling synchronisation of the database with Reuters News Analytics.

For each news item Reuters News Analytics includes entity relevance, sentiment, novelty, volume of news and other NLP-based scores. It covers 56,000 companies and all commodities. It also scores macroeconomic news and provides a wealth of metadata such as topic and country codes and many more. This database, in combination with MRN, is an excellent labelled training set for fine-tuning LLMs to perform sentiment scoring of financial content (both news and social media, as well as others) for various asset classes.

StarMine Text Mining Credit Risk Model

This remarkable model assesses the credit risk in publicly traded companies via NLP techniques, using Reuters News, StreetEvents conference call transcripts, corporate filings and broker research reports to predict which firms are likely to undergo financial distress and which are likely to thrive. It is a percentile ranking (1-100) of stocks, with 100 corresponding to the healthiest companies. Again, this dataset can be used as a labelled training set to fine-tune an LLM to provide credit ratings on the basis of similar textual information.

Many other datasets use advanced analytics to provide both labelled and quantitative data, such as StarMine Quantitative Models. Their creative use in LLM fine-tuning is still to be explored.

Quantitative and financial data

This data category is excellently represented in the LSEG feed: it has a wealth of diverse quantitative databases that can serve as prime resources for RAG. The most expansive datasets include, among others:

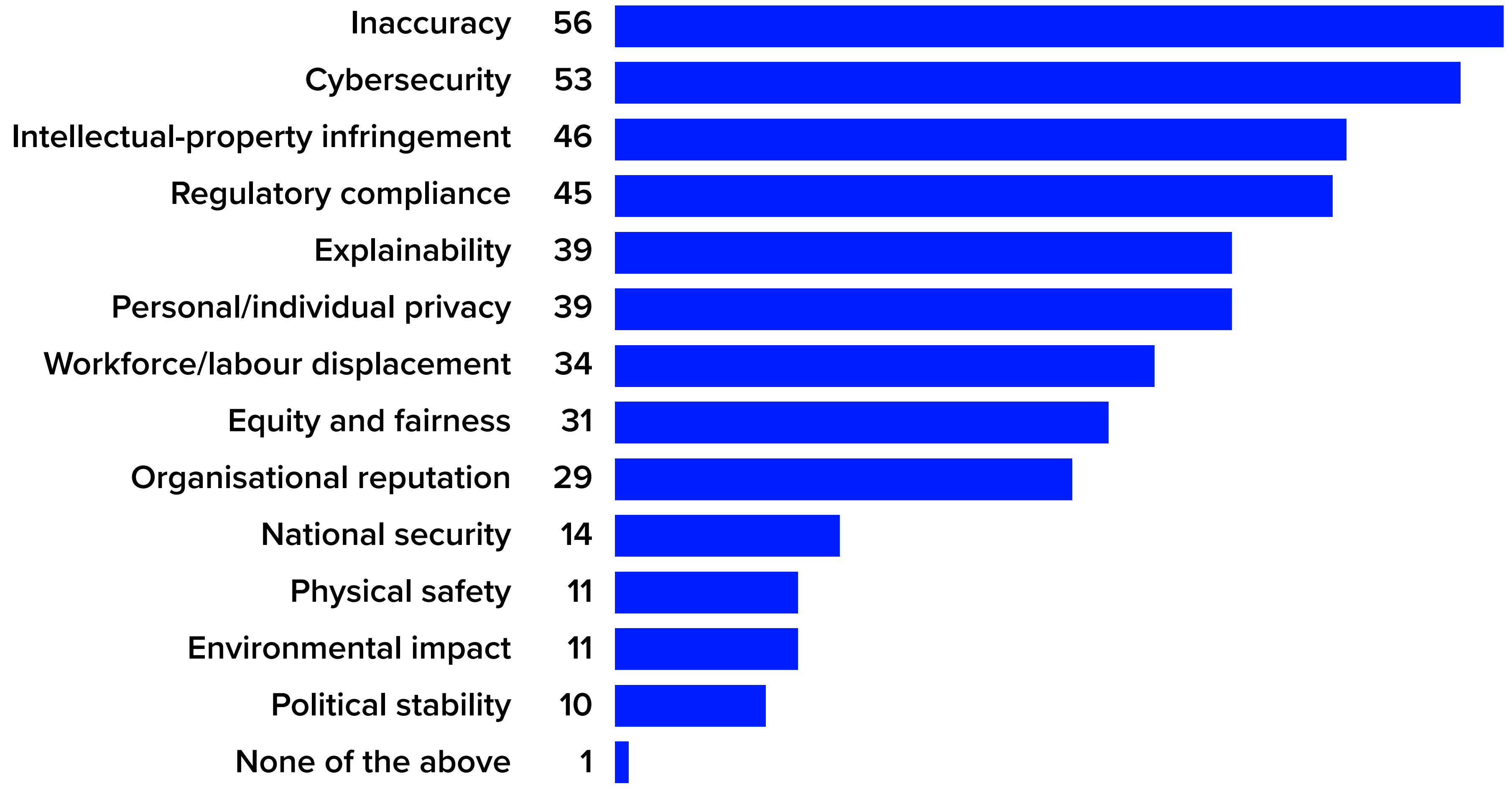
- Company fundamentals, providing global coverage of publicly traded firms
- Deals such as M&As, bond and equity issues, private equity and venture capital activity
- Lipper Global Data Feed, which provides detailed information about over 200K funds, including portfolio allocations and fund performance
- LSEG Business Intelligence, commodity and energy fundamentals and forecasts.

Using this wealth of data creatively can help to develop fine-tuned LLM variants which are effective in numerous financial applications.

Pitfalls of using LLMs and mitigation strategies

We have established that LLMs can be cost-effectively trained on relevant datasets to become useful in a variety of financial applications. However, several serious issues must be addressed for these models to be used in a real business environment. The McKinsey report summarises the risks related to LLMs that companies identify, as shown in the graph below.

Organisation considers risk relevant



The graph identifies issues that impact companies in all sectors but we will examine those risks and mitigation strategies that are particularly relevant to finance, asset management and investment management.

Timeliness

LLMs are trained on textual data generated up to specific point in time and remain static thereafter – ChatGPT 4, for instance, is trained on data up to September 2021. Consequently, the responses generated by LLMs do not incorporate the latest information about financial markets, regulation, geopolitical developments or other real-time updates. This may change in the future if LLMs are retrained more frequently to include recent information. However, the lengthy and costly nature of the retraining process of foundational models means that there will still be some delay, albeit less than at present.

Approaches to mitigate this challenge are similar to those applicable to the next issue: hallucinations, which we will address next.

Hallucinations

Since the release of ChatGPT 3.5, it has become apparent that the model is prone to generating plausible-sounding but factually incorrect answers. Numerous amusing examples of this phenomenon, known as 'hallucinating', can be found on the internet. In finance applications, where reputation and trust are paramount and factual correctness is crucial, dealing with hallucinations becomes a significant challenge.

So, how can we address this issue? Although it remains a complex problem, some methods are starting to emerge. One approach is based on combining the LLM's answers with web or database searches. This technique, known as Retrieval-Augmented Generation (RAG), involves incorporating the retrieval of external data (from the internet or a relevant database) as part of the processing chain. Note that this technique can be also used to tackle some timeliness issues.

Care should be taken when formulating suitable prompts. Even for tasks not requiring external information retrieval, it is often difficult to guide the model to follow multiple instructions in a single prompt. In these cases, chains of prompts provide a way to progress from the initial question to the accurate response through multiple iterations. While chains are optional for some tasks, they are necessary for implementing RAG as they are the only viable instrument for retrieving additional information and incorporating it in the final response.

Another useful intervention to limit hallucinations when no relevant information is available is to explicitly instruct the LLM to respond 'I do not know' if the analysed document lacks the requested information, rather than generating speculative or inaccurate answers.

Problem of back-testing: the main hurdle in applying LLM to investment strategies

As soon as ChatGPT was released, investors and asset managers began contemplating the possibilities of using it to develop profitable investment and trading strategies – but these seem to be quite some way in the future. The first challenge is the constantly-moving nature of financial markets. The timeliness issue, as discussed above, means that the LLM cannot keep up with these rapid developments. Beyond the timeliness problem, there exists a more fundamental challenge when applying LLMs to the design of trading and investment strategies.

At the core of quant investing lies the concept of back-testing – analysing how an envisioned strategy would have performed in the past by using historical asset price data. Quant analysts simulate the performance of their strategies as if they were operational in the historical context, collecting various performance metrics such as return, alpha, volatility, drawdown, tracking error and ratios such as Sharpe or information ratio. Based on these simulated investment results and the assumption that past performance is indicative of future outcomes, they decide whether the suggested strategy is viable.

The critical aspect of back-testing design is careful elimination of any forward-looking bias. In other words, one must ensure that at each point in time the strategy does not incorporate any forward-looking information. Unfortunately, this is inherently impossible with LLMs. Foundational LLMs are trained on an entire textual data corpus up to a specific date (e.g., September 2021 for ChatGPT4) and then their weights are frozen. For back-testing of a strategy constructed with the help of ChatGPT or another LLM, we need, at each historical point in time, the state of this LLM at that point, i.e., trained solely on data available up to that point but not beyond. But this deviates from the standard training process of LLMs. Consequently, there will be inevitably a forward-looking bias in back-testing, potentially leading to strategies appearing more profitable than they truly are. Even if a fine-tuned LLM is used for strategy design, allowing for moving window retraining simulation, the foundational model's forward-looking bias remains inherent and cannot be eliminated.

If new LLM-based strategies are tested in real time, the testing process would not have a forward-looking bias. However, evaluating the performance of such strategies remains difficult because there will not be sufficient out-of-sample history available. This lack of historical data beyond the present moment makes it difficult to assess the strategy's robustness and performance in various market conditions.

Model risk, explainability and fairness

The use of LLMs in a financial institution's model landscape introduces considerable model risks arising from various factors, such as representativeness of the training data, correct and appropriate use of the model, model robustness in different market conditions and many more. Assessing model risk is an important requirement for many financial institutions and a subject of stringent regulation.

Machine-learning models bring another level of complexity and new risks into the model landscape. Regulatory bodies and financial institutions are particularly concerned about two aspects of machine learning models not shared by traditional models: the lack of explainability and transparency, as well as potential for bias and unfairness – issues greatly amplified in the case of LLMs.

Explainability and transparency

Explainability in machine learning refers to one's ability to understand and interpret a model's outcome based on input data. Traditional models like logistic regression can provide clear explanations in terms of feature importance and how they affect the model outcome. For example, a logistic regression for credit acceptance is easily explainable in terms of odds ratios of the applicant's features: if salary and LtV are significant determinants of credit quality, a credit might be rejected if the applicant's salary is 20% below average or loan-to-value of the house is 20% above average.

Excellent and well-established explainability tools for 'small' machine learning models, such as the famous SHAP (Shapley Additive Explanations), LIME (Local Interpretable Model-agnostic Explanations) and Counterfactual Explanations, are not readily applicable to LLMs due to their sheer size and complexity. Researchers and practitioners are currently developing explainability tools suited to the unique characteristics of LLMs.

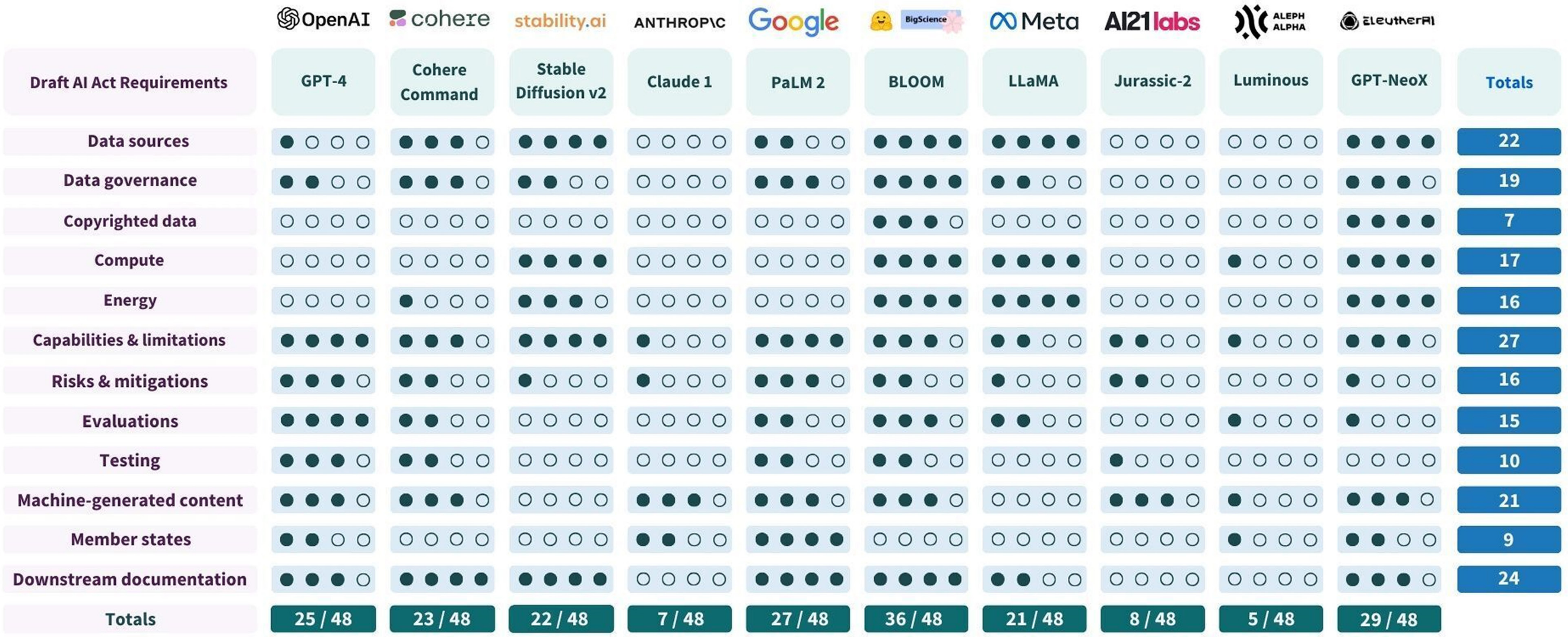
Two emerging explainability tools for LLMs are called **Attention** and **Attribution Visualisation** and both are designed for transformer architectures. The former attempts to visualise how much attention the entire model or an individual layer pays to each token in the sequence using attention matrices and heat maps. However, this method is still far from being truly 'explainable'. The information at the individual token level is too granular and not easily interpretable and the resulting matrices and heat maps are complex and hard to read. Attribution Visualisation is also performed at the level of tokens, so it suffers from the same shortcomings as the Attention Visualisation. While Attribution Visualisation may flag inconsistencies or anomalies in the model's outcomes, it cannot provide explanations for the model's predictions or outcomes, especially regarding the significance of large-scale features (rather than tokens).

The amendments proposed by the European Parliament as part of the ongoing negotiations of the EU AI act – which would be the world's first regulation of AI – outline the rules of the AI use in different risk categories. It contains transparency requirements for generative AI such as:

- Disclosing that the content was generated by AI
- Designing the model in such a way as to prevent it from generating illegal content
- Publishing summaries of copyrighted data used for training

The Stanford Center for Research on Foundation Models has recently developed The Foundation Model Transparency Index, based on various characteristics of data, compute, model and deployment, and assigned scores to various LLMs. The summary of their findings is presented in the figure overleaf. It shows a great variety of scores across models but also demonstrates that almost all of them fall short of these requirements.

Grading foundation model providers’ compliance with the draft EU AI act



	OpenAI	cohere	stability.ai	ANTHROP\C	Google	BigScience	Meta	AI21labs	ALEPH ALPHA	EleutherAI	
Draft AI Act Requirements	GPT-4	Cohere Command	Stable Diffusion v2	Claude 1	PaLM 2	BLOOM	LLaMA	Jurassic-2	Luminous	GPT-NeoX	Totals
Data sources	●○○○	●●●○	●●●●	○○○○	●●○○	●●●●	●●●●	○○○○	○○○○	●●●●	22
Data governance	●●○○	●●●○	●●○○	○○○○	●●●○	●●●●	●●○○	○○○○	○○○○	●●●○	19
Copyrighted data	○○○○	○○○○	○○○○	○○○○	○○○○	●●●○	○○○○	○○○○	○○○○	●●●●	7
Compute	○○○○	○○○○	●●●●	○○○○	○○○○	●●●●	●●●●	○○○○	●○○○	●●●●	17
Energy	○○○○	●○○○	●●●○	○○○○	○○○○	●●●●	●●●●	○○○○	○○○○	●●●●	16
Capabilities & limitations	●●●●	●●●○	●●●●	●○○○	●●●●	●●●○	●●○○	●●○○	●○○○	●●●○	27
Risks & mitigations	●●●○	●●○○	●○○○	●○○○	●●●○	●●○○	●○○○	●●○○	○○○○	●○○○	16
Evaluations	●●●●	●●○○	○○○○	○○○○	●●○○	●●●○	●●○○	○○○○	●○○○	●○○○	15
Testing	●●●○	●●○○	○○○○	○○○○	●●○○	●●○○	○○○○	●○○○	○○○○	○○○○	10
Machine-generated content	●●●○	●●●○	○○○○	●●●○	●●●○	●●●○	○○○○	●●●○	●○○○	●●●○	21
Member states	●●○○	○○○○	○○○○	●●○○	●●●●	○○○○	○○○○	○○○○	●○○○	●●○○	9
Downstream documentation	●●●○	●●●●	●●●●	○○○○	●●●●	●●●●	●●○○	○○○○	○○○○	●●●○	24
Totals	25 / 48	23 / 48	22 / 48	7 / 48	27 / 48	36 / 48	21 / 48	8 / 48	5 / 48	29 / 48	

Fairness and bias

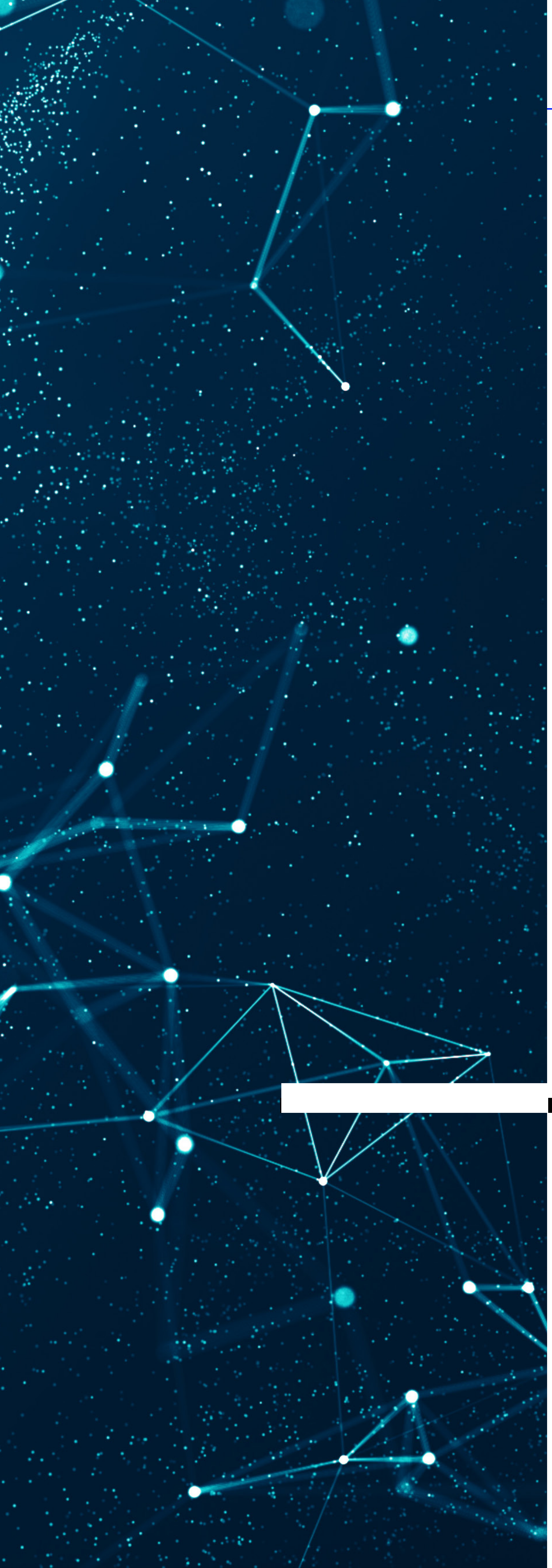
Machine learning algorithms, including LLMs, learn patterns from historical data which they then perpetuate. This means that any potential bias, i.e., unfair treatment of different groups of individuals, reflected in past data will be replicated – and possibly amplified – in the outcomes of a ML algorithm – and LLMs are no exception. Avoiding such disparate treatment of certain groups of society – based on protected attributes such as gender, race or age – is at the heart of AI fairness. At the end of 2019, the ECB introduced Ethics Guidelines for Trustworthy AI. Other central banks have followed suit and issued similar guidelines for the use of AI in the financial sector.

Measuring bias in the LLM’s outcomes is the important first step. There are three formal definitions of the model’s fairness: independence, separation and sufficiency and thus three ways of measuring bias – all three directly applicable to LLMs. However, mitigating bias is more complex for LLMs than for smaller machine learning models.

Among the three classes of bias mitigation algorithms – pre-processing (manipulating training data), in-processing (modifying the model) and post-processing (modifying model outcomes) – only post-processing mitigation algorithms are directly applicable to LLMs. While existing post-processing bias measurement and mitigation techniques can be used directly, new methods tailored to LLMs are emerging, such as ConstitutionalChain by LangChain (also called Self-critique chain with constitutional AI) or Weights-and-Biases (W&B) tracer.

Generative LLMs present a unique ethical challenge: controlling generated text, e.g., to prevent offensive or inappropriate content. Various methods addressing this issue appear every day, such as applications of reinforcement learning from human feedback (RLHF), direct preference optimisation (DPO), token penalisation, specific prompt engineering and others.

Addressing transparency and fairness challenges is crucial for financial institutions to manage model risk, comply with regulations and ensure responsible use of LLMs.



Other issues

Sustainability and power consumption

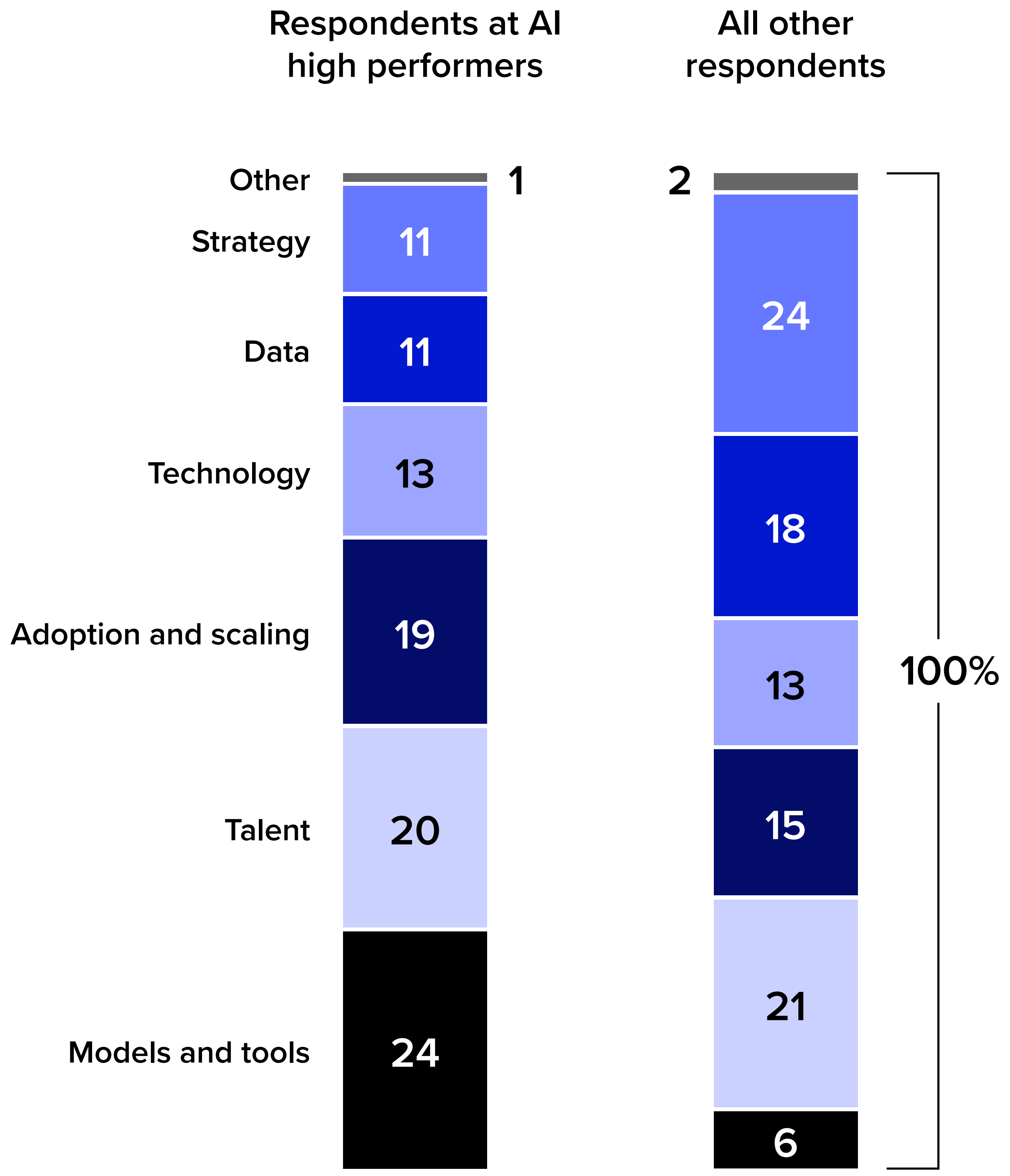
There has been much discussion about the amount of power consumed by Bitcoin mining but the training of LLMs requires even more GPU, compute power and storage – and as their use becomes more prevalent, this problem will only get worse. Presently, data centres consume an estimated 6% of global power supply; by 2030 this percentage is projected to increase to 13%. So, sustainable solutions must be found to meet these energy requirements. There are already compute power and cloud providers operating entirely on renewable energy and hopefully this trend will continue in the future.

Commercial solutions for GPU, cloud computing and retraining LLMs: the debate over in-house vs outsourcing

Presently, companies encounter many challenges in extracting value from LLMs, often stemming from a lack of internal capabilities (such as technological solutions, powerful computers and storage, knowledgeable personnel, training data) needed to develop, implement and maintain LLMs. According to the McKinsey survey discussed earlier, companies that already use AI identify models, tools and talent as the main challenges they face (see graph below). Among companies not yet generating value from AI, the lack of strategy or vision is cited as the main challenge, alongside a shortage of necessary models, tools and data.

Multiple commercial and semi-commercial solutions are rapidly emerging for GPU, cloud computing, training data, data lakes, models and tools, complemented by numerous open-source alternatives. Deciding whether to develop or fine-tune an LLM in-house or to outsource the work to an external solution provider is an important and complex dilemma that needs careful consideration. Organisations should be mindful and prudent in selecting their trusted AI or LLM partner, as it seems that everyone is ready to jump on the LLM bandwagon, including consultants and service providers without any proven track record in technology or AI.

Element that poses the biggest challenge in capturing value from AI
% of respondents



Conclusion

Generative LLMs represent an exciting technological breakthrough with the potential to change how we operate in ways that we cannot yet imagine. For now, they act more like co-pilots, relieving us from having to perform tedious and time-consuming tasks.

The key element for the successful application of LLMs in financial services is their retraining or fine-tuning by brilliant new tools such as LoRA. Trustworthy, domain-specific and labelled data will play a crucial role in this process. It is anticipated that, in the domain of LLMs, unique training data will be the main differentiating factor, rather than specific models or methodologies.

LLMs come with a range of challenges, ranging from hallucinations to possible bias to many more. However, techniques for mitigating these issues are rapidly emerging, so it is important to stay informed about these developments.

On a broader note, another caveat regarding LLMs centres on their limited ability for abstract reasoning. It is important not to overestimate the current capabilities of LLMs such as ChatGPT and not to rely on them for material or sensitive applications at this point.

Companies, especially those in the financial sector, face many challenges in extracting value from Generative AI. These challenges range from lack of suitable models and tools to acquiring necessary talent and expertise, obtaining relevant training data and formulating a broad AI and LLM strategy. To navigate these challenges and explore the potential of LLMs, it is important to become and stay informed and it is my hope that this essay has been a helpful step in that direction.

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention is all you need, 2017. https://arxiv.org/abs/1706.03762

Using GPT-4 with prompt engineering for financial industry tasks. LSEG Analytics, May 2023. https://solutions.yieldbook.com/content/dam/yieldbook/en_us/documents/publications/using-chatgpt-with-prompt-engineering.pdf

The state of AI in 2023: Generative AI's breakout year. McKinsey survey, August 2023. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2023-generative-ais-breakout-year

Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen. LORA: LOW-RANK ADAPTATION OF LLMS. Microsoft Corporation, October 2021. https://arxiv.org/pdf/2106.09685.pdf

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs, 2022. https://arxiv.org/abs/2305.14314

PROBABILITY
& PARTNERS

LSEG DATA &
ANALYTICS