TECHNOLOGY



Unstructured Data and AI

Fine-Tuning LLMs to Enhance the Investment Process

Brian Pisaneschi, CFA

CFA Institute
Research & Policy Center

INTRODUCTION

The explosive growth of unstructured data has reshaped the way investment professionals think about the sources of information that go into their investment process. In a July 2023 CFA Institute survey[1] on alternative and unstructured data, 55% of investment professionals reported incorporating unstructured data in their workflow and 64% indicated using alternative data.

The use of alternative data can be traced back to the early 1980s with the dawn of quantitative investing. Following the mid-20th century adoption of fundamental analysis, analysts began seeking additional data sources to secure a competitive edge. Such strategies as counting cars in parking lots to determine footfall and predict sales became an integral part of a fund's alpha-generating process. The adoption of data-driven strategies was further accelerated by the advent of the computer and the emergence of statistical arbitrage and algorithmic trading. Today, an analyst has access to an unprecedented amount of data; the digital revolution has led to exponential growth in data generation. According to the International Data Corporation (IDC), data volume was expected to grow by 28% in 2023, roughly doubling every 2–3 years (Muscolino, Machado, Rydning, and Vesset 2023). Moreover, they estimate that 90% of the data being generated are in an unstructured form, hindering the ability to be widely used. In fact, in an earlier study, the IDC estimated that a mere 0.5% of the generated data were actually being used for analysis (Gantz and Reinsel 2012). The limited extent of data used for analysis calls into question the extent of market efficiency (see, e.g., Wigglesworth 2023).

Over the past few decades, the predominant approach to financial analysis has centered on leveraging structured, numerical data. As this method has become commonplace, the ability to achieve a competitive edge and create differentiated value has become increasingly challenging and complex. As the digital revolution continued, new alternative data providers sprouted up, capitalizing on the notion of data being the "new oil." The exponential growth of unstructured data boosted demand for methods to process and extract valuable insights, leading data science to emerge as a highly sought-after domain of expertise within investment firms.

Early adopters of alternative data were confronted with a critical "buy vs. build" dilemma, facing the decision to either purchase vendor data or invest in developing in-house data science capabilities. This decision was heavily influenced by the notable talent gap in the industry. Research conducted by the Hong Kong Institute for Monetary and Financial Research (2021) on AI and big data showed that in the Asia-Pacific region alone, the talent pool

[1]In July 2023, 40,000 global constituents of CFA Institute were asked to participate in a survey centered around advances in AI, unstructured and alternative data, and the open-source tools and datasets most valuable for their workflows. 1,210 responses were received for a participation rate of 3%. 95% of the responses received were CFA charterholders. The full survey dataset is available exclusively to members on the Research and Policy Center website. Select findings from the survey are contained in this report.

 © 2024 CFA Institute. All rights reserved.

must grow by 23% annually to bridge the gap. In response, some firms have adopted a T-shaped teams approach (Cao 2021), promoting a workforce adept in specialized knowledge and capable of cross-disciplinary collaboration, to bridge the gap. Despite these efforts to cultivate such teams, the talent gap is likely to remain significant, underscoring the complexity of the buy vs. build conundrum.

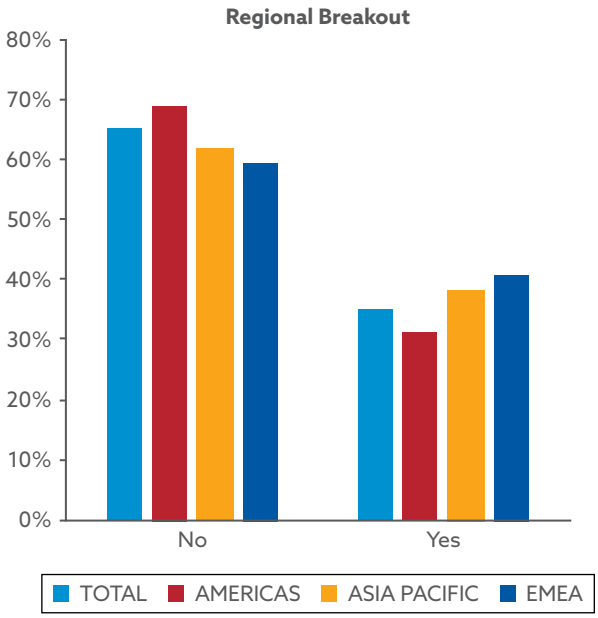
In recent years, the field of natural language processing has witnessed remarkable advancements. For example, the dawn of transformer architecture dramatically increased the contextual awareness of language processing models and gave rise to foundation models, like OpenAI's ChatGPT. These breakthroughs dramatically altered the role and significance of NLP when using alternative data. The progress can largely be attributed to three factors: the development of more sophisticated algorithms, the availability of more extensive and diverse datasets, and the exponential growth in computational power. One major contributing factor to the developments in algorithms has been the thriving open-source community. Researchers and developers worldwide collaborate and share their work on online platforms, facilitating the collective development and dissemination of cutting-edge tools, techniques, and models. This collaborative environment has accelerated the pace of NLP development, transforming NLP into a powerful tool capable of unlocking valuable insights from vast amounts of unstructured textual data. As a result, NLP has become an indispensable resource for professionals in various industries.

The combination of advances in NLP, the exponential rise in computing power, and the thriving open-source community has led to the emergence of generative artificial intelligence models. The rapid adoption of GenAI technology in the investment industry has placed it at the forefront of everyone's attention. **Exhibit 1** shows the total and regional breakouts of participant responses from the July 2023 CFA Institute survey to the question of whether they have used GenAI tools, with 35% of all participants indicating they have done so.

These state-of-the-art models have democratized access to powerful AI capabilities, empowering even lay programmers to rapidly iterate and experiment with new ideas, thereby transforming the traditional dynamics surrounding the choice of buy vs. build. In fact, in the same CFA Institute survey, 18% of participants indicated that ChatGPT has directly influenced them to take on projects they would have otherwise deferred to a specialist and 12% indicated that it has influenced their buy vs. build decision. This evidence highlights the necessity to experiment and evaluate the capabilities of these new models in relation to older tools and vendor solutions.

This paper aims to provide readers with a comprehensive framework to understand and use the tools necessary for ethically building investment models in the open-source community. The first chapter begins with essential background knowledge and information required to start building open-source projects. It introduces alternative and unstructured data, clearing some confusion on their definitions, and addresses the importance of ethical

Exhibit 1. Have You Used Generative AI Tools Like ChatGPT? (survey responses)



Note: 1,210 responses.

Source: From the July 2023 CFA Institute survey on alternative and unstructured data. See footnote 1 for more details.

considerations while handling these models. The second chapter focuses on the quantitative methodologies that underpin the advancements in NLP, presenting a range of options for developing models and exploring the practical applications that can assist in the buy vs. build debate.

Finally, we examine a case study that showcases the tools at our disposal in action and demonstrates their potential in driving the evolution of conversations surrounding environmental, social, and governance (ESG) issues. This area in finance is full of opportunities to apply AI and ML because a large portion of the data remain unstructured and fragmented. Additionally, the data are often lagged and self-reported and present time, managerial, and other biases.

This paper will provide readers with a solid understanding of alternative and unstructured data and the quantitative techniques currently supporting their use.

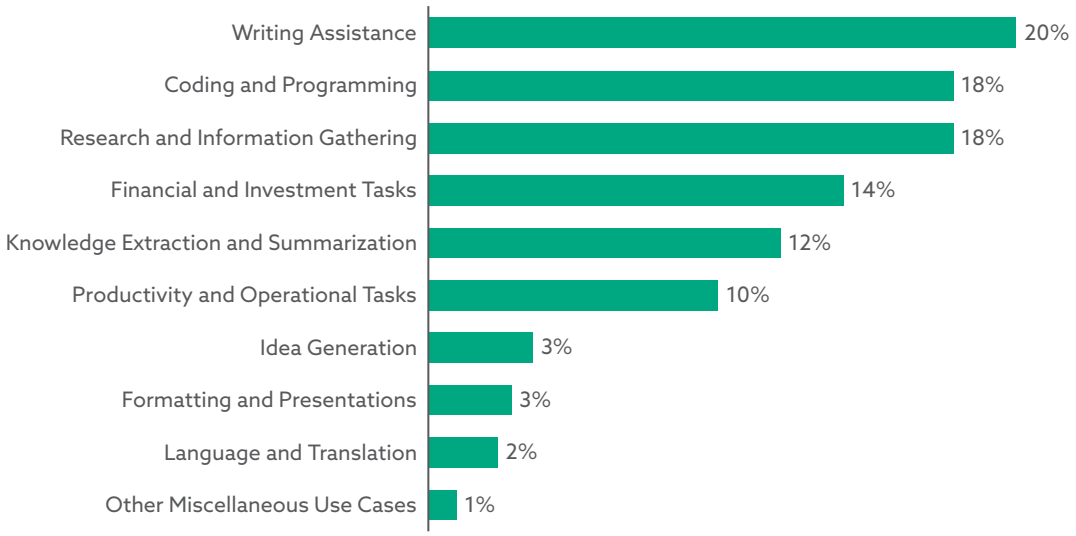
1. UNSTRUCTURED, ALTERNATIVE, AND OPEN-SOURCE DATA

Generative AI has ushered in a new age of programming and development, empowering even those with minimal programming experience to craft their own projects. This groundbreaking technology can generate code, automate repetitive tasks, and offer guidance to help newcomers navigate the complexities of software development. Survey data (**Exhibit 2**) show that coding is the second most popular ChatGPT use case for investment professionals. However, having these tools at our disposal can lead to a false sense of security. There is still a great need to have a solid foundation in programming principles to truly understand the generated code and ensure its accuracy and relevance.

Building a solid foundation in programming allows developers to identify potential issues, optimize performance, and fine-tune code generated by AI. This understanding becomes particularly important when testing and debugging because AI-generated code will contain errors and "hallucinations," whereby the AI might produce code that seems reasonable but does not accurately accomplish the intended task or meet the desired specifications.

It may be tempting to avoid coding and wait for generative AI to eliminate the need to code altogether; however, a generalist knowledge of programming will remain important even as AI advances beyond its current stage. This knowledge

Exhibit 2. ChatGPT Use Reported by Investment Professionals (survey responses)



Note: 299 responses.

Source: From the July 2023 CFA Institute survey on alternative and unstructured data. See footnote 1 for more details.

© 2024 CFA Institute. All rights reserved.

will still be necessary because competitive advantage is gained at the edge of existing technologies, current AI takes time to train on new information, and technologies constantly evolve. Thus, staying up to date with the latest programming languages, frameworks, and development methodologies is imperative.

Python has solidified its position as the leading programming language in data science, ML, and AI applications, making it the language of choice for anyone looking to leverage the vast open-source tools available. In the July 2023 CFA Institute survey, 70% of respondents chose Python as their preferred language to deal with unstructured data, followed by R at 11%; the other preferred languages chosen were Excel, VBA, SQL, and proprietary software. Up until the last five years, R was in direct competition with Python as the preferred tool for investment professionals due to its rich statistical libraries and ease of use in data analysis. However, with the exponential growth of the Python open-source community and its widespread adoption in AI, ML, and data science, Python has emerged as the leader, offering a more versatile and comprehensive tool kit for professionals across industries. Data from GitHub shows that the number of Python repositories has grown from less than one million in 2015 to approximately seven and a half million in 2022, whereas R repositories are still less than one million as of 2022.

The open-source community, centered around such platforms as GitHub, has played a pivotal role in developing and proliferating AI, ML, and data science tools. GitHub is used by developers to collaborate and to contribute and share their work, housing an ecosystem that drives innovation and growth in the field.

Libraries, such as pandas, BeautifulSoup, and scikit-learn, are popular examples of open-source tools actively used by investment professionals for various purposes. Pandas is the Python replacement for Excel, allowing users to import, clean, and transform financial data at scale. BeautifulSoup is a web-scraping library that allows investment professionals to extract data from websites, such as financial news articles, stock prices, and economic indicators. A similar library, Selenium, is another web-scraping library that is particularly important when the data needed are embedded in JavaScript, as the library mimics a user's interaction with a website that allows it to access dynamic content. Scikit-learn, in contrast, is a comprehensive ML library that offers a wide range of algorithms and tools for data analysis, modeling, and evaluation.

While investment professionals have long used these traditional open-source libraries, a surge of innovation has occurred in the AI community that is pushing the boundaries of what is possible with these tools. One notable example of how the AI community has evolved within open source is the emergence of such start-ups as Hugging Face. Hugging Face has made a significant impact on the open-source AI community by serving as a platform for models and datasets. Through its highly popular Transformers Python library and user-friendly online interface, Hugging Face gives researchers and developers easy

access to a vast array of ML models, tools and resources, democratizing their access in a collaborative environment.

One such ML model on Hugging Face that gained interest in the investment community is FinBERT, a language model specifically designed to address tasks in the financial domain. Developed by ProsusAI, FinBERT is a fine-tuned version of the widely known BERT model that is pretrained on a large corpus of financial text. This enables FinBERT to better understand the financial sector's unique language, terminology, and context, providing investment professionals with more accurate and relevant insights when analyzing financial news, earnings call transcripts, and analyst reports.

FinBERT is just one example of the many open-source tools on the Hugging Face platform. With an explosion of new models being added, the platform now boasts over 580,000+ models, each designed to perform a specific task. In addition to these models, Hugging Face also provides a way for open-source datasets to further support research and development efforts. These datasets can be used for multiple purposes, such as training, fine-tuning, and benchmarking models. Benchmarking datasets are used to standardize the evaluation of a model's performance by providing a common ground for assessment.

The open-source community has also cultivated a culture of collaboration and learning through competitions hosted on such platforms as Kaggle. Kaggle is an online platform that brings together data scientists, ML practitioners, and AI enthusiasts worldwide to solve complex problems and showcase their skills. These competitions tackle real-world challenges in various industries, including finance, health care, and retail.

Another highly useful and collaborative tool is Google Colab. This web-based platform offers a user-friendly environment for writing and executing Python code, emphasizing ML and data science applications. Google Colab provides free access to essential computing resources, such as graphics processing units (GPUs), which significantly reduce the time needed to train complex models.

Keeping Up

Staying up to date with ML and AI can be daunting but is achievable through a strategic approach and adept utilization of available resources. Engaging with online communities like Hugging Face and Reddit, subscribing to newsletters, and following industry behemoths like OpenAI, Google AI, Meta AI, and DeepMind are instrumental in staying informed. Additionally, for those who find technical research reports formidable, blogging platforms like Medium offer synthesized, sometimes lucid renditions of key advancements. While the online communities and blogs can be extremely helpful in staying informed on cutting-edge technology, users should exercise caution because these sources have few or no barriers to publication.

Alternative vs. Unstructured Data

The difference between alternative and unstructured data can be somewhat confusing because the terms are often used interchangeably to describe the same data. The critical distinction lies in understanding that "unstructured" describes the data form, which can be classified as structured, semistructured, or unstructured. In contrast, "alternative" distinguishes the data type. In this case, the distinction is between nontraditional information sources and traditional ones, such as financial statements, market data, and economic indicators. Next, we will outline the different levels of distinction of the data used in the investment management process to further clarify the various data types and forms.

Data Generators

The first level of distinction in defining the data used in investment decision-making processes is understanding the various generators of the data, which include companies, governments, individuals, and satellites and sensors.

Data generated by companies can be classified into two main categories:

- **Company data:** This type of data directly relates to the company's own functions and characteristics. It includes financial statements, operational metrics, strategic plans, and other data that describe the company's health, performance, and business operations.
- **Interaction data:** This type of data arises when individuals or entities interact with the company's products and services. Examples include credit card transactions, app download statistics, and email receipts. These data can be especially valuable because they reflect real-world user behaviors and trends and can often be sold or licensed to third parties for various purposes.

Government-generated data can also be classified into two main categories:

- **Economic data:** Analogous to company data, this type of data provides a snapshot of the health, performance, and status of a country's economy.
- **Interaction data:** Drawing a parallel to company interaction data, these data are generated from the day-to-day functions of government activities, including business permits, patents granted, and public service usage, such as transport ridership and facility utilization.

Individuals generate data by contributing through their online activities, such as social media engagement, consumer reviews, and search engine queries. Lastly, such technologies as satellites and sensors generate data in the form of geolocation information, satellite imagery, and internet of things (IoT) devices, like manufacturing equipment usage patterns.

Data Types

The second level of distinction is the type of data—that is, whether the data are traditional or nontraditional. Nontraditional data have been labeled as "alternative" data and are thereby defined as any data that differ from traditional investment sources, such as financials statements, market data, and economic indicators.

The ambiguity surrounding this definition often adds to the confusion about "alternative data" because what determines a classification of alternative versus traditional data often depends on the context of the data's source and use. For example, the classification of consumer sentiment depends on the data source. Consumer sentiment has traditionally been gauged using surveys, like the University of Michigan's Consumer Sentiment index. In this traditional source, consumer sentiment would likely be considered traditional data. In contrast, gauging consumer sentiment by classifying online forum posts into sentiment categories would likely be considered "alternative." An example of the distinction where the classification depends on the use is training an ML model to detect patterns in words or tone from an investor conference call that indicate an effect on performance. In this case, the source is the investor conference call, which is a traditional source, but the use is nontraditional and thus considered alternative. Regarding company- and government-generated data, the first two respective categories, company data and economic data, are most often considered traditional, whereas the second category, interaction data, is often considered alternative. Data generated by individuals and satellites and sensors are most often considered alternative.

Data Forms

The last level of distinction we will discuss is the data form. From the July 2023 CFA Institute survey on alternative and unstructured data, 79% say that less than half of the alternative data they use are unstructured. This may be due to alternative data being delivered in a structured form, such as vendor-provided sentiment scores and other transformed data. Unstructured data lack a specific format or organization, making them harder to analyze using traditional data processing tools. Examples of unstructured data include free-text social media posts, consumer reviews, satellite images, and raw sensor data from IoT devices. Unstructured data are characterized by their nontabular and nonrelational nature.

Structured data, in contrast, are well organized and easily searchable. This data form includes such alternative data as credit card transactions and app download data.

There is also the semistructured data form, such as email receipts and JSON (JavaScript Object Notation) files. These files have some level of organization but are not as rigidly structured as databases or spreadsheets.

Third-party data providers have traditionally offered solutions that deliver unstructured data in a clean and usable format, which has been particularly beneficial for organizations without in-house data science capabilities. However, this advantage can have some drawbacks when the data are not provided in raw form. When data are preprocessed or aggregated by third-party providers, there might be a risk of losing some of the granularity or context that could be valuable for specific use cases or analyses. Furthermore, this preprocessing might introduce unintentional biases or errors, which could impact the quality of the insights derived from the data. Therefore, organizations must carefully evaluate the trade-offs between the convenience of preprocessed data and the potential limitations that may come with such data to make informed decisions about their data-sourcing strategies.

Exhibit 3 breaks down the data types and the data structure in a 2x2 matrix using an earnings release as the data generating event to help conceptualize these data in a visual format.

Alternative Data Types

With the sheer growth of data has come growth in new alternative data types and new ways investors are able to extract value. **Exhibit 4** shows the popularity of the various alternative data sources indicated in the July 2023 CFA Institute survey on alternative and unstructured data. For a detailed breakdown of the various alternative data types and use case examples, see Appendix A.

Exhibit 3. Example Data Type and Structure Matrix: Earnings Release Event

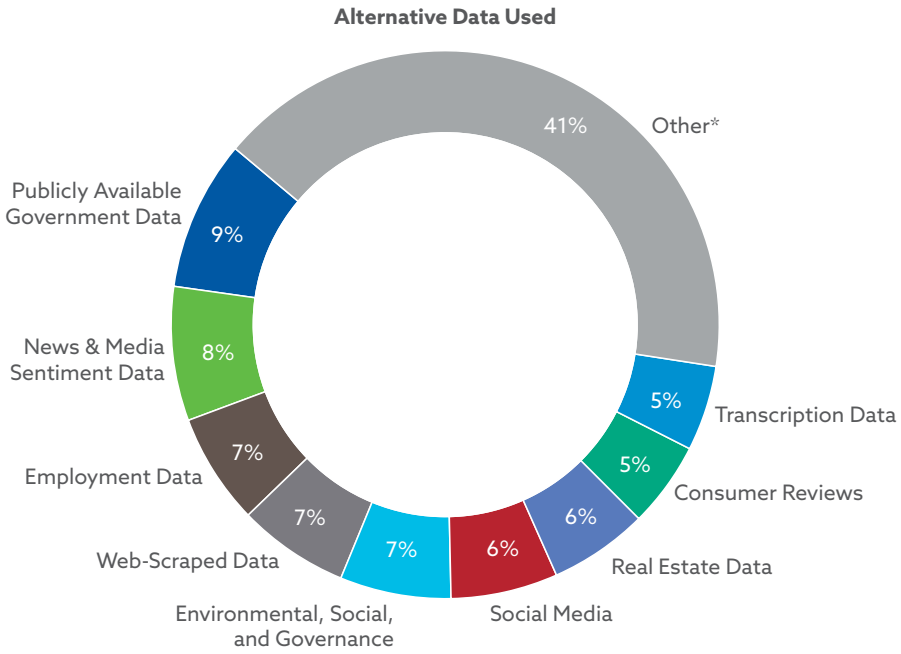
	Structured	Unstructured
Traditional	Tabular Financial Statements	PDF Financial Statement Conference Call Transcript: used to extract performance metrics or management guidance
Alternative	Vendor Sourced: Earnings Sentiment Score Vendor Sourced: Financial Statement Language Complexity Score[a]	Financial Statement Textual Analysis: using ML to detect YOY language consistency in MD&A[b] Conference Call Recording: using ML to detect tone of voice patterns related to earnings confidence[c]

[a]Language Complexity is a score used to determine the complexity of the language used in financial statements. A lower score means simpler language and is viewed favorably (Patel 2023).

[b]Companies with high textual similarity in financial statements year-over-year have shown some evidence of outperforming companies with low textual similarity (Zhao 2021).

[c]Companies that exhibit a less optimistic tone of voice in conference calls have shown to have higher stock price crash risk the following year (Fu, Wu, and Zhang 2021).

Exhibit 4. Alternative Data Used (survey responses)



Note: 254 responses.

*In descending order, Other (<5%) includes: Energy Consumption Data, E-commerce Data, Supply Chain & Logistic Data, Credit Card Transactions, Weather Data, App Download Data, Court Records and Legal Documents, Satellite Imagery, Insider Trading Data, Patent and Intellectual Property Data, Crypto Data, Geolocation Data from Mobile Foot Traffic, Data from IoT Devices, Flight Tracking Data, Clickstream Data, Sensor Technologies, and Wearables Data.

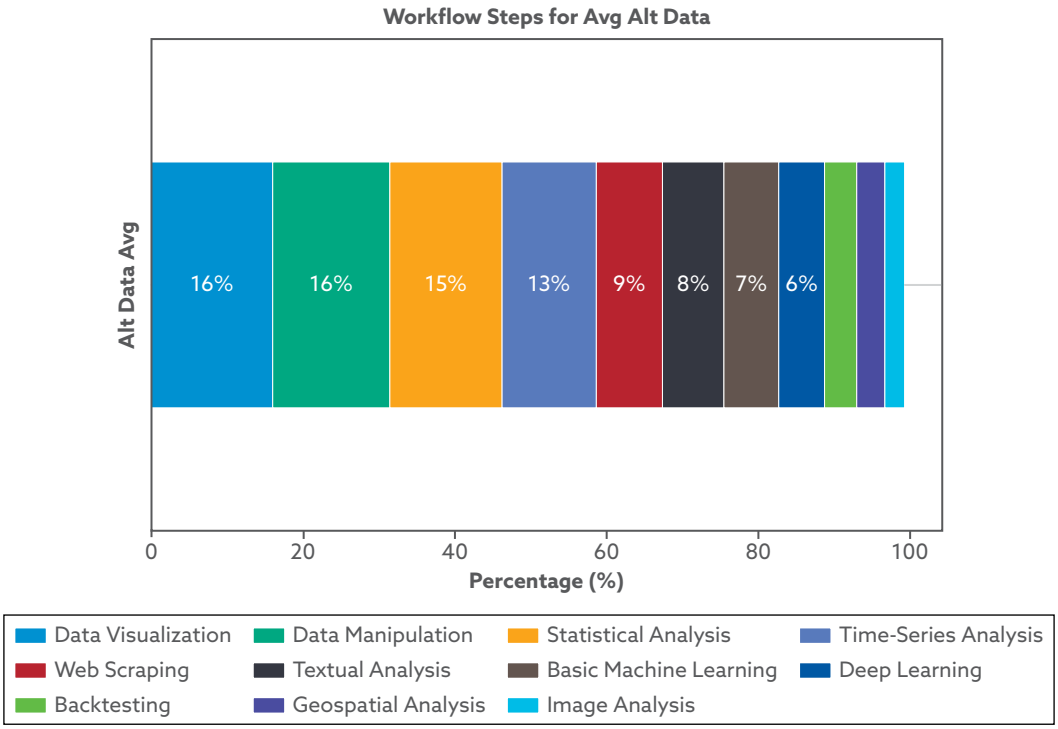
Source: From the July 2023 CFA Institute survey on alternative and unstructured data. See footnote 1 for more details.

Open-Source Alternative Data

Investment professionals' ability to extract valuable insights from unstructured data have greatly improved with advances in NLP and computer vision techniques. And with the proliferation of the open-source community, even some of the most advanced tools are now freely available. These widely available tools have made it easier to use unstructured data and find value in open-source alternative data sources. For investment firms, having in-house capabilities in parsing unstructured data will become increasingly important as the barriers to using these tools continue to decline.

According to the July 2023 CFA Institute survey, 48% of the unstructured data investment professionals deal with now come from open-source data sources. **Exhibit 5** breaks down the workflow steps that are used on average when dealing with alternative data, as indicated from the survey data. For a detailed breakout of the workflow steps for various alternative data types, see Appendix B.

Exhibit 5. Workflow Steps for the Average Alternative Data Type (survey responses)



Note: 171 responses.

Source: From the July 2023 CFA Institute survey on alternative and unstructured data. See footnote 1 for more details.

Getting Comfortable Building

There is immense value in being able to rapidly iterate new ideas. Starting on the journey to building models is by no means a replacement for the essential collaborative efforts and expertise of various departments within an organization. In fact, it is likely to lead to a growth in collaboration across departments as more ideas can be tested quickly, each one requiring proper checks and sign-off from engineering, compliance, and legal. With this in mind, it is essential that firms and professionals become comfortable in the building process whereby they can efficiently and effectively explore new concepts while maintaining the highest ethical and legal standards.

To assist professionals in traversing the complex ethical and legal dimensions, CFA Institute has created an ethical decision framework for AI (Preece 2022). This resource equips investment professionals with the means to approach projects with greater confidence and clarity surrounding the ethical dimensions of projects. It consists of a set of key questions and prompts that encourage users to critically evaluate various aspects of their data-driven tasks, such as data collection, storage, processing, and sharing. **Exhibit 6** summarizes the ethical considerations and key questions professionals should evaluate along each step of the workflow.

Exhibit 6. Ethical Considerations for AI Project Development

		Ethical Considerations			
		Data Integrity	**Accuracy**	**Transparency and Interpretability**	**Accountability**
Workflow Step	**Obtain input data**	What is the source of the data? What sampling methods are used, and how are data cleansed? Are data labels fair and accurate (if using supervised ML)? Is the dataset representative? How are potential biases accounted for or corrected? Do data sourcing initiatives respect data privacy laws? Is the confidentiality of client data protected? Do the input data contain any potentially material, non-public information?	Check the validity and veracity of the data. Are the data relevant to the problem specified? Do the data permit fair and accurate inferences?	Are descriptions of the input data retained? How are data described and referenced in the investment process or in reporting to clients and to supervisors?	How are data sourcing initiatives governed? How are input data stored, and are they securely maintained? Are roles and responsibilities clear?
	Build, train, and evaluate model	Is there sufficient sampling history to effectively train the model? Does the sample contain biases that may cause the model to inappropriately weight certain features or groups?	Does the model perform as intended? Will the model deliver accurate and suitable outcomes for the client? Does the desired level of accuracy come at the cost of excessive model complexity? Refine and iterate model parameters as appropriate.	Are the model features and their contribution to the outcome interpretable? Can the model features be adequately communicated to clients and supervisors?	Is there a robust evaluation and approval process (such as via a committee) before models enter a live environment? How are potential conflicts of interest evaluated? How are potential adverse client outcomes or potential market distortions addressed?

(*continued*)

Exhibit 6. Ethical Considerations for AI Project Development (*continued*)

		Ethical Considerations			
		Data Integrity	**Accuracy**	**Transparency and Interpretability**	**Accountability**
	Deploy model and monitor	Conduct periodic reviews of the input data to monitor for the emergence of biases. Does the dataset remain sufficiently representative?	Does the model deliver good out-of-sample performance, with results that are accurate, robust, and generalizable? Conduct regular testing and reviews to understand if there are any changes to model performance over time.	Does the process by which the AI tool learns from the data evolve over time? Does the contribution of features to the outcome change over time? If so, how are such issues explained and communicated to clients?	Conduct periodic testing to ensure the model stays true to the client mandate, and check for style drift where appropriate. Where models deviate from their original parameters, what controls are in place to negate adverse client outcomes? Is model performance disclosed appropriately in client reporting?

Source: Preece 2022.

Web Scraping

One of the most common areas of ethical ambiguity when building a new project is the practice of web scraping. Web scraping involves extracting data from websites and can be a valuable tool for collecting information otherwise unavailable through traditional application programming interfaces (APIs) or data feeds. However, this practice raises several ethical and legal issues that developers should be mindful of.

The Investment Data Standards Organization has created a comprehensive web crawling best practices guide (IDSO 2019), aiming to provide developers and investment professionals with a thorough understanding of the ethical and legal implications surrounding web-scraping activities. IDSO notes that web harvesting can be considered low risk as long as noncopyrighted content is extracted, website access does not negatively impact website usage, and the content is used for internal research and development purposes. Bearing this in mind, the guide outlines several areas to focus on to maintain a low legal risk profile while extracting data from websites.

First, web scraping can infringe on website owners' rights by accessing and using their content without permission, which could violate copyright, trademark, or database rights and lead to legal repercussions. To avoid these issues, it is crucial for developers to check a website's terms of service or robots.txt file, which often contains guidelines on what types of data scraping are allowed or prohibited.

Second, web scraping may place an undue burden on a website's server, causing it to slow down or crash and thus negatively affecting its performance and user experience. To mitigate this risk, developers should implement responsible scraping techniques, such as limiting the rate and frequency of requests, scraping during off-peak hours when server load is typically lower, and using the API if one is available.

Third, data privacy is a critical concern with web scraping because it is imperative to avoid collecting sensitive or personally identifiable information. Developers should make sure that they are scraping public information and adhering to applicable data protection regulations, such as the General Data Protection Regulation in the European Union or the California Consumer Privacy Act in the United States. This may require obtaining explicit user consent, anonymizing the collected data, or implementing secure data storage and handling practices.

Fourth, developers should also be aware of the potential for data inaccuracies and biases in the information obtained through web scraping. Websites may contain outdated, incorrect, or misleading information, which could lead to flawed analyses or decision making. It is essential to verify and validate the accuracy of the scraped data through cross-referencing with other sources or using rigorous data cleaning and preprocessing techniques.

Lastly, one area likely to carry meaningful legal risk regarding web scraping is engaging in activity that could be considered in direct competition with the host website. Developers must be aware of this potential issue because it may lead to legal disputes or liability for damages incurred by the host.

For many years, the ambiguity surrounding web scraping made it difficult for legal cases against web scrapers to hold up. The lack of clear legal guidelines has resulted in much uncertainty for web-scraping activities. In a 2019 case in the United States, *hiQ Labs v. LinkedIn 938 F.3d 985*, a court ruled that one cannot be criminally liable for scraping publicly available data under the Computer Fraud and Abuse Act (CFAA; see Růžičková 2022).

In that case, hiQ Labs, a data analytics company, scraped publicly available LinkedIn profiles to generate insights for its clients. LinkedIn objected to this practice and sent hiQ Labs a cease-and-desist letter, claiming that its web-scraping activities violated LinkedIn's terms of service and the CFAA. In response, hiQ Labs sued LinkedIn, arguing that its web-scraping activities were legal since the data it collected were publicly available. Ultimately, the

court ruled that hiQ Labs cannot be criminally liable under CFAA but did violate contractual obligations under LinkedIn's terms of service. This case highlights the importance of understanding and abiding by contractual agreements when scraping, especially when dealing with website terms and conditions. Web scrapers should exercise caution when accessing data behind a login because that data may be subject to additional restrictions.

Licensing

Another area of legal importance when building open-source projects is understanding the various licenses on which open-source projects are released. These licenses govern the terms under which software can be used, modified, and redistributed. They play a crucial role in defining the level of freedom and control developers and users have over the software. The three most commonly used licenses are the MIT License, Apache License 2.0, and General Public License (GPL). Each license has unique characteristics. For example, the MIT License is the most permissive, allowing for free use, modification, and distribution of the software, both for commercial and noncommercial purposes, with the only requirement being the inclusion of the original copyright notice and license text in all copies or substantial portions of the software.

The Apache License 2.0, while still permissive, offers some additional terms compared to the MIT License. It provides an express grant of patent rights from contributors to users, protecting them from potential patent infringement claims. This creates a safer environment for developers and users while maintaining the freedom to modify and distribute the software.

In contrast, the GPL is more restrictive. It requires that any changes or modifications made to the software be released under the same General Public License, ensuring that derivative works remain open source. This concept, known as "copyleft," is intended to promote knowledge sharing and prevent the privatization of open-source software.

These three licenses represent the majority of the licenses that professionals will encounter when building open source; however, this list is not exhaustive. Developers should research and understand the various licenses available before incorporating any open-source software into their projects.

Conclusion

The democratization of access to cutting-edge tools and resources, the explosion of unstructured data, and the open-source community's collaborative spirit have created a fertile environment for innovation and discovery. As the data depth and breadth continue to grow and new technologies emerge, professionals will need to embrace a more holistic and scientific approach to investing to stay ahead. The possibilities offered by this new era of data invite a more creative mentality to unlock the potential of the data that surround us.

2. FINE-TUNING LARGE LANGUAGE MODELS

As the use of unstructured and alternative data has grown increasingly important, the rise in use of deep learning algorithms to extract value from such data has grown in step. NLP, a form of deep learning, has been particularly useful in finance as so much of the perceived value in investing comes from interpreting textual data. The advancements in NLP have created a new generation of powerful language models. GPT-4 and similar LLMs embody the cutting edge of this technology, demonstrating impressive abilities in a range of tasks. Despite their robust capabilities, however, these models may need to be adapted or "fine-tuned" to achieve optimal performance in certain cases. The process of fine-tuning, while immensely valuable, is not without complexities and costs. This chapter explores the intricacies of fine-tuning LLMs, outlining various methods, the role of contemporary libraries, and the challenges and opportunities that lie ahead in this exciting field. The purpose is to equip practitioners with an understanding of modern methods to work with unstructured data.

The Evolution of NLP

In this section, we track the progression of NLP from its early history to the development of transformers and finance specific models, such as BloombergGPT.

Early History

NLP encompasses the automated comprehension, interpretation, and generation of human language. In its infancy, NLP was underpinned by elementary algorithms that used an array of manually crafted linguistic rules. These rules aimed to dissect text and draw basic conclusions based on predetermined grammar and syntactic frameworks. However, this approach proved to be arduous and was impotent in grappling with the multifaceted and ambiguous nature of human language.

Around the same time that quantitative analysis was emerging, researchers started using statistical methods to learn the patterns from large amounts of text data. Such models as hidden Markov models and, later, Naive Bayes classifiers became popular in the field. These models use the probability of observing certain words given previous words or sequences to make predictions. This shift allowed for the creation of more nuanced and robust models that could better handle the intricacies of language. However, these models made strong assumptions that often did not hold true in real-world data and lacked the ability to capture long-range dependencies.

To address these limitations, researchers turned to neural networks, which provided a more flexible and powerful framework for learning from data.

© 2024 CFA Institute. All rights reserved.

The advent of recurrent neural networks (RNNs) marked a significant advance in the field. RNNs introduced a unique feature that was absent in conventional neural networks: a form of memory. This memory was accomplished by adding loops in the network architecture, allowing information to be passed from one step in the sequence to the next. This feature made RNNs particularly suited for tasks involving sequential data, such as time-series analysis and, most notably, language processing. RNNs, with their ability to maintain a form of memory, were better equipped to deal with sequential data compared to their statistical predecessors.

Limitations in RNNs' capabilities became apparent, particularly when dealing with long sequences, largely due to what is known as the "vanishing gradient" problem. During the training process, RNNs use the backpropagation algorithm, which adjusts the weights in the network based on the calculated gradient. This gradient measures how much we need to change the learnable parameters in the model for a given reduction in the loss function, which is a measure of the error between the predicted and actual values. However, when backpropagating through many time steps, the gradient from early time steps can become exceedingly small, approaching zero. When this happens, the updates to the weights become very small, making the network unable to effectively learn from the early parts of long sequences.

To address this problem, researchers developed an advanced RNN architecture called long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997). This model includes mechanisms that allow the network to maintain the gradient over longer sequences, enabling them to better capture long-term dependencies. Still, the challenge of accurately capturing the complexity of human language remained because these models often struggled to fully grasp the surrounding context of the words and sentences they were processing. By their nature, they were unidirectional, processing text from left to right. This meant that the models had limited capacity to consider the future context; that is, they could not leverage the information from the words appearing later in a sentence while processing the current word.

For example, in the sentence, "The bank consists of the sides of the channel, between which the flow is confined," the model would take the first two words and likely associate the sentence with a financial institution. To mitigate this problem, researchers proposed bidirectional RNN (BRNN) models (Schuster and Paliwal 1997). These models operate on the input sequence in both directions, enabling them to capture both past and future context when processing a given point in the sequence. This increased understanding of context made BRNNs more effective at tasks that require nuanced comprehension of sentence structure and meaning.

Following these advancements, a new wave of techniques focusing on vector representation of words started to gain traction. This development marked another significant leap forward as these techniques, most notably Word2Vec (Mikolov, Sutskever, Chen, Corrado, and Dean 2013) and GloVe (Pennington,

Socher, and Manning 2014), changed the way we represent words in a computable format.

Word2Vec and GloVe, developed by teams at Google and Stanford University, respectively, gave rise to the concept of "word embeddings." In this paradigm, words were no longer viewed as discrete, isolated entities but were instead represented as vectors in a high-dimensional space. This meant that semantically similar words would be mapped to proximate points in this vector space, capturing nuances of language that were previously difficult to quantify. This vector representation allowed neural networks to understand words in terms of their context and semantic relationships with other words, overcoming the limitation of traditional methods that treated words as isolated symbols.

These models still had drawbacks, however. Because such techniques as Word2Ve cand GloVe offered a single-vector representation for each word, they were unable to accommodate the fact that many words carry multiple meanings, each dependent on context.

Recognizing these limitations, researchers started exploring models that could not only capture context more effectively but also process sequences more efficiently. The drive for more effective context understanding, combined with the power of word embeddings, paved the way for the emergence of the next influential development in NLP—the transformer architecture (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin 2017).

Transformers

The transformer architecture was a significant departure from the prevailing recurrent architectures. It introduced a new approach to handling sequential data using a mechanism known as "attention" or "self-attention," a concept that had been part of earlier models but was never used as the main architectural component. The crux of the attention concept is that it allows the model to focus on different parts of the input sequence when producing an output, giving the model the capacity to weigh the importance of input components differently. The use of attention introduced the capacity to simultaneously process entire sequences rather than sequentially processing input data, as in RNNs or LSTMs. This feature makes them highly parallelizable and, therefore, faster and more efficient to train.

In addition to attention, transformers also took inspiration from the concept of embeddings, as popularized by Word2Vec, albeit with a significant variation. Rather than having a fixed representation for each word regardless of its context, as in Word2Vec, transformers introduced the concept of context-dependent embeddings. This means that the same word can have different representations based on its surrounding words, providing a more nuanced understanding of language.

These advancements in the attention mechanism and context-sensitive embeddings set the stage for the era of large language models. By processing

information in parallel and adapting to the contextual nuances of language, transformers were able to scale up, handle vast amounts of text, and learn more effectively from this text. Given the importance of the attention mechanism for recent advances in large language models, a deep dive covering the concept is provided in Appendix C for interested readers.

The original transformer model was designed with the specific objective of translating text. This task was accomplished using a novel architecture consisting of two main components: an encoder and a decoder. The encoder was responsible for learning patterns from the words and converting them into vector space, effectively capturing the contextual and semantic information of the input text. The decoder then took these encoded vectors and transformed them back into readable text in the target language, leveraging the learned patterns to generate accurate translations. However, not long after the introduction of the original transformer model, researchers began to exploit this versatile infrastructure to achieve a much wider array of tasks. This led to the development of a model that marked another pivotal advancement in NLP, BERT (bidirectional encoder representations from transformers; see Devlin, Chang, Lee, and Toutanova 2019).

BERT

BERT exploited the transformer architecture in a new and creative way. Instead of using both the encoder and decoder, BERT leveraged only the encoder part of the transformer model. It introduced a new pretraining objective known as masked language modeling (MLM), which allowed it to learn by predicting masked tokens, namely a word or subword. A subword is just a piece of word broken into parts. For example, the word "jumped" could be split into two subwords as "jump" and "ed"—reflecting two tokens. In MLM, a portion of the tokens in the sentence have been randomly replaced with a [MASK] token. In this case, a training example sequence might look like:

["the", "cat", "jump", "ed", "[MASK]", "the", "l", "edge"]

And the model will attempt to predict "on" as the masked token.

This approach enables BERT to understand the context of a word in relation to all the other words in a sentence, irrespective of their order, thereby truly capturing the bidirectional nature of language. The new learning objective was a significant leap forward because it took the concept of unsupervised learning, traditionally used in ML to identify patterns in input data without explicit labels, and adapted it for the complexities of human language. This advancement made it possible to train a model on massive quantities of textual data, leading to the creation of powerful models that are capable of understanding the complexities of human-generated text.

These large language models became known as foundation models due to their ability to serve as a basis for numerous downstream tasks. Once the models

have been pretrained, the model can be fine-tuned on a smaller, task-specific dataset, leveraging the rich, general-purpose language understanding the model has already acquired. This process of transferring the capabilities of a foundation model to specific tasks is known as transfer learning.

GPT

With the success of the transformer architecture, NLP shifted toward larger and more powerful models. The MLM strategy in BERT, while effective, had its limitations, particularly in generating coherent and fluent text. BERT's ability to take in an entire passage of text to understand the context made it particularly suited for such tasks as sentiment analysis. However, BERT is not inherently a generative model; it does not predict the next token in a sequence but rather fills in the blanks. This is where the generative pretrained transformer (GPT) model (Radford, Narasimhan, Salimans, and Sutskever 2018), developed by OpenAI, excelled. Unlike BERT, which uses the MLM strategy, GPT uses a variant of the transformer model that solely consists of the decoder component. GPT's pretraining objective, known as causal language modeling (CausalLM), is designed to predict the next token in a sequence based only on the preceding tokens. Although unidirectional, GPT models made a major jump over BERT models, partly due to the computational efficiency in the pretraining task. BERT models' pretraining task is to predict the masked token given all the tokens around them, which are a small fraction of the total tokens in the sentence. GPT models, however, predict all tokens given all the previous tokens in the sequence. Thus, while BERT may be able to take in the context of the entire sentence before predicting the masked token, it still lags a GPT model's sheer quantity in training examples. This subtly different approach to pretraining, combined with the scale of the model and data it was trained on, led to significant breakthroughs in the generation of text that closely mimics human language.

The emergence of GPT demonstrated that the performance of these models was dependent not only on the quantity and quality of data used for training but also on the scale of the models themselves in terms of the number of learnable parameters. In other words, more data and more parameters often led to better results.

This understanding led OpenAI and other research organizations to push the boundaries of what was previously thought feasible in terms of model size and training datasets. As OpenAI scaled up its models with GPT-2 (1.5 billion parameters) and GPT-3 (175 billion parameters), it began to see remarkable improvements in the quality of the generated text. These models were not only producing more coherent and contextually relevant output, but they also began to demonstrate an understanding of nuanced linguistic and cultural concepts that were previously thought to be the domain of humans.

Moreover, these larger models, due to their ability to capture more complex patterns and dependencies in the data, were found to be even more effective

when prompted for specific tasks. This phenomenon is often referred to as "few-shot learning" or "in-context learning," where the model, after being pretrained on a massive corpus of text, can adapt to a specific task with just a few examples and no parameter fine-tuning.

With these emergent capabilities and in anticipation of its release of GPT-4, OpenAI took the technology that was predominantly being used by researchers and developers and democratized it through the release of ChatGPT (OpenAI 2022).

ChatGPT

While the underlying GPT models were highly proficient at understanding and generating text, they were still missing a crucial piece of the puzzle: alignment with human preferences. To enhance this alignment, OpenAI used a two-step process of supervised fine-tuning and reinforcement learning with human feedback (RLHF) to build ChatGPT.

Supervised fine-tuning involved training the model to emulate human responses, using datasets where AI trainers played both the user and the AI assistant roles. This process introduced the model to a wide array of conversational scenarios and responses, forming a more human-like interaction style. In the second step, OpenAI incorporated RLHF. This technique involved the model generating multiple responses and AI trainers ranking these responses. The model then uses this feedback to learn and adjust its responses for similar prompts in the future, thereby improving the quality of its responses over time.

The combined technique increased alignment, effectively bridging the gap between the capabilities of the raw GPT models and the intuitive, human-like responses expected by everyday users.

LLaMA

In the wake of ChatGPT's debut, a team of researchers at Meta introduced a new model, LLaMA (Touvron, Lavril, Izacard, Martinet, Lachaux, Lacroix, Rozière, Goyal, Hambro, Azhar, et al. 2023). LLaMA (large language model Meta AI), which was originally available only for academic use, was a collection of foundation models ranging from 7 billion to 65 billion parameters. With the release of Llama 2, the models now extend to 70 billion parameters and are available for commercial use. LLaMA was significant for several reasons. It was the first open-source model that was trained on trillions of tokens and small enough to be able to be run on consumer-grade hardware. It also demonstrated that it is feasible to create state-of-the-art models using solely publicly available datasets, avoiding the reliance on proprietary and inaccessible datasets. But its most notable accomplishment was that the 13 billion parameter model surpassed the performance of GPT-3, which contained 175 billion parameters, on most benchmarks. This reignited the debate on whether more parameters

are always better and shifted the debate toward getting the best performance for any given budget.

Open-Source LLMs

Within weeks of the release of LLaMA, researchers at Stanford University saw an opportunity and assembled a dataset of 52,000 instruction-following demonstrations using text-davinci-003, the model underlying ChatGPT at the time. Armed with this curated dataset, they fine-tuned the 7 billion parameter LLaMA model (LLaMA 7B), an endeavor that remarkably required only $100 and three hours of training time.

The result was a new model called Alpaca (Taori, Gulrajani, Zhang, Dubois, Li, Guestrin, Liang, and Hashimoto 2023). Despite its relatively modest size and training budget, Alpaca demonstrated strong performance in instruction-following tasks. This achievement was significant for several reasons. First, it represented a cost-effective and replicable model with performance comparable to much larger, proprietary models. Second, it opened up the door to new advancements in fine-tuning open-source LLMs for downstream tasks. Fine-tuning LLMs requires a massive amount of computational resources—a prohibitive cost for many in the research community. However, with smaller models, such as LLaMA 7B, researchers could explore innovative techniques in fine-tuning, which has led to an explosion in open-source LLMs that are competitive with proprietary models.

Where Is the Moat?

The implications of the advances in the open-source community have left even the most powerful players in AI wondering where to find differential value, as noted in a leaked letter from a Google engineer (Patel and Ahmad 2023). The investment industry is, by its very nature, a confidential arena. Financial institutions, investment firms, and individual traders often hold close their trading strategies, market analyses, and proprietary algorithms. This guarded approach extends to data, where exclusive data sources and proprietary databases can provide an edge. Customizing models on proprietary data and industry-specific expertise seemed to be a way forward, as evident from the introduction of BloombergGPT (Wu, Irsoy, Lu, Dabravolski, Dredze, Gehrmann, Kambadur, Rosenberg, and Mann 2023).

Bloomberg leveraged its vast repository of exclusive financial data and industry know-how to train an advanced language model called BloombergGPT. This model is trained on 363 billion tokens from Bloomberg's proprietary data, including financial reports, market analyses, and news articles, and is augmented with 345 billion tokens of general-purpose datasets. The result is an AI tool that is deeply integrated into the financial domain and illustrates how institutions with exclusive data and sector-specific expertise can create custom AI solutions tailored to their industry's unique demands.

In this case, Bloomberg did not fine-tune a foundation model with their data but actually built the model from scratch. Still, the precedent it set on training with proprietary data had broad implications that extend beyond large institutions with vast datasets. The potential of customizing language models can also be harnessed by smaller teams and organizations with unique ideas, expertise, and datasets. These entities can use fine-tuning processes to create bespoke AI solutions that provide differential value within their particular domain. Next, we discuss these fine-tuning processes in detail.

Fine-Tuning

In this section, we provide a detailed discussion of fine-tuning, including when to use it, which methods are available, and what to consider when building a dataset.

Getting Started

The first question for professionals to address when working with unstructured or alternative data and contemplating fine-tuning is whether it is truly an indispensable step. Fine-tuning, while powerful, is not always the optimal approach. It involves several intricate steps, each requiring effort, time, and resources, posing a significant challenge. A helpful question to ask is how far the simplest solution goes. For example, if we were to try to identify all the ESG-related tweets by companies in an index, we might contemplate fine-tuning a model to identify ESG-related tweets. However, the simplest solution of coming up with a list of ESG-related key words, such as #CorporateSocialResponsibility or #Sustainability, could get us a large percentage of the way there.

When a simple keyword search will not suffice, the next most simple solution is usually zero-shot learning. Zero-shot learning takes advantage of the text embeddings in transformer models by leveraging the similarity between two vectors. When a sentence is transformed into a vector representation, we can use basic trigonometry to calculate a cosine similarity between the two vectors. The model then uses the category vector with the highest cosine similarity as its label. **Exhibit 7** demonstrates a basic Python implementation of zero-shot classification of tweets into ESG or non-ESG related.

The next simple solution is to find an already publicly available fine-tuned model on the Hugging Face hub that will suffice for the problem at hand. As of this writing, there are 168 models on the hub that are ESG related, many designed to categorize text into various ESG issues.

Sentiment analysis is another widely used application among investment professionals, and a plethora of models are available on Hugging Face for this task. Applying one of these models to study behavior patterns in different groups of investors using Reddit data (see, e.g., Pisaneschi 2023) is an example of taking a familiar tool and using it for a different purpose.

Exhibit 7. ESG vs. Non-ESG Tweets: Zero-Shot Python Example

```
from transformers import pipeline
classes = ['ESG Related','Non-ESG Related']

# Initialize zero-shot classification pipeline
classifier = pipeline("zero-shot-classification")

# Specify the text that you want to classify
text =list(tweets_samples.Tweet)

# Specify the labels (classes) you are interested in
labels = classes

# Perform zero-shot classification
result = classifier(text, labels)

# Gather labels from classifier
results = [elem['labels'][0] for elem in result]

# Create Zero Shot Label Column
tweets_samples['Zero Shot Label'] = results
```

These simple steps are ways for developers to rapidly iterate an idea to glean some insights before a major investment is made in a particular problem. Weighing the cost and complexity of the various fine-tuning methods against simpler solutions is ultimately the first place to start.

Few-Shot Learning

With the advent of ChatGPT, the cost of running API inference on extremely powerful models has dropped dramatically. As the competition ramps up, further price drops and more powerful models in this price range are likely. Given these models have shown strong capabilities in being few-shot learners (Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell, et al. 2020) and with their low cost, few-shot learning is a logical next step. As a caveat, this method is technically not a fine-tuning method because no model weights are being changed. Instead, this method is known as in-context learning because it involves giving the model training examples inside the prompt, which increases the probability of the correct next words given the previous words (our training examples). This method heavily relies on the user's ability to develop a well-thought-out prompt tailored to the model's capabilities. This is where prompt engineering becomes extremely important, not only in getting the right answer but also in avoiding added steps for cleaning the output. Professionals can use the following list when building a prompt.[2]

[2]For more comprehensive guidelines, see the "Prompt Engineering" guide on the OpenAI platform: https://platform.openai.com/docs/guides/prompt-engineering.

- **Step by step:** By their autoregressive nature, these models perform better when they have time to "think" through their answer step by step, saving their final answer for the end of their response. That is, they perform better with chain-of-thought reasoning (Wei, Wang, Schuurmans, Bosma, Ichter, Xia, Chi, Le, and Zhou 2023). Prompting the model to think through its answer step by step has been shown to increase its accuracy dramatically.
- **Demonstrate output:** Providing the model with a specific format for the output will save you time processing its responses later. For example, give it clear instructions on how to format its output, such as, "I want you to structure your output in JSON format, as {"thoughts": "your thoughts", "label": "your label"}." This will allow you to easily search for the label in its response.
- **Clear separations:** Using delimiters to define sections can help the model understand your intention better. In the previous example, we used a JSON format to separate the sections for easy post inference searching, but ### or """ can be used for defining sections. For example, "how would you label the following tweet. Tweet: ### {example tweet}###".
- **More examples:** Giving the model several examples of its intended output is synonymous with providing the model labeled data. Usually, the more examples we can provide, the better its output. Of course, this improvement comes with the added cost of tokens.

Exhibit 8 provides an illustration of a Python implementation of a classification task for GPT-4 using the OpenAI API. This task is a multiclass classification task where the model is asked to classify ESG-related tweets into four categories. This task is used in the case study discussed in chapter 3.

Two additional considerations to keep in mind when implementing this process are rate limits[3] and model overload. OpenAI has rate limits incorporated into its API, so any requests over the limits will produce an error in your code. Additionally, due to the extreme popularity of the ChatGPT app at the time of this writing, the model gpt-3.5-turbo, which underlies the app, is constantly at capacity and will error out randomly for this reason. It is best to include error handling in your code to avoid costly re-runs when looping over multiple examples.

Traditional Fine-Tuning Methods

For many traditional NLP tasks—such as named entity recognition, binary and multiclass classification, and question answering—175 billion parameter generative models, such as ChatGPT, may be disproportionate. A professional that has already gone through the process of curating a labeled dataset may be able to use a much smaller model to get even better performance than with ChatGPT. In fact, on tasks that require a classification given an entire sequence, such as sentiment analysis, smaller bidirectional models like BERT still show strong performance once fine-tuned for the task.

[3]Rate limits are set in place to reduce the overall requests received on the server so as not to overload the platform or slow down request processing for other users.

Exhibit 8. Python GPT-4 API Call Example: ESG Classification

```
# Importing necessary libraries
import pandas as pd
import openai
import json

# Setting the OpenAI API key
openai.api_key = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'

# Function to generate labels for ESG-related tweets using OpenAI's GPT-4 model
def gen_label(tweet):
  # Creating a chat completion with GPT-4, providing context and examples for the classification task
  response = openai.ChatCompletion.create(
      model="gpt-4", # Specifying the model to use
      messages=[
        # System message providing context and definitions for the classification task
        {"role":"system", "content":"You are a multi-class classification agent trained to classify ESG related tweets in the following categories:\
        ['Not-material','Community Outreach','ESG Industry Recognition','Actions and Innovations']. The categories are defined as follows:\
        {'Not-material': this category is for tweets that are not considered actions or likely to materially affect a company's fundamentals., \
        'Community Outreach: this category is for tweets that are actions related to ESG but are unlikely to materially affect the company's fundamentals., \
        'ESG Industry Recognition': this category is for when the company is tweeting about some recognition they are getting for being ESG aligned, \
        'Actions and Innovations': this category is for tweets where the company is showing real actions that they have done related to ESG that could \
        materially affect a company's fundamentals. The goal with this task is to discern a company's level of greenwashing and to bucket their tweets into \
        these categories that carry different levels of importance to investors. Make sure you think through your label step by step and be mindful of \
        quotation marks so that your output is in json format"},
        # Example messages between user and assistant to guide the model's response
        {"role": "user","content":"We're proud to be named to the #DJSI for the 10th consecutive year! 'We continue to integrate \
        #Sustainability into our operations, stretching across the way we grow, source, produce, and share our food. @CampbellCSR #sustainabilityatwork"},
        {"role":"assistant","content":'{"thoughts": "this tweet is speaking about being recognized for their sustainable accomplishments by industry peers", \
        "label":"ESG Industry Recognition"}'},
        {"role":"user","content":"we're thrilled to announce our partnership with teco 2030 to develop #carboncapture solutions for the maritime industry. \
        our goal is to help shipowners make their vessels more environmentally friendly, helping them to exceed imo standards. read more here:"},
        {"role":"assistant", "content":'{"thoughts":"this tweets is referring to specific actions that are occurring within the company regarding carbon \
        sequestration activities. A new partnership is likely to impact the company\'s fundamentals","label":"Actions and Innovations"}'},
        {"role":"user","content":"our elmore, oh team rolled up their sleeves and volunteered to clean-up the local river in their community. the team worked \
        together to collect 500 lbs. of waste out of the portage river. thank you to the team for making a difference in their community! #ecofriendly #gogreen"},
        {"role":"assistant","content":'{"thoughts":"this tweet is regarding actions the company is taking to clean up the local river. This is not likely to affect \
        a company\'s fundamentals","label":"Community Outreach"}'},
        {"role": "user", "content": f"{tweet}"}]) # Including the tweet to be classified
  return response['choices'][0]['message']['content'] # Extracting and returning the model's response

# Reading the ESG-related tweets from a CSV file
esg_tweets = pd.read_csv('path_to_esg_tweets.csv', index_col=0)

# Generating labels for each tweet using the gen_label function
responses = [gen_label(i) for i in esg_tweets['text']]

# Extracting the 'label' field from the JSON response and adding it as a new column to the DataFrame
esg_tweets['labels'] = [json.loads(i)['label'] for i in responses]
```

The Transformers Python library at Hugging Face offers a robust set of tools that facilitate the process of fine-tuning for downstream tasks. The library abstracts away a lot of the complexities of working with deep-learning models, providing a more user-friendly experience that simplifies model training, evaluation, and usage. Still, some complexities remain and can be difficult for beginners to navigate. Another library built on top of Transformers called Simple Transformers takes the abstraction a step further, allowing a user to fine-tune a model with only a few lines of code. **Exhibit 9** provides an example of fine-tuning a BERT model for multiclass classification using Simple Transformers and a set of 3,223 human-labeled tweets corresponding to the labeling task in the case study in chapter 3.

In this implementation, all the learnable parameters in the model are trained, which can be computationally heavy given the model size. For example, the original BERT model has 110 million trainable parameters. A professional can choose a smaller model, DistilBERT, which has 66 million parameters and maintains 95% of BERT's performance. Another option is to fine-tune a model that has been pretrained in the financial domain, such as FinBERT (Araci 2019). The Hugging Face platform is loaded with pretrained and fine-tuned models.

Exhibit 9. Traditional Fine-Tuning: Multiclass ESG Materiality Classification

```python
# Importing necessary libraries
from sklearn.model_selection import train_test_split
from simpletransformers.classification import ClassificationModel
from sklearn.utils.class_weight import compute_class_weight
import numpy as np

# Splitting the dataset into training and test sets (80% training, 20% testing)
train, test = train_test_split(esg_tweets, test_size=.2, random_state=42)

# Defining the path for the output directory where the model will be saved
path = f'{root}Model/Output'

# Computing class weights to handle class imbalance in the training dataset
# 'balanced' mode uses the values of y to automatically adjust weights inversely proportional to class frequencies
classWeights = list(compute_class_weight(class_weight='balanced', classes=np.unique(train['labels']), y=train['labels']))

# Initializing the classification model using DistilBERT architecture
model = ClassificationModel(
    model_type='distilbert', # Specifying the model architecture
    model_name='distilbert-base-uncased', # Pre-trained model name
    weight=classWeights, # Applying class weights to handle imbalance
    num_labels=4, # Number of unique labels in the classification task
    args={
        'train_batch_size': 10, # Batch size for training
        'learning_rate': 3e-5, # Learning rate
        'num_train_epochs': 4, # Number of training epochs
        'max_seq_length': 512, # Maximum sequence length for tokenization
        'output_dir': f'{path}', # Directory to save model outputs
        'cache_dir': f'{path}', # Directory to cache model components
        'overwrite_output_dir': True, # Overwrite the output directory if it exists
        'use_multiprocessing': False, # Disable multiprocessing (necessary for Google Colab)
        'use_multiprocessing_for_evaluation': False # Disable multiprocessing for evaluation (necessary for Google Colab)
    }
)

# Training the model on the training dataset
model.train_model(train)

# Predicting the labels for the test dataset
predictions = model.predict(list(test['text']))[0]

# Adding the predictions as a new column to the test dataset
test['prediction'] = predictions
```

When fine-tuning, professionals should be sure to use a pretrained model that has not already been fine-tuned for a downstream task (unless your fine-tuning dataset is in the same format as the original fine-tune), because the output will be affected by its prior downstream task.

Class imbalance occurs when the training data are disproportionate to one category versus another. Thus, without adjustments, the training processes will be biased toward the category it has seen most in the training dataset. We can handle this easily in the Simple Transformers library by adjusting the loss function for the empirical probability of the different classes. Other methods require manipulation of the training data, such as oversampling the various classes so that the training data have an equal amount of each class.

Hyperparameters are the nontrainable parameters for the model, thus differing from the parameters of the model that are learned through training. They are set when constructing the architecture and training arguments and include such things as learning rate and epochs (which we explain below). Setting these parameters is where the "art" of ML comes into play as most hyperparameter optimization is done through a trial-and-error process. For this reason, professionals should start with a very small model that can be run quickly and adjust the hyperparameters iteratively. Professionals interested in building models need to understand the following basic concepts regarding hyperparameter optimization:

- **Learning rate:** This determines how much you will adjust the parameters after each iteration of the learning objective. Having too large of a learning rate could adjust the parameters too much at each step, causing the model to overshoot the optimal value of the loss function and continually oscillate around it without ever reaching it. In contrast, too small of a learning rate could cause the loss function to converge too slowly or get stuck in a local minimum without reaching the overall optimal value.
- **Epochs:** An epoch is an entire pass over the training dataset, meaning after a single epoch the model has seen all the training data and iteratively changed its parameters based on minimizing the loss function. More epochs run the risk of overfitting on the training data.

A good starting point for fine-tuning is to use the Simple Transformers library and iteratively test out a small model on a labeled dataset. Using Google Colab's free-tier GPU, professionals can get up and running quickly and become familiar with hyperparameter tuning at little or no cost. After experimenting, one can move on to using the Transformers library, which offers more robust tools for fine-tuning outside of traditional NLP tasks.

Local Generative LLM Fine-Tuning

With the explosive growth in open-source generative LLMs (GenLLMs), new techniques for fine-tuning are being developed rapidly. We make the distinction here between GenLLMs and LLMs because the fine-tuning methods up to this point have focused on nongenerative, bidirectional LLMs like BERT. GenLLMs like GPT and LLaMA are trained on much larger sets of data and have higher parameter counts, thus fine-tuning these models locally has required some innovation from the open-source community. One such novel method that emerged is low-rank adaptation (LoRA; see Hu, Shen, Wallis, Allen-Zhu, Li, Wang, Wang, and Chen 2021). This technique reimagines the fine-tuning process by freezing the pretrained weights in the model and inserting a new, smaller matrix of trainable weights into each transformer layer. By only adjusting this smaller matrix during fine-tuning, LoRA significantly reduces the computational resources required, making it feasible to locally fine-tune GenLLMs without a substantial budget. This method retains the model's ability to generate high-quality responses while allowing researchers to adapt the model to specific tasks more efficiently.

Another method, known as quantization, takes a different approach to optimizing the fine-tuning process. It reduces the precision of the numerical values in the model's parameters without significantly sacrificing the quality of the model's outputs. This technique effectively compresses the model's size, thereby reducing the memory and computational resources required for fine-tuning and inference.

The process for fine-tuning GenLLMs on consumer-grade hardware has been made more accessible by a novel method known as QLoRA (quantization low-rank adaptation; see Dettmers, Pagnoni, Holtzman, and Zettlemoyer 2023). This technique represents an innovative fusion of the LoRA and quantization approaches. QLoRA's unique combination of these two methods drastically reduces the memory footprint and computational intensity, making it feasible for individuals and small teams to fine-tune GenLLMs on consumer-grade hardware, such as ordinary GPUs commonly found in personal computers and free-tier Google Colab instances. A Python demonstration of this approach is beyond the scope of this paper.[4]

Fine-tuning GenLLMs locally is by far the most complex form of fine-tuning we will discuss in this report. However, this method comes with advantages over fine-tuning options with proprietary models through the likes of OpenAI and ChatGPT. As of this writing, GPT-4 is available for fine-tuning only experimentally. The advent of novel base models is inevitable. Given that fine-tuning cannot be ported across different models, each new model will necessitate its own fine-tuning process. Implementing the open-source approach first could be an initial step before committing to a paid option that ultimately may need to be retrained with new base model releases. Additionally, perhaps the most compelling reason to use local models is that running a model and fine-tuning it on local servers give the user the ability to keep all internal data behind the company firewall.

As local models begin to meet or exceed the capabilities of ChatGPT, the combination of both privacy and cost-efficiency will make them an increasingly attractive option for businesses and researchers.

Comparing the Methods

A comparison of the results of few-shot learning and traditional fine-tuning based on the classification task in chapter 3 is shown in **Exhibit 10**. In essence, we are taking ESG-related tweets made by several companies on their own official Twitter pages and using GPT-3.5 and GPT-4 in a few-shot manner to classify these tweets into four categories of materiality and then comparing with the same classification task done using a labeled dataset and traditional fine-tuning methods (refer to the "Labeling" subsection in chapter 3 for a detailed description of the classification task).

Exhibit 10 shows that traditional fine-tuning still outperforms the few-shot classification by both GPT-3.5 and GPT-4 on this task. However, the delta

[4]For a Google Colab tutorial on the topic, see Belkada, Dettmers, Pagnoni, Gugger, and Mangrulkar 2023.

Exhibit 10. Comparing OpenAI Models vs. Traditional Fine-Tuning for Material ESG Classification

Class	F1-Score		
	GPT-3.5 (few shot)	GPT-4 (few shot)	Traditional Fine-Tuning[a]
Not Important	66%	87%	97%
Community Outreach	20%	27%	60%
Industry Recognition	41%	86%	92%
Actions and Innovations	41%	40%	84%
Average	42%	60%	83%

[a]3,223 training examples.

between GPT-3.5 and GPT-4 is significant, at 18%, for the overall average F1 score, suggesting large jumps in performance between model iterations. Moreover, GPT-4 achieved an average 60% F1 score with only a single example for each ESG materiality category vs. 3,223 training examples for traditional fine-tuning methods. The area where traditional fine-tuning seems to shine most in our example is the Actions and Innovations category. This category carried the most nuance and the least consistent language (see chapter 3 for details). Thus, at least for now, traditional fine-tuning can be a valuable addition to the NLP tool kit when the task requires significant nuance and many training examples to capture this nuance.

Dataset Curation

Probably the most critical aspect of fine-tuning any large language model is crafting accurate and robust training datasets. When building a dataset, sometimes only a few bad examples can significantly alter the accuracy of the fine-tuned model. Dedicating time and effort to verify, standardize, and ensure consistency among all training examples can significantly enhance the model's accuracy. Often datasets are shared in the open-source community, and it has become increasingly evident that a smaller, carefully crafted dataset will outperform a larger dataset containing errors.

Optimizing Performance

After exhausting efforts in hyperparameter optimization, professionals may consider exploring ensemble methods to optimize performance. Ensemble methods involve training multiple models and combining their predictions to reach the final prediction. They capitalize on the premise that a group of weak learners can come together to form a strong learner. This is because different models may capture different patterns or aspects in the data, and by

aggregating their outputs, a professional can usually achieve a more holistic and accurate prediction. Ensemble models can be more computationally intensive and time-consuming to train. They also can be more complex and harder to interpret than individual models. Still, when performance is of the utmost importance, these techniques can be a valuable fine-tuning tool.

Conclusion

Fine-tuning large language models is an incredibly valuable technique, yet it carries substantial costs. Both the initial time investment for curating a high-quality dataset and the complex task of understanding neural networks and optimizing hyperparameters are significant hurdles. However, with user-friendly libraries, such as Transformers and Simple Transformers, along with guiding tools, such as ChatGPT, it can be an achievable and rewarding endeavor. One important factor to keep in mind is how quickly the field of NLP is evolving. New techniques and more advanced foundation models are constantly being released. As such, the balance of benefits between fine-tuning and few-shot learning may start to shift in the latter's favor.

3. ESG CASE STUDY

ESG criteria present a dynamic domain for investors to navigate because of the inherent subjectivity and complexity of ESG data. With the high velocity of information in today's environment, material ESG criteria can shift, expanding beyond such conventional issues as CO_2 emissions to encompass such issues as the implications of the war in Ukraine (for supply chain shifts and weapons manufacture) and other geopolitical issues. Furthermore, a large portion of the ESG data available are self-reported and typically lag the current state of a company's ESG actions.

This intricate environment has catalyzed the emergence of solutions leveraging advances in machine learning, notably NLP. NLP offers powerful mechanisms to analyze large volumes of textual data in near real time, helping bridge the information gap that often exists with traditional, lagged ESG reporting. This, in turn, can facilitate a more timely and insightful assessment of a company's ESG practices. Additionally, teams that command the knowledge and expertise to harness these technological tools can evolve in step with ESG dynamics.

Given the highly subjective nature of ESG criteria, significant effort has been directed toward parsing the data to illuminate the types of information that investors react to. Notably, Serafeim and Yoon (2022) conducted extensive research by analyzing investor reactions to ESG news using data from FactSet's Truvalue Labs (TVL). Their work suggested that investors are predominantly driven by news likely to impact a company's fundamentals. While the conclusions drawn by Serafeim and Yoon may seem intuitively obvious, their work importantly addressed the nuanced question of what specific aspects of ESG news capture investors' attention.

In parallel, the case study presented here explores a different path to shed light on what types of ESG disclosures resonate most with investors, utilizing publicly available alternative data and open-source tools as opposed to vendor NLP solutions. The focus lies in demonstrating the application of open-source tools and fine-tuning methods to answer similar, yet uniquely framed, questions.

Overview

This chapter synthesizes the discussion of unstructured and alternative data sources and open-source NLP tools used to work with such data. The synthesis is presented in the form of an illustrative example applying fine-tuning methods to unstructured ESG data. It shows how investment professionals and investment data scientists can work with these tools to enhance their investment processes. Here we discuss the objective of and philosophy behind the case study. We also provide details on our approach, how we generated the signal for our analysis, and our method of index construction.

© 2024 CFA Institute. All rights reserved.

Objective

The essence of this case study is to probe the efficacy of leveraging ML techniques, particularly LLM fine-tuning, to generate alpha from unstructured data by exploring the following three research questions:

1. **Fine-Tuning Feasibility:** Can LLM fine-tuning methods effectively discern various categories of materiality in ESG-related communications?
2. **Materiality Effect:** Among various disclosure classifications, which ones have the most impact on stock prices?
3. **Size Effect:** How does company size influence the material impact of ESG disclosures?

Philosophy

Our philosophical underpinning is the belief that a company's engagement with ESG issues, as reflected in the narrative of management's public communications, leads to expected future improvements in the company's overall ESG profile. Using this belief to explore our questions and to maintain a dynamic repository of such communications, we tapped into the real-time discourse provided by corporate feeds on Twitter (now known as X), leveraging the API to gather a historical archive of company disclosures.[5] Using Twitter provided a single access point to company communications for thousands of companies, thus simplifying the sourcing of the corporate disclosures.

Approach

With this backdrop, we present the following baseline hypotheses and our methods for testing them for each research question:

1. **Fine-Tuning Feasibility:** Our baseline is that fine-tuning can effectively discern between categories of materiality of ESG communications. To test this hypothesis, we fine-tune an LLM in a supervisory fashion[6] with manually labeled tweets classified into four distinct categories based on the level of material impact the ESG-related information may have on stock price. We then evaluate the accuracy of the model predictions on a holdout dataset of unlabeled tweets.
2. **Materiality Effect:** Our baseline is that only the most material tweets resonate with investors, as manifested in stock prices. We use the predictions from the fine-tuned model to classify tweets into the various material categories for all the companies in the Russell 1000 Index. Utilizing the Russell 1000 as a parent index allows for a sufficiently

[5]In July 2023, Twitter was rebranded to X. This study was conducted before this rebranding occurred.

[6]In supervised ML, our labeled data are acting as a supervisor to the training process for the model to effectively discern between categories. This is opposed to unsupervised ML, where there are no labeled data to guide the model.

large number of companies across sectors to be included even after accounting for companies without a Twitter feed. For each company, we use the number of tweets in each category as a portfolio weighting signal (see the "Portfolio Construction" section for weighting signal details). Consequently, we create four portfolios, one for each of the four ESG materiality categories. We then backtest each portfolio's performance using historical returns. We define our most material category of tweets as ones where a firm is disclosing information about actions and innovations it has taken related to ESG (see the "Labeling" section below for other material category definitions). For example, the "Actions and Innovations" category portfolio comprises the Russell 1000 Index constituents reweighted according to the proportion of "actions and innovations" tweets each company has over our sample period. Constituents are reweighted on a quarterly basis (see the "Portfolio Construction" section that follows for further details).

3. **Size Effect:** With ESG data, company size can often play a role in the material impact of the information. This is because much of the ESG data comes from self-reported sustainability reports, which may not be available for smaller companies that have fewer reporting resources (Bos 2017). Additionally, smaller companies typically have less analyst coverage and therefore often have a lower prevalence of information and insight on ESG issues. Therefore, gathering ESG data in near real-time for smaller companies may have higher reward potential than for larger companies. Accordingly, we postulate that the material impact of ESG information is greater for smaller companies than larger companies. To test this baseline, we create four additional portfolios based on a small-cap index—the Russell 2000—with the same weighting scheme as the Russell 1000. We then calculate historical performance and compare the performance of the four category portfolios for the smaller companies to those created for the larger companies (i.e., those in the Russell 1000 mentioned above).

Our ESG tweet categorization focuses on positive information on a company's ESG practices or general ESG awareness issues. Thus, we are not focused on sentiment of the tweets or controversies the company may be involved in. This is because our philosophy is rooted in capturing the growth in ESG scores as management places more dedication toward ESG matters rather than their current ESG scores.

Exhibit 11 outlines the process workflow for the case study. This case study is for illustrative purposes only and does not amount to investment advice or endorsement of a particular investment strategy. The exercise is to enhance the reader's knowledge of ways to use unstructured data to explore interesting ideas that could guide research or resources.

Exhibit 11. Case Study Process Workflow



Data

In this section, we discuss our methods for collecting and labeling the data used in the case study.

Collection

We use the Russell 1000 and 2000 as the large-cap and small-cap parent indexes to explore our hypotheses. Using these indexes allowed us a wide-ranging representation of large and small US companies, ensuring an ample dataset to facilitate the collection of numerous ESG-related tweets and evaluate the differences between the two categories of market capitalization. Moreover, the scale of these indexes mitigates against overconcentration in particular names, thus providing a diversified basis for backtesting the strategies.

Constituents for the Russell 1000 are obtained for the end of the second quarter of 2022. Each company's Twitter handle is obtained using a custom-built Twitter handle scraper that runs a web search on "[company name] + Twitter" using various observed patterns to recognize the most accurate match. Of the 1,023 companies for the Russell 1000, only 54 were deemed to not have an official Twitter account, and of the 2,015 companies in the Russell 2000, 168 were deemed to not have an official Twitter account. Thus, these companies were not included in the analysis.

The tweets for all the companies with a Twitter account are pulled for the period 1 January 2016 to 30 June 2022 using the Twitter API. Replies are filtered out because they are quantity heavy and less reflective of management's initial disclosure intention. After preprocessing and removing duplicates, the final tweet dataset contained 2,854,646 tweets for the Russell 1000 and 2,744,624 tweets for the Russell 2000.

To identify ESG-related tweets, a keyword hashtag list is compiled manually. To construct the list, we start with obvious hashtags, such as #CorporateResponsibility and #sustainability, and from these hashtags, we propagate new ones based on their co-occurrence. For example, in a search for #sustainability, we might find the following:

> Light rail vehicles produce near-zero emissions, making them the right choice for the environment. #sustainability, #NetZero, #GHGreduction

From this tweet, we add #net-zero and #GHGreduction to the list and repeat the processes with these new hashtags. The final list contains 99 unique hashtags.[7] The total number of tweets containing one of the unique hashtags is 39,291 for the Russell 1000 and 33,359 for the Russell 2000, which we dub ESG-related tweets.

Labeling

Our goal with the multiclass model is to separate out the ESG-related tweets into categories of importance. These separate classes are created to filter out potential greenwashing and to identify real actions being taken by the company that may reflect their ESG dedication and innovation. The following are the classification definitions used for labeling.

Not Important (Class 0): Tweets identified as ESG related by a keyword search but that contained no actionable items taken by management

Community Outreach (Class 1): includes certain ESG commitments and small actions taken by the company to highlight these ESG commitments but not likely to affect a company's fundamentals

Industry Recognition (Class 2): includes any mentions of recognition of staff or the company for ESG-related leadership in its industry

Actions and Innovations (Class 3): includes ESG actions that have seen tangible effects or innovative ESG technologies in the works that are most likely to affect a company's fundamentals

Of the tweets identified as ESG related, we manually labeled a random subset of tweets into the four categories. The subset consisted of 3,223 tweets. This amount reflected a balance of gathering more training data and a reasonable amount of dedicated work hours (roughly nine work hours) for the manual labeling task. Of the total, 2,702 are labeled "Not Important" (Class 0), 118 are labeled "Community Outreach" (Class 1), 262 are labeled "Industry Recognition" (Class 2), and 141 are labeled "Actions and Innovations" (Class 3). **Exhibit 12** contains examples of labeled tweets for each category. These were obtained by randomly selecting three labeled tweets from each category.

Model Methods

In this section, we discuss the methods we used to train and evaluate our model. Because much of the material in this section requires a working knowledge of data science concepts, we recommend that members refer to the CFA Program refresher reading on machine learning (CFA Institute 2024) or the CFA Institute Data Science for Investment Professionals Certificate[8] for better comprehension of what follows. Still, we provide content that should be manageable to follow, even for the novice data science practitioner.

[7]See Appendix D for a complete list of hashtags used in the analysis.

[8]https://store.cfainstitute.org/data-science-for-investment-professionals-certificate/.

Exhibit 12. Labeled Tweet Examples

Tweet	Label
Last month China released ambitious plans to curb planet-warming greenhouse gases. With #GreenTechnologies we can tackle the impacts of #ClimateChange.	Not Important
MSCI in the news: Why Does #ESG Matter? Key Items For Investors To Consider In 2018. Read MSCI's Global Head of ESG Research, Linda-Eling Lee's interview @Forbes.	Not Important
#ESG priorities are becoming increasingly important for shareholders. See how this trend impacted #ProxySeason.	Not Important
Celebrating 10 years of our #Harmony wheat #sustainability program @Salondelagri in #France. Through #Harmony, we partner with farmers to preserve the environment in Western European countries. #Impact4Growth #SIA2018	Community Outreach
#CSR: Volunteers from Genpact Bengaluru, India continually work to protect & rejuvenate #SowlkereLake. Watch video:	Community Outreach
Thanks to our NS volunteers who collected 960 lbs. of trash for Clean the Bay Day, sponsored by @chesapeakebay. #teamwork #CSR	Community Outreach
Continuing our commitment to climate action, sustainability and #CSR efforts we were awarded the Bronze Class Distinction by RobecoSAM and we were the #2 biotech company on @RobecoSAM 2017 #DJSI World Index. #ESG #ProudMoments2017	Industry Recognition
We are proud of our Oregon team for winning @SWANA's Silver Excellence Award in the Composting Systems category. #BluePlanet #Sustainability	Industry Recognition
We're proud to be recognized as a global #sustainability leader on @CDP, 2017 Climate #AList:	Industry Recognition
we recently became the first u.s. kidney care provider to completely power our north american operations with 100% renewable energy! read more about our #sustainability efforts in @denverpost:	Actions and Innovations
Our @gladproducts plant in Amherst, Virginia, is now our 9th facility to achieve zero-waste-to-landfill status, bringing us closer to our goal of 10 zero-waste-to-landfill sites by 2020. Learn more in the blog. #Sustainability	Actions and Innovations
Proud of our #CocoaLife #sustainability program! In 2017, the program reached 120K+ farmers and we grew sustainably sourced cocoa to 35% of our needs. More in our 2017 Progress Report:	Actions and Innovations

Training

The original labeled dataset, comprising 3,223 labeled samples, is divided into an 80% training dataset and a 20% testing dataset. The 80% training portion is further partitioned into training and validation sets, using a 5-fold partitioning approach, meaning five datasets are created, each split 80/20 training/validation. For each fold, a separate model is trained on 80% of the training data, resulting in five distinct models, each covering a different portion of the training data. These individual models are then combined using a majority vote classifier to form an ensemble, thus capturing the information in the

entire training dataset. All evaluations are conducted solely on the separate 20% testing dataset to highlight the benefits from the ensemble approach. This method, leveraging elements of K-fold partitioning and majority voting, aims to enhance the labeling task's robustness and accuracy, especially in the context of a small dataset.

As noted above, our dataset is class imbalanced, meaning we have a different amount of samples for each class. For example, there are 2,702 "Not Important" tweets vs. 141 "Actions and Innovations" tweets. This class imbalance is an important feature to account for when training our data as without accounting for it, the model would overclassify the data into the "Not Important" class. In our case, class weights are a hyperparameter for the model that change the loss function during training to account for the class imbalances in our dataset. We use DistilBERT as our foundation model, which is a smaller BERT model that runs 60% faster than BERT and maintains 95% of BERT's accuracy. We fine-tune for 4 epochs and a starting learning rate of 0.00003. This approach was chosen after observing the training of several models with an early stopping around 3.5 epochs and an evaluation of the validation loss starting to increase around this point.

Evaluation

For the evaluation of the individual models, we calculate the arithmetic mean of F1, precision, and recall scores for each ESG materiality class. For example, the F1 score for the majority vote classifier is the arithmetic mean of the F1 score for each class. By taking the arithmetic mean, we are placing an equal weight across the classes, thereby avoiding a bias due to class imbalance. Precision is the ratio of true positive predictions to the sum of true positive and false positive predictions, which essentially quantifies the accuracy of the positive predictions made by the model. Positive predictions refer to the model predicting a tweet to correctly belong to a class. For example, when calculating the precision of the "Actions and Innovations" class, we are calculating the correctly identified instances of tweets belonging to that class relative to the total predicted instances of the class. Recall, in contrast, is the ratio of true positive predictions to the sum of true positive and false negative predictions, which measures the ability of the model to capture the actual positive cases. For example, if we had a tweet that stated "We achieved carbon neutrality for the 7th year in a row!"—its true label would be "Actions and Innovations"; however, a false negative could be a prediction that the tweet was "Not Important." The F1 score is the harmonic average of precision and recall, providing a single metric that balances both the concerns of precision and the need for recall.

Portfolio Construction

We create a portfolio for each of the tweet classification categories.

$$P_{NI,} P_{CO,} P_{IR,} P_{AI},$$

where

NI = Not Important

CO = Community Outreach

IR = Industry Recognition

AI = Actions and Innovations

The signal for entry into a category portfolio is a tweet identified as belonging to that category. Duplicate tweets are removed to avoid double counting. The portfolios are rebalanced quarterly by recalculating the weights based on the new distribution of tweets from the previous quarter. This approach of using the previous-quarter tweets for the current-quarter weights allows us to avoid any look-ahead bias from data that would not be available at the time.

Each tweet is given a score of 1 and smoothed by employing an 18-month exponentially weighted moving average (EWMA). The EWMA method is advantageous for several key reasons:

- **Realistic Holding Assumptions:** It mitigates the impact of sporadic ESG-related disclosures by companies, allowing more companies into the portfolio each quarter (rather than just those that tweeted in that quarter), avoiding excessive turnover.
- **Investor Reaction Time:** The smoothing effect of EWMA grants investors additional time to assess and act on company disclosures vs. absolute quarterly rebalancing.
- **Recency Weighting:** By applying a greater weight to more recent information, the EWMA model ensures that the latest disclosures are more reflective in the scores.
- **Portfolio Inclusion:** Smoothing allows for broader inclusion of companies in portfolios, particularly when portfolio categories have sparse tweet volumes in a given quarter.

The EWMA score is calculated using the formula:

$$S_{i,t}^{(k)} = \lambda X_{i,t} + (1-\lambda) S_{i,t-1}^{(k)},$$

where

$S_{i,t}^{(k)}$ is the EWMA score for company i in portfolio k at time t

k represents the portfolio category (NI, CO, IR, AI)

t represents the relevant quarter in the period from Q3 2019 to Q3 2022

$\lambda = \frac{2}{(\text{Number of periods}+1)}$ is the decay factor, representing the degree to which the previous period's score is weighted. λ is chosen to reflect the intended memory span; in this case, 18 months

$X_{i,t}$ is the most recent number of unique tweets for the company in a given quarter

$S_{i,t-1}^{(k)}$ is the EWMA score from the previous period

This formula ensures that newer scores have a more significant impact while older scores gradually diminish in influence over the 18-month period.

The weight for a single company in a given portfolio is the previous quarter EWMA score for the company relative to the previous quarter total EWMA score for all companies in the given portfolio:

$$w_{i,t}^{(k)} = \frac{S_{i,t-1}^{(k)}}{\sum_{i=1}^{n} S_{i,t-1}^{(k)}},$$

where

$w_{i,t}^{(k)}$ is the weight of company i in portfolio k at time t

n is the total number of companies in the parent index

For portfolio P_k at time t, the portfolio is an aggregation of the weights of all companies included:

$$P_{k,t} = \sum_{i=1}^{n} w_{i,t}^{(k)}$$

The total weight of portfolio $P_{k,t}$ must be equal to 1.

To maintain a diversified portfolio and mitigate the risk of overconcentration in any single asset, we introduced a 5% weight cap on individual holdings. This cap reflects realistic investment constraints and ensures that our portfolio does not become overly dependent on the performance of any one name. The 5% weight cap is implemented as follows:

1. **Ordering and Identification**
   - We begin by ordering the existing weights of our assets from largest to smallest.
2. **Reassigning Weights**
   - To ensure that no individual asset exceeds the 5% threshold, we set a new weight to be the lesser of the original weight or 5%.

3. **Calculating Allocation Differences**
   - We determine the total difference in allocation caused by this capping, which is the difference between 100% and the sum of the new weights (excess weight).
4. **Proportionate Redistribution**
   - We redistribute the excess weight proportionately across assets below the 5% cap. Each asset's additional weight is calculated relative to the sum of the weights of all uncapped assets.
5. **Iterative Adjustment**
   - We update the weights with this additional allocation. The process is repeated—starting from the identification step—until all assets comply with the 5% cap.

This iterative approach ensures that the cap is respected without creating significant distortions in the intended allocation of the portfolio.

This entire process is done for both the companies in the Russell 1000 and separately for the companies in the Russell 2000, for a total of eight portfolios. In doing so, we can compare the portfolios across the different materiality categories to help answer research question 2 and evaluate the differences in the returns between the two market capitalizations to help answer research question 3. The portfolios derived from the Russell 1000 constituents are subsequently labeled large-cap portfolios, and the ones derived from the Russell 2000 are subsequently labeled small-cap portfolios.

Results

In this section, we provide details on the performance of our model and the various portfolios.

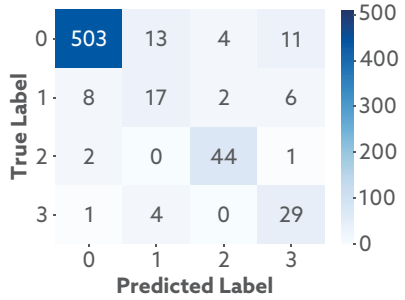
Model Performance

Exhibit 13 shows the model metrics and confusion matrices for each of the 5-fold sample datasets validated on the test dataset of 645 tweets. The labels in the matrix represent the classes (0 = Not Important, 1 = Community Outreach, 2 = Industry Recognition, and 3 = Actions and Innovations).

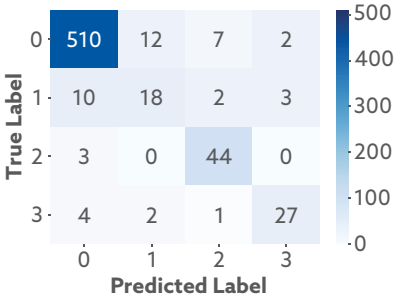
Exhibit 14 shows the metrics and confusion matrix based on the majority vote classifier, which is used as the final prediction for the model. To interpret the confusion matrix, for example, we can see that 516 Not Important (class 0) tweets were accurately predicted and 3 tweets were predicted as Actions and Innovations (class 3) that were in fact Not Important.

Exhibit 13. Model Metrics and Confusion Matrices

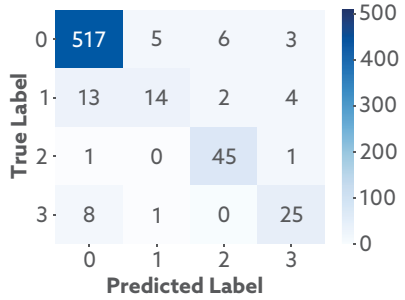
Model 1	
F1 Score	77%
Precision	74%
Recall	81%



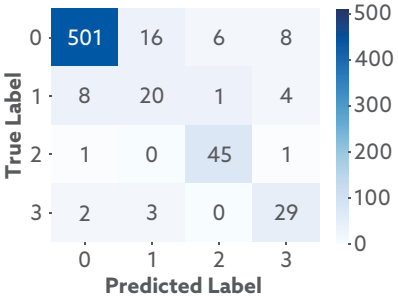
Model 2	
F1 Score	80%
Precision	80%
Recall	81%



Model 3	
F1 Score	79%
Precision	82%
Recall	77%



Model 4	
F1 Score	80%
Precision	76%
Recall	84%



Model 5	
F1 Score	79%
Precision	78%
Recall	81%



Notes: A confusion matrix is a table used in classification problems to visualize the performance of a model. It is a special kind of contingency table, with two dimensions ("True Label" and "Predicted Label"). Each dimension has "Positive" and "Negative" values. The matrix illustrates when the model gets confused and mislabels the classes, shown by the off-diagonal elements. In this case, the numbers in the matrix represent the number of tweets in the test set being classified, with each matrix summing to 645.

Lastly, **Exhibit 15** breaks out the performance metrics for each individual materiality category for the ensemble model.

Portfolio Performance

The quarterly weights are applied to the daily returns for the evaluation period for each portfolio. To maintain a comparable track record across all portfolios, the first quarter for which all portfolios have an 18-month EWMA track record is used as the beginning of the evaluation period, which is Q3 2019. To benchmark our performance, we generate hypothetical returns for 1,000 portfolios that were randomly assigned weights (subject to the same 5% weight cap) to each constituent of the parent index on a quarterly basis. Our benchmark portfolios

Exhibit 14. Majority Vote Model Metrics and Confusion Matrix

Majority Vote	
F1 Score	83%
Precision	83%
Recall	84%

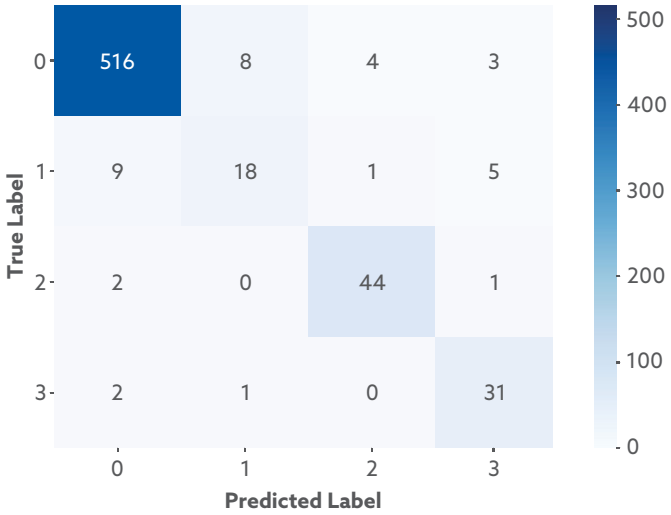


Exhibit 15. Quality Category Model Metrics Breakout

Class	Precision	Recall	F1-score	Count
Not Important	98%	97%	97%	531
Community Outreach	67%	55%	60%	33
Industry Recognition	90%	94%	92%	47
Actions and Innovations	78%	91%	84%	34

thus follow the same methodology as in the creation of the main category portfolios only with random weights as opposed to actual weights generated from tweets.

Exhibit 16 and **Exhibit 17** show the hypothetical growth of a dollar invested in each of the large-cap and small-cap portfolios, respectively, relative to the 95th percentile confidence interval for the daily return of random portfolios. The percentile confidence interval is a nonparametric confidence interval that uses the interval between the 2.5th percentile and the 97.5th percentile of the generated random portfolio returns. Using this method allows us to avoid assumptions of the underlying distribution of returns.

From Exhibit 16, for large-cap portfolios, we see that there is not much difference in the returns among the various ESG portfolios, and the least material, "Not Important" portfolio actually slightly outperformed. In small-cap portfolios, shown in Exhibit 17, the "Actions and Innovations" portfolio clearly outperforms. This latter result is consistent with the findings of Serafeim and Yoon (2022)—that only the most material ESG disclosures drive performance.

Exhibit 16. Large-Cap 18-Month EWMA Dollar Growth vs. 1,000 Random Portfolios' 95th Percentile Confidence Interval

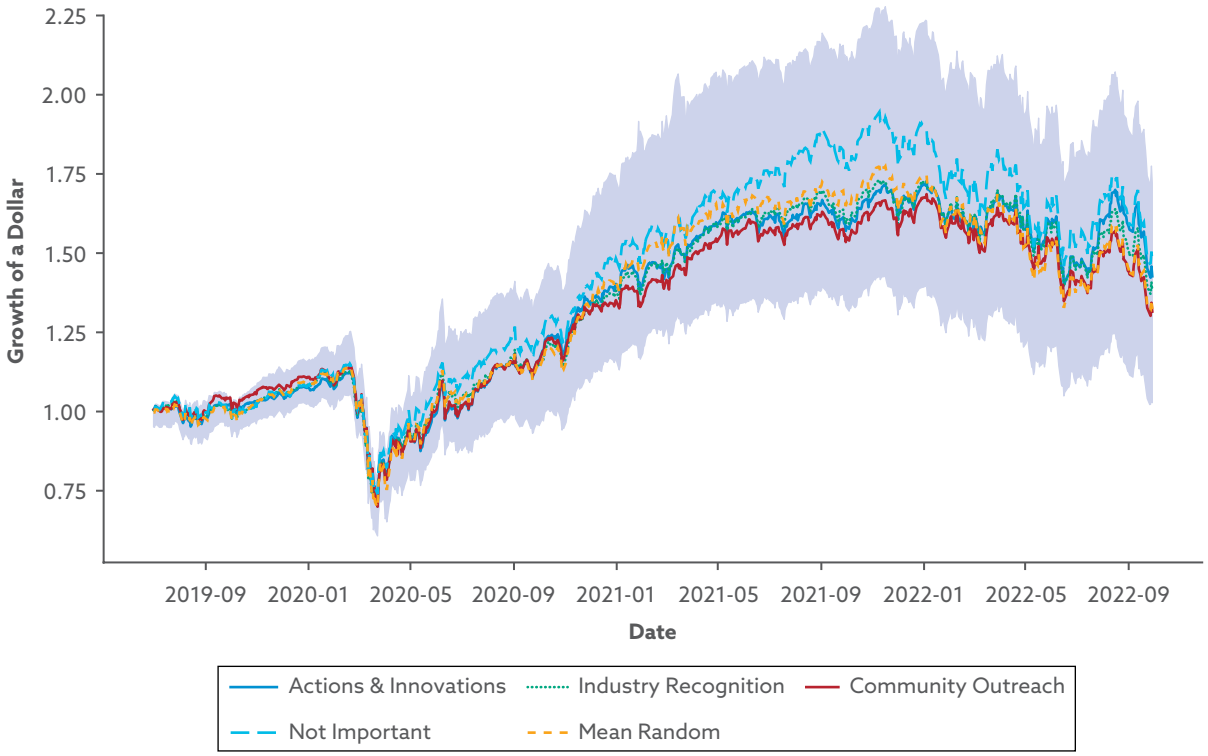


Exhibit 18 and **Exhibit 19** show the number of companies in the index in the first quarter for each year for the large-cap and small-cap portfolios, respectively.

From both Exhibit 18 and 19, we see a steady rise in constituents over time. This isn't surprising as the popularity in ESG grew heavily during this time period, making companies more likely to tweet about their ESG practices and enter the portfolios.

Exhibit 20 and **Exhibit 21** show the hypothetical Sharpe ratios of the various portfolios and the benchmark random portfolios for the large-cap and small-cap portfolios, respectively.

The Sharpe ratios for large cap in Exhibit 20 show a slightly different story than absolute performance: All ESG portfolios marginally outperformed the average with the exception of "Community Outreach" over the 3-year period. Similarly, for small cap, with the exception of the "Industry Recognition" portfolio, the ESG portfolios outperformed the average, with "Actions and Innovations" having a significantly higher Sharpe ratio over the 3-year period.

Exhibit 17. Small-Cap 18-Month EWMA Dollar Growth vs. 1,000 Random Portfolios' 95th Percentile Confidence Interval

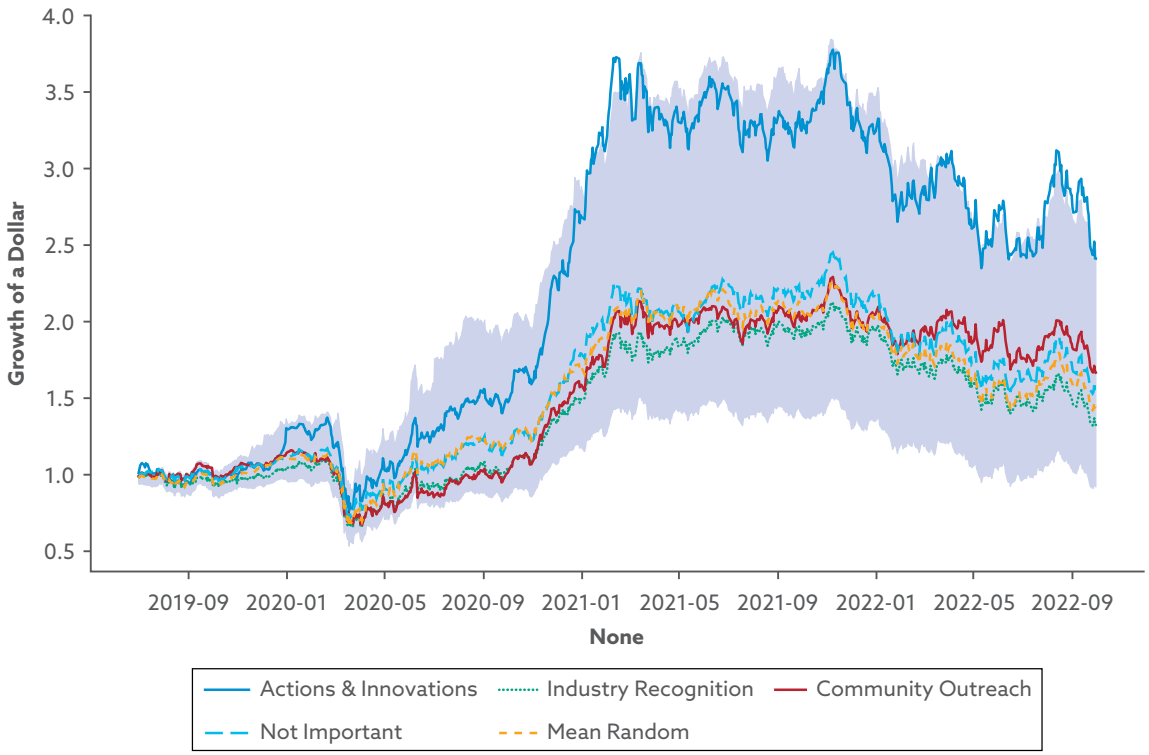


Exhibit 18. Large-Cap Number of Constituents

Portfolio	2020	2021	2022
Actions and Innovations	167	207	252
Industry Recognition	282	331	369
Community Outreach	154	183	221
Not Important	521	586	639

Exhibit 19. Small-Cap Number of Constituents

Portfolio	2020	2021	2022
Actions and Innovations	76	110	152
Industry Recognition	101	157	198
Community Outreach	57	83	118
Not Important	512	622	761

Exhibit 20. Large-Cap Sharpe Ratios

Portfolio	1 Year	3 Year
1000 Random Portfolio Avg	(1.01)	0.25
Actions and Innovations	(0.68)	0.34
Industry Recognition	(0.84)	0.31
Community Outreach	(0.90)	0.19
Not Important	(0.82)	0.40

Note: Parentheses indicate negative values.

Exhibit 21. Small-Cap Sharpe Ratios

Portfolio	1 Year	3 Year
1000 Random Portfolio Avg	(1.22)	0.33
Actions and Innovations	(0.85)	0.77
Industry Recognition	(1.19)	0.25
Community Outreach	(0.73)	0.41
Not Important	(1.05)	0.38

Note: Parentheses indicate negative values.

Discussion

To facilitate our discussion of the results, we start with the first research question we posed earlier in chapter 3.

1. Fine-Tuning Feasibility: Can LLM fine-tuning methods effectively discern various categories of materiality in ESG related communications?

Our initial hypothesis was, yes, fine-tuning could effectively discern between categories of materiality of ESG communications. Overall, the classifiers were consistent with this hypothesis.

As presented earlier, Exhibit 13 shows the five models with relatively similar results (averaging around 79% in F1 score), with several models showing a significant drop in F1 score due to the poor classification of class 1, "Community Outreach." This class also seemed to be very ambiguous for human labelers, because small actions that do not have a material impact on stock price can vary easily between "Not Important" and "Community Outreach." Class 0, "Not Important," had the highest performance, with a 97% F1 score, and it had

the highest number of observations. Class 2, "Industry Recognition," had the second highest performance, with a 92% F1 score. This class contained the most consistent language throughout the tweet samples. For example, many of the tweets in this class contained such language as "we are proud to be recognized" that could be easily interpreted by the model to belong to this class. Class 3, "Actions and Innovations," had lower but still strong performance, with an 83% F1 score. This class had a more semantically complex labeling task. For example, the classifier seemed to understand that when the tweet was describing detailed actions, it belongs to this class; however, sometimes the detailed actions were not done by the firm or were not fully relevant to ESG issues. Still, some of this ambiguity was overcome by using the majority vote classifier as our final label. The majority vote classifier showed a nice boost in F1 score, bringing it up to 83%, 4 percentage points higher than the average of the individual models (see Exhibit 14). Collectively, the models performed well, given a training set of only 3,223 samples, and reasonably classified the various categories of disclosures.

Now, we turn to our second research question.

2. Materiality Effect: Among various disclosure classifications, which ones have the most material impact on stock price?

Our initial hypothesis was that only the most material tweets resonate with investors. Our findings show mixed results for large cap but are consistent with our hypothesis for small cap, which will be discussed in the next section on size effect.

In large cap, the "Not Important" portfolio marginally outperformed the other portfolios. Its performance exceeded that of the mean average portfolio but was well within the 95th percentile confidence interval, as shown in Exhibit 16. This outperformance was maintained on a risk-adjusted basis, as shown in Exhibit 20, with a three-year Sharpe ratio of 0.4, versus 0.25 for the random portfolio benchmark.

The "Not Important" portfolio had the highest number of constituents; however, when looking at the sector weights shown in **Exhibit 22**, we see a high weight to technology. When investigating these tweets, it was apparent that many tweets related to data security were in the "Not Important" class. For example, many data security companies often tweeted about the threats posed to firms with no material actions or innovations mentioned. After removing data security tweets from the portfolio, the sector allocations are less concentrated in technology and the outperformance is reduced, making it closer in line with the average random portfolio, as shown in **Exhibit 23**. All the other large-cap portfolios are in line with the random portfolio average, thus making it difficult to discern any differences in reactions from the various categories for large-cap ESG disclosures.

Exhibit 22. Large-Cap Sector Weight Snapshot for "Not Important" vs. "Not Important ex Data Security" Category Portfolios

	Not Important		Not Important ex Data Security	
Sector	Q3 2019	Q3 2022	Q3 2019	Q3 2022
Basic Materials	6%	5%	9%	8%
Communication Services	0%	0%	0%	0%
Consumer Cyclical	6%	4%	8%	6%
Consumer Defensive	6%	3%	10%	5%
Energy	1%	1%	1%	1%
Financial Services	12%	14%	14%	16%
Healthcare	6%	6%	8%	9%
Industrials	19%	16%	21%	20%
Real Estate	2%	3%	2%	3%
Technology	36%	45%	18%	26%
Utilities	6%	4%	9%	6%

Exhibit 23. Large-Cap "Not Important" Category Q3 22 Annualized Return Performance Snapshot

Portfolio	1 Year	3 Year
1000 Random Portfolio Avg	–21%	10%
Not Important	–17%	14%
Not Important ex Data Security	–16%	11%

These findings lead into the third research question.

3. Size Effect: How does company size influence the material impact of ESG disclosures?

Our initial hypothesis was that due to lack of resources in sustainability reporting and generally lower analyst coverage, the reward potential from near real-time material ESG disclosures may be higher for smaller companies versus larger ones. Our findings are consistent with this hypothesis.

Exhibit 17 showed that the highest-performing small-cap portfolio is the "Actions and Innovations" category. Furthermore, the performance of this

portfolio exceeds the 95th percentile confidence interval of the random portfolio at various times throughout the testing period. This outperformance is also consistent on a risk-adjusted basis as shown in Exhibit 21, with a 0.77 Sharpe ratio versus the average of 0.33 for the 3-year period. All other portfolios seem to be in line with or close to the average random portfolio.

When exploring sector concentrations to explain the performance of the "Actions and Innovations" small-cap portfolio, we can see a large allocation to industrials in **Exhibit 24**.

Diving further into the details, **Exhibit 25** shows five random samples of tweets from companies in the industrials sector. Clearly, some are more material than others, but the classifier seems to capture some of the most likely tweets that could affect a company's fundamentals. Thus, the time advantage to small-cap investors in gaining insight into material ESG issues through Twitter versus traditional ESG data may be rewarded by portfolio outperformance. Consequently, while the materiality effect showed mixed results for large cap, only when it interacts with the size effect does ESG materiality deliver performance dispersion.

Exhibit 24. Small-Cap Sector Weights over Time for "Actions and Innovations" Portfolio

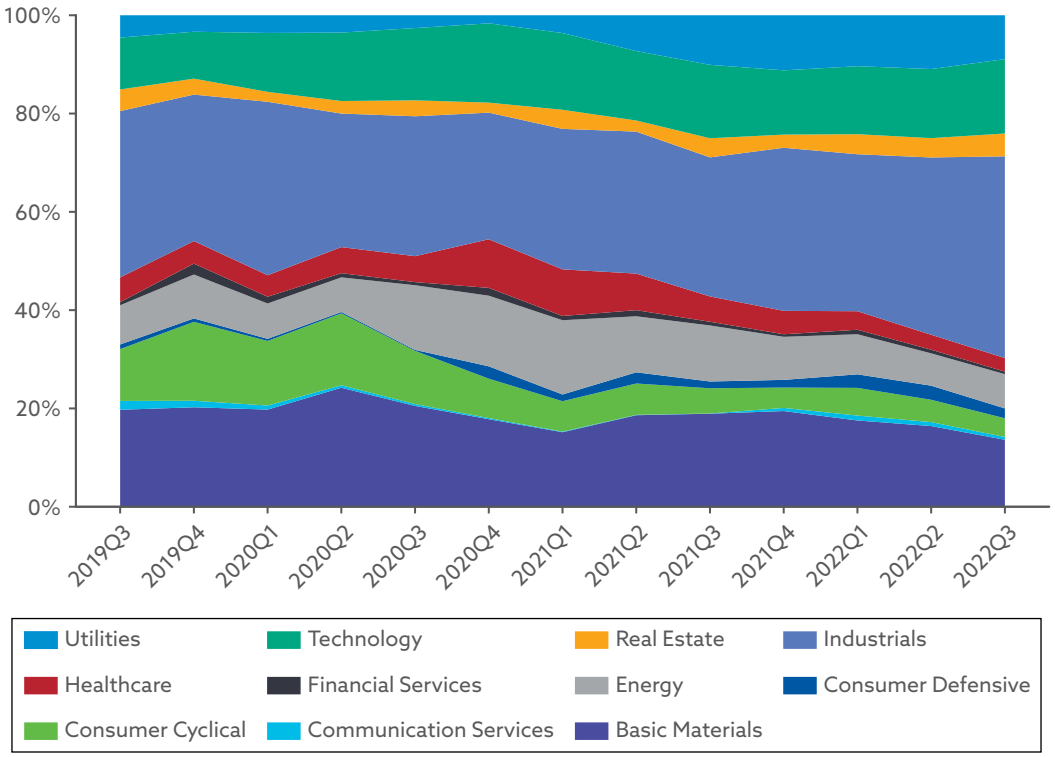


Exhibit 25. Example "Actions and Innovations" Tweets from the Industrials Sector

Tweet
vectrus is among contractors leading base composting initiatives with biodegradable materials in support of the @usarmy's environmental sustainability goals. read more here: #sustainability
breaking: we're excited to announce we've entered an agreement with the @augustaeda to build our second us plant, a new 200-acre location in augusta, georgia. @gdecd big day for purecycle! full press release: #expanding #construction #recycling
We're thrilled to announce our partnership with teco 2030 to develop #carboncapture solutions for the maritime industry. our goal is to help shipowners make their vessels more environmentally friendly, helping them to exceed imo standards. read more here:
white construction was awarded the glacier sands wind farm construction contract in mason county, illinois. engineering, procurement and construction of 43 wind turbines will be self-performed generating 185-megawatts of #renewableenergy
our scientists and microbiologists analyze thousands of potable and non-potable water samples annually. we have one of the largest microbiology capacities in california. #environmentalservices

Conclusion

Overall, fine-tuning large language models for specific use cases has its advantages. In this case study, we used the technique to gain further insight on how investors react to ESG-related company disclosures. While this case study yielded interesting results by pointing to the importance of real-time information advantages from alternative data, we note that these performance results occurred during a time when ESG issues were at the forefront of investors' attention and may not persist outside the limited evaluation period. Additionally, such imperfect models as the fine-tuned model in this case study should be used with caution when applied to small datasets. The value in using such models comes from being able to test a hypothesis on a large dataset—in this case, the Russell 1000 and 2000—providing a basis for further investigation and resource allocation.

We saw throughout this report the various events leading to the proliferation and value of alternative and unstructured data. Being able to leverage the tools and techniques to parse these data, particularly with NLP, is an invaluable resource that should not be left unexplored. We saw in this case study how we can leverage these tools, like LLM fine-tuning, to yield meaningful insights about ESG and performance. The role of the investment professional is changing rapidly. Staying abreast of technological trends, mastering programming languages for parsing complex datasets, and being keenly aware of the tools that augment our workflow are necessities that will propel us forward in an increasingly technical finance domain.

APPENDIX A. ALTERNATIVE DATA GLOSSARY

Publicly Available Government Data: Data published by government agencies or departments. Examples of use include census data to provide insights into population demographics, which can be useful for real estate or retail investments.

News and Media Sentiment Data: Analysis of news or media broadcasts to gauge sentiment toward a company, commodity, or market. An example is training an AI model to detect the sentiment from news articles to predict stock movements.

Employment Data: Information on job vacancies, job descriptions, salaries, and so on. This category includes assessing hiring activity in a sector to determine its growth.

Web-Scraped Data: The automated extraction of information from websites. An example is scraping retail websites to determine product price trends to indicate changes in demand.

Environmental, Social, and Governance Data: Information that measures the sustainability and societal impact of an investment in a company or business. These data can include metrics on environmental impact, social responsibility, and corporate governance. For example, investors might use ESG data to identify companies that are leaders in sustainable practices, which could signal long-term value and stability.

Social Media Data: Data derived from social media platforms regarding user sentiment, trends, and behaviors. An example is tracking brand sentiment on Twitter to predict sales growth.

Real Estate Data: Tracking property values, rental rates, property transactions, and so on. An example is tracking property price trends to inform decisions on REIT sector allocations.

Consumer Reviews Data: Feedback from consumers on products or services. Examples include training an AI model to detect sentiment on a product to project future growth potential.

Transcription Data: The use of written records of spoken content, such as earnings calls. In this use case, an investor could train an AI model to predict stock price movements based on frequencies of certain words used in earnings calls.

Energy Consumption Data: Information on energy usage patterns. An example is monitoring electricity consumption trends to inform investments in utility companies.

E-Commerce Data: Data about online sales, cart abandonment rates, and customer behavior. An example use case is tracking best-selling items on a platform to provide insights into consumer preferences.

 © 2024 CFA Institute. All rights reserved.

Supply Chain and Logistic Data: Information about the flow of goods from manufacturer to consumer. An example is monitoring disruptions in supply chains to predict stock shortages.

Credit Card Transaction Data: Data about credit card purchase activity. Example uses are often based on evaluating consumer spending habits in real time.

Weather Data: Data relating to climate conditions. Uses are usually based on predicting weather patterns to inform decisions for agricultural investments.

App Download Data: Data on the number and frequencies of application downloads. For example, monitoring the download frequency of a new gaming app could be used to predict revenue potential.

Court Records and Legal Data: Public records of legal proceedings can be used to track litigation against a company to highlight potential risks to its stock price.

Satellite Imagery: Images captured from satellites to analyze changes on earth. This category includes images of parking lots of retail stores to indicate consumer traffic patterns.

Insider Trading Data: Data about stock purchases and sales by company executives. Purchases and sales by insiders could signal confidence or concerns about the future growth of the company.

Patent and Intellectual Property Data: Data on patents filed or granted. Patterns in patents related to a specific technology can help predict sector performance.

Crypto Data: Data related to cryptocurrency markets, including but not limited to price, volume, transactions, and blockchain analytics. Examples include analyzing transaction volumes on different blockchain networks to gauge cryptocurrency adoption rates or utilizing price and exchange data to develop trading strategies.

Geolocation Data from Mobile Foot Traffic: Data derived from mobile device locations. These data can be used to predict retail sales from foot traffic in malls.

Data from Internet of Things (IoT) Devices: Data generated from devices connected to the internet. For example, tracking energy usage patterns from smart thermostats can guide utility investment decisions.

Flight Tracking Data: Public data on flights, schedules, delays, and passenger counts. For example, monitoring frequent flight cancellations might indicate challenges for an airline.

Clickstream Data: Data on user interaction patterns on a website. An example is accessing the click trends of customers to evaluate the effectiveness of a new product launch.

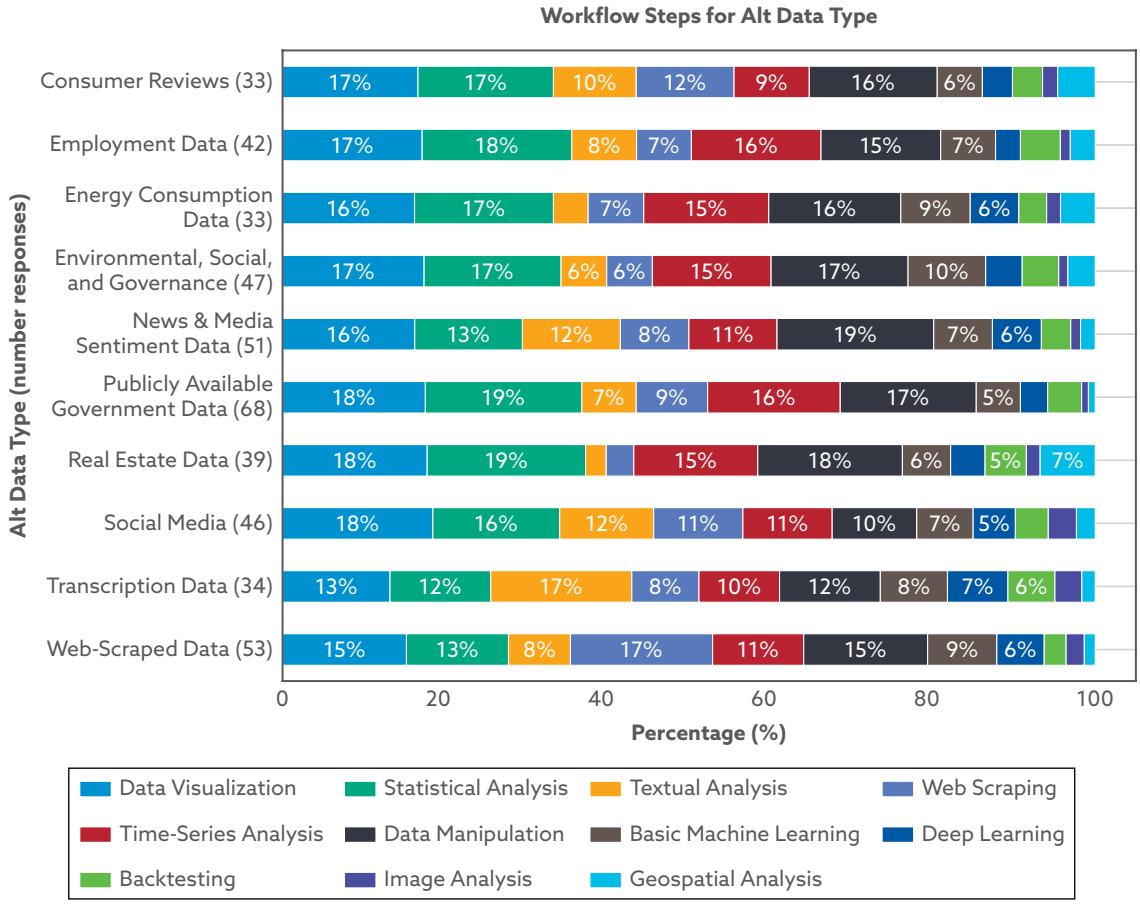
Sensor Technology Data: Data from various sensors, such as manufacturing equipment. Monitoring machine health in real time can predict maintenance costs and downtimes.

Wearables Data: Data generated from wearable devices, such as smartwatches. Upticks in the use of wearable devices can indicate adoption patterns and drive performance predictions for companies investing in this sector.

APPENDIX B. WORKFLOW STEPS FOR ALTERNATIVE DATA TYPES

Exhibit B1 showcases the variation in workflow steps for each alternative data type, with the number of respondents indicated in parentheses. Only alternative data types with 30 or more respondents are included.

Exhibit B1. Detailed Breakout of Workflow Steps for Various Alternative Data Types



Source: From the July 2023 CFA Institute survey on alternative and unstructured data. See footnote 1 for more details.

© 2024 CFA Institute. All rights reserved.

APPENDIX C. SELF-ATTENTION

This section focuses on self-attention, which is the underlying engine of modern large language models, like GPT and BERT. Thus, the section is not specific to any one model but is a generalized overview of what allows these models to learn the complexities of our language.

One of the first steps in pretraining an LLM is to establish the vocabulary. This step is typically accomplished by tokenizing a large corpus of text into distinct units, with each unit being a word or subword in the language; these are tokens. Previous models used whole words, but this approach becomes problematic when encountering rare or unseen words. Instead, the original transformer model adopted a subword tokenization approach known as byte-pair encoding (BPE). This method starts by treating each word as a sequence of characters, and then it progressively merges the most frequent pairs of characters or character sequences into a single unit. Through this process, BPE generates a vocabulary of variable-length tokens, ranging from single characters to common words or subwords.

Each sentence is tokenized, and each token is mapped to a vector representation in high-dimensional space. For example, in the first paper on transformer architecture (Vaswani et al. 2017), the vector dimensionality was 512, so the word "jumped" might, for example, be split into "jump" and "ed" as two tokens that would be mapped to vectors like the following:

$$\text{jump} \rightarrow [0.25, -0.1, 0.84, -0.2, \ldots, 0.64, -0.75, 0.02, -0.46]$$

$$\text{ed} \rightarrow [-0.55, 0.93, 0.08, 0.72, \ldots, -0.33, 0.43, -0.39, 0.69]$$

The length of the arrays in this example would equal 512. Of course, the vector would contain continuous values with a higher level of precision. Keep in mind that this high-dimensional space is not something our brains can visualize, but this complexity is exactly what is needed to capture the subtleties of our language.

Before training begins, these numbers are completely random. They are the first set of learnable parameters in the model, which means as the model trains, these parameters will be adjusted to minimize the loss function, similar to how an iterative version of ordinary least squares would change the beta parameters to minimize the squared errors. These vectors are known as the word embeddings and capture the nuances of the language in the training dataset once adequately trained. An additional embedding is made for the position in the sequence, very similar to word embeddings but with the position of the token in the sequence mapped to a vector. Adding the two vectors together yields the input for the attention layer, which is the first layer in the neural network.

 © 2024 CFA Institute. All rights reserved.

The attention layer is designed to learn patterns and relationships between words in a sentence, just as a search engine might recognize the relevance of certain websites to a given query. However, instead of relying on hard-coded rules or ranking algorithms, the attention layer learns these patterns directly from the data. To achieve this, each token in a sequence is processed by the attention layer and is transformed into three different vectors: Query (**q**), Key (**k**), and Value (**v**). These transformations are performed through multiplication with three distinct sets of learned weights—one for each type of vector. These weights are the second set of learnable parameters in the model (see **Exhibit C1**).

Each token in the sequence is transformed to a **q**, **k**, **v** vector creating three identical matrices of $n \times a$ dimension, where n is the sequence length and a is the dimension of the weight matrix.

For example, in the sequence "The cat jumped on the ledge," we might have the three matrices shown in **Exhibit C2**.

Exhibit C1. Visualization of Token Transformation into Query, Key, and Value Vectors

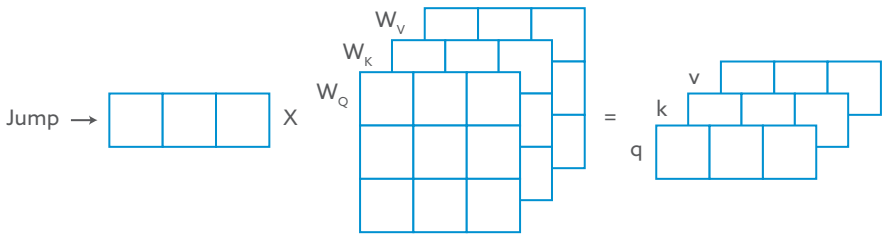
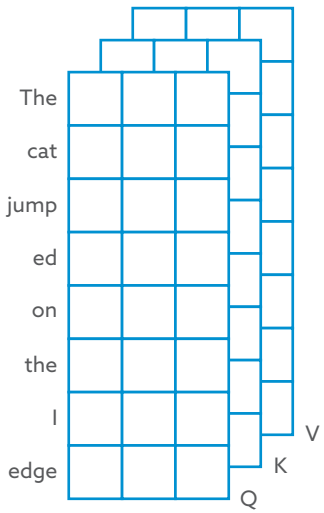


Exhibit C2. Visualization of Sequence Query, Key, and Value Matrices

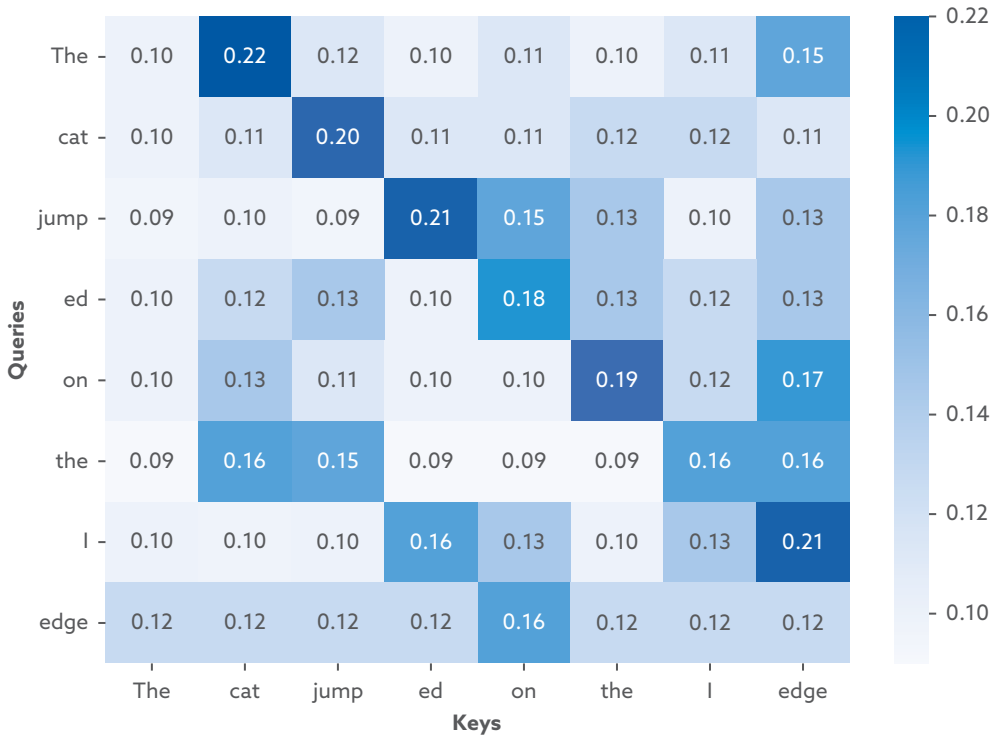


The "Query" is like a search query you input into the search engine, the "Key" is like the tags or metadata associated with every webpage in the engine's database, and the "Value" is like the actual content of the webpage. When a query is made, the model computes an attention score by multiplying the Q and K matrices, essentially matching each token as a query against all the other tokens in the sequence as keys. This is analogous to how a search engine ranks webpages based on how well their metadata match the search query. The scores are then normalized using the softmax function. This function turns the matrix into values between zero and one, creating a distribution where higher scores indicate greater relevance of one token to another. You can think of this as an attention weight. In our oversimplified example, the trained attention matrix might look like the matrix in **Exhibit C3**.

In the third row, we see that the hypothetical model places the most weight for the token "jump" with the token "ed"—while also placing weight on the token "on." The model has learned the attention it should place on tokens given other tokens and also the position at which the tokens lay.

These normalized scores are then used to weight the V (Value) matrices, a process akin to reading and aggregating content from the

Exhibit C3. Simplified Representation of an Attention Matrix



Note: These values are made up for illustration purposes and do not reflect model values.

most relevant webpages. The entire process is known as scaled dot-product attention (see **Exhibit C4**):

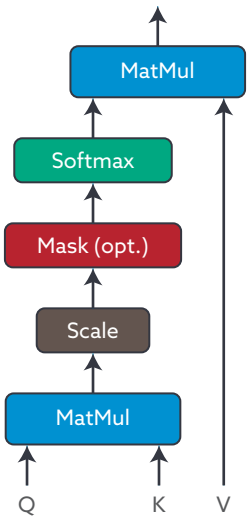
$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V.$$

Note that the term $\sqrt{d_k}$, where d_k is the dimensionality of the **k** vector, is used to scale down the dot products, which helps with the stability of the model training.

The output is a new representation of the token that's a weighted combination of all other token representations, with greater weight given to the tokens deemed most relevant. In a classic transformer model, the attention mechanism isn't used just once; instead, it is used multiple times in parallel (see **Exhibit C5**). This is called multi-head attention. Each "head" operates independently and gets its own set of learnable parameters, which means that each head can learn to pay attention to different types of relationships between the words in the sequence.

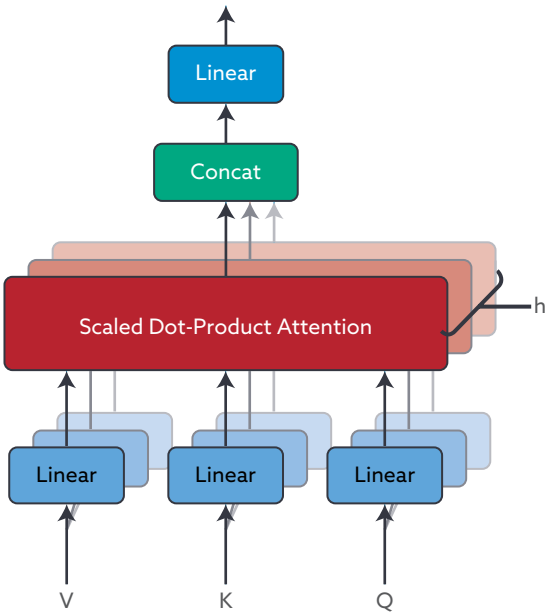
Let's take another look at our sentence, "The cat jumped from the ledge." One attention head might focus on syntactic relationships and notice the connection between "cat" and "jump," capturing the subject–verb relation, and another might focus on semantic relationships and notice the connection between "jump" and "ledge," capturing the context between these words. By having multiple heads, the model can simultaneously explore different aspects of the relationships between words in the sentence.

Exhibit C4. Scaled Dot-Product Attention



Source: Vaswani et al. (2017).

Exhibit C5. Multi-Head Attention Consists of Several Attention Layers Running in Parallel



Source: Vaswani et al. (2017).

After all the heads have completed their attention operations, their output matrices are concatenated and multiplied by an additional weight matrix to transform the result back to the original dimension, creating a single output for each token. This weight matrix is the third set of learnable parameters. This procedure ensures that the outputs from different heads are suitably integrated and can be fed forward to the next layer of the model.

Lastly, the output from the multi-head attention layer goes through a second layer of a position-wise, feed-forward neural network. This process is essentially like sending each token through a two-layer traditional feed-forward neural network independently, with the same weights and biases being used for each token. These are the last set of learnable parameters in a transformer block. This layer captures nonlinearity patterns in the model by wrapping the neural network nodes in a rectified linear unit (ReLU) activation function. The ReLU function is a common nonlinear activation function used in neural networks that outputs the input directly if it is positive; otherwise, it outputs zero. It has become a default choice because of its simplicity and its ability to reduce issues related to the vanishing gradient problem.

This whole process, starting from input to the position-wise, feed-forward network, constitutes a single transformer block.[9] But in practice, multiple such transformer blocks are stacked on top of each other to form a deep transformer model, allowing it to capture even more complex relationships in the data.

Each token in the final transformer layer is linearly transformed to the dimensionality of the entire vocabulary and run through the softmax function, producing a probability distribution for each token in the vocabulary. This linear projection is the final set of learnable parameters of the model. These probabilities are used as the basis for the model's predictions. For example, in language generation tasks, the model may be tasked with predicting the next word in a sentence. It would do so by selecting the word from its vocabulary that has the highest associated probability.

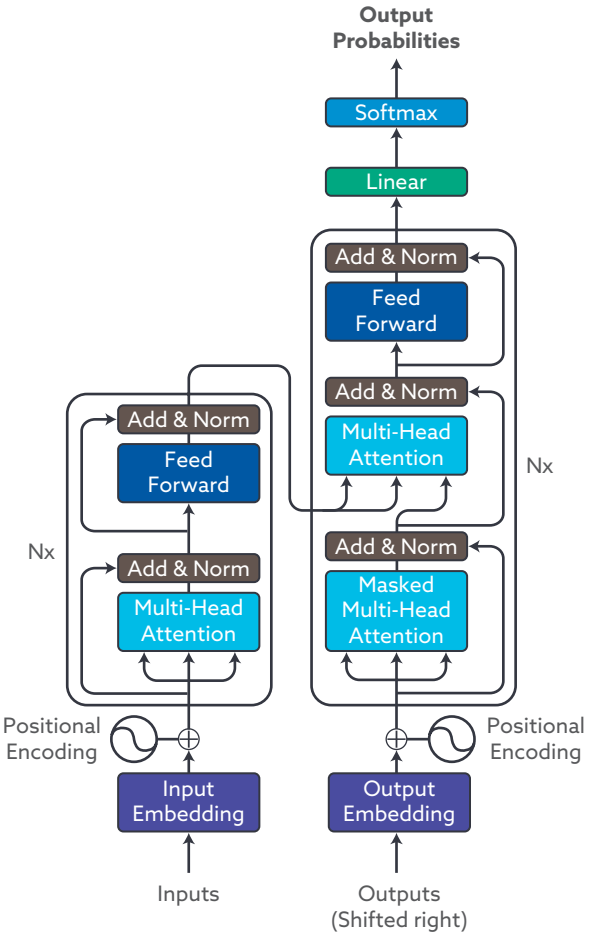
To optimize the model's parameters during training, we use a loss function that quantifies how far off the model's predictions are from the true values. A common loss function used in this setting is the cross-entropy loss, which measures the dissimilarity between the predicted probability distribution and the true distribution. The goal of training is to minimize this loss, which effectively maximizes the likelihood of the model correctly predicting the true word in the context. This is done using gradient descent optimization algorithms, which iteratively adjust the model's learnable parameters to reduce the loss. Through this process of prediction and learning from error, the transformer model is able to continually improve its performance on the task at hand.

One additional step is made in self-attention that will depend on the architecture used. As mentioned in chapter 2, the transformer model is composed of an encoder and a decoder block. **Exhibit C6** shows the transformer model architecture.

In the BERT architecture, the model is composed of only the encoder block. This architecture is different from GPT, which is composed of only the decoder block. With GPT, our prediction task is autoregressive, meaning we are predicting the next words given the previous words. To ensure the model is trained without a look-ahead bias (i.e., seeing the future words before they are present), an additional step is made in self-attention to mask the left-forward words. This step ensures that while training, GPT cannot cheat by relying on future context, maintaining the autoregressive nature of its predictions.

[9]An additional step is made after each sublayer (multi-headed attention, feed-forward). In this step, each sublayer output is normalized and the original input to the transformer block is added back to the output vector in a process called residual connection. This step helps in training by stabilizing the output distribution and allowing the gradients to flow to deeper layers directly, thereby mitigating the vanishing gradient problem.

Exhibit C6. The Transformer Model Architecture



Source: Vaswani et al. (2017).

APPENDIX D. ESG KEYWORD HASHTAG LIST

General ESG	SASB Label	General ESG	SASB Label
#airquality	Air Quality	#ClimateWeek	GHG Emissions
#AirPollution	Air Quality	#SDG	GHG Emissions
#cleanair	Air Quality	#carboncapture	GHG Emissions
#AirAware	Air Quality	#ClimateChange	GHG Emissions
#GoodGrowth	Business Ethics	#CarbonNeutral	GHG Emissions
#ethical	Business Ethics	#worldgreenbuildingweek	GHG Emissions
#infosecurity	Customer Privacy	#gogreen	GHG Emissions
#dataprivacyregulations	Customer Privacy	#chemistsinventgreen	GHG Emissions
#foodsafety	Customer Welfare	#EmbracingSustainability	GHG Emissions
#publichealth	Customer Welfare	#GHGemissions	GHG Emissions
#datasecurity	Data Security	#co2	GHG Emissions
#Cybersecurity	Data Security	#GHGreduction	GHG Emissions
#dataprotection	Data Security	#carbonoffset	GHG Emissions
#informationsecurity	Data Security	#zerogreenhousegasemissions	GHG Emissions
#dataprivacy	Data Security	#ClimateCrisis	GHG Emissions
#cloudsecurity	Data Security	#LowCarbonEconomy	GHG Emissions
#databreach	Data Security	#CarbonFootprint	GHG Emissions
#ransomware	Data Security	#CarbonSequestration	GHG Emissions
#computersecurity	Data Security	#NetZero	GHG Emissions
#securityawareness	Data Security	#emissions	GHG Emissions
#CircularEconomy	Ecological Impacts	#SocialImpact	Human Rights & Community Relations
#environment	Ecological Impacts	#corporateresponsibility	Human Rights & Community Relations
#pollution	Ecological Impacts	#improvingtheworld	Human Rights & Community Relations
#reforestation	Ecological Impacts	#stewardship	Human Rights & Community Relations
#ecologicalimpact	Ecological Impacts	#TradeFair	Human Rights & Community Relations

© 2024 CFA Institute. All rights reserved.

General ESG	SASB Label	General ESG	SASB Label
#DiversityandInclusion	Employee Engagement, Diversity & Inclusion	#fairtrade	Human Rights & Community Relations
#GenderEquality	Employee Engagement, Diversity & Inclusion	#Humanrights	Human Rights & Community Relations
#womeninSTEM	Employee Engagement, Diversity & Inclusion	#righttofood	Human Rights & Community Relations
#womenofchemisty	Employee Engagement, Diversity & Inclusion	#SocialJustice	Human Rights & Community Relations
#HeForShe	Employee Engagement, Diversity & Inclusion	#corporatecitizenship	Human Rights & Community Relations
#LesbianRights	Employee Engagement, Diversity & Inclusion	#sustainability	Physical Impacts of Climate Change
#UnionOfEquality	Employee Engagement, Diversity & Inclusion	#buildAfuture	Physical Impacts of Climate Change
#EmpoweringWomen	Employee Engagement, Diversity & Inclusion	#UNSDG	Physical Impacts of Climate Change
#womenintech	Employee Engagement, Diversity & Inclusion	#SASB	Physical Impacts of Climate Change
#SupportedEmployment	Employee Engagement, Diversity & Inclusion	#ecofriendly	Product Design & Lifecycle Management
#Deafawareness	Employee Engagement, Diversity & Inclusion	#productsafety	Product Quality & Safety
#Inclusion	Employee Engagement, Diversity & Inclusion	#recycling	Waste & Hazardous Materials Management
#LGBT	Employee Engagement, Diversity & Inclusion	#reduced resource	Waste & Hazardous Materials Management
#Employee safety	Employee Health & Safety	#StopSewerSpills	Waste & Hazardous Materials Management
#healthyworkplace	Employee Health & Safety	#oilspill	Waste & Hazardous Materials Management
#wellbeingatwork	Employee Health & Safety	#zerowaste	Water & Wastewater Management
#Greenbuild	Energy Management	#SaveWater	Water & Wastewater Management
#greeninfrastructure	Energy Management	#WaterUse	Water & Wastewater Management
#greenerenergyfuture	Energy Management	#watermanagement	Water & Wastewater Management

General ESG	SASB Label	General ESG	SASB Label
#energyefficiency	Energy Management	#WorldWaterDay	Water & Wastewater Management
#renewableenergy	Energy Management	#Water4All	Water & Wastewater Management
#EnergyManagement	Energy Management	#WaterManagement	Water & Wastewater Management
#EnergyAccessibility	Energy Management	#wastewater	Water & Wastewater Management
#EnergyStorage	Energy Management		
#energypolicy	Energy Management		

REFERENCES

Araci, Dogu. 2019. "FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models" (27 August). https://arxiv.org/abs/1908.10063.

Belkada, Younes, Tim Dettmers, Artidoro Pagnoni, Sylvain Gugger, and Sourab Mangrulkar. 2023. "Making LLMs Even More Accessible with bitsandbytes, 4-Bit Quantization and QLoRA." *Hugging Face* (blog, 24 May). https://huggingface.co/blog/4bit-transformers-bitsandbytes.

Bos, Jeroen. 2017. "Sustainability Scores for Investment Funds." *CFA Institute Magazine* (March). www.cfainstitute.org/-/media/documents/article/cfa-magazine/2017/cfm-v28-n1-13.ashx.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. "Language Models Are Few-Shot Learners." OpenAI (22 July). https://arxiv.org/pdf/2005.14165.pdf.

Cao, Larry. 2021. "T-Shaped Teams: Organizing to Adopt AI and Big Data at Investment Firms." CFA Institute (30 August). https://rpc.cfainstitute.org/-/media/documents/article/industry-research/t-shaped-teams.pdf.

CFA Institute. 2024. "Machine Learning." CFA Program Level II Refresher Reading. www.cfainstitute.org/en/membership/professional-development/refresher-readings/machine-learning.

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. "QLoRA: Efficient Finetuning of Quantized LLMs" (23 May). https://arxiv.org/pdf/2305.14314.pdf.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." Google AI Language (24 May). https://arxiv.org/pdf/1810.04805.pdf.

Fu, Xi, Xiaoxi Wu, and Zhifang Zhang. 2021. "The Information Role of Earnings Conference Call Tone: Evidence from Stock Price Crash Risk." *Journal of Business Ethics* 173: 643–60.

Gantz, John, and David Reinsel. 2012. "The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. IDC (December). www.cs.princeton.edu/courses/archive/spring13/cos598C/idc-the-digital-universe-in-2020.pdf.

Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80. www.bioinf.jku.at/publications/older/2604.pdf.

 © 2024 CFA Institute. All rights reserved.

Hong Kong Institute for Monetary and Financial Research. 2021. "Artificial Intelligence and Big Data in the Financial Services Industry" (October). www.aof.org.hk/docs/default-source/hkimr/applied-research-report/aibdrep4fa377c27c1649a6a219501ac8df30b7.pdf?sfvrsn=96c26d1d_6.

Hu, Edward, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. "LoRA: Low-Rank Adaptation of Large Language Models" (16 October). https://arxiv.org/pdf/2106.09685.pdf.

IDSO. 2019. "Web Crawling Best Practices." Document IDSO-WC-BP-001 (18 January). www.investmentdata.org/publications.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." https://arxiv.org/pdf/1310.4546.pdf.

Muscolino, Holly, Amy Machado, John Rydning, and Dan Vesset. 2023. "Untapped Value: What Every Executive Needs to Know about Unstructured Data." White paper, IDC (August). https://cloud.app.box.com/s/8jn28aniv99w59jlsh9t4wc473o55yuu.

OpenAI. 2022. "Introducing ChatGPT" (blog, 30 November). https://openai.com/blog/chatgpt.

Patel, Chirag. 2023. "Analyzing Sentiment in Quarterly Earnings Calls—Q3 2023." *S&P Global Market Intelligence* (blog, 26 September). www.spglobal.com/marketintelligence/en/news-insights/blog/analyzing-sentiment-in-quarterly-earnings-calls-q3-2023.

Patel, D., and A. Ahmad. 2023. "Google: 'We Have No Moat, and Neither Does OpenAI.'" SemiAnalysis. www.semianalysis.com/p/google-we-have-no-moat-and-neither.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. "GloVe: Global Vectors for Word Representation." Stanford University. https://nlp.stanford.edu/pubs/glove.pdf.

Pisaneschi, Brian. 2023. "Decoding the Crypto Mindset with NLP: Bitcoin, Reddit, and FTX." *Enterprising Investor* (blog), CFA Institute (17 February). https://blogs.cfainstitute.org/investor/2023/02/17/decoding-the-crypto-mindset-with-nlp-bitcoin-reddit-and-ftx/.

Preece, Rhodri G. 2022. "Ethics and Artificial Intelligence in Investment Management: A Framework for Professionals." CFA Institute (14 October). https://rpc.cfainstitute.org/en/research/reports/2022/ethics-and-artificial-intelligence-in-investment-management-a-framework-for-professionals.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. "Improving Language Understanding by Generative Pre-Training." OpenAI. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Růžičková, Lucie. 2022. "Developments in the hiQ v. LinkedIn Case." *Apify* (blog, 11 November). https://blog.apify.com/developments-in-hiq-v-linkedin-case/.

Schuster, M., and K. K. Paliwal. 1997. "Bidirectional Recurrent Neural Networks." *IEEE Transactions on Signal Processing* 45 (11): 2673–81. https://deeplearning.cs.cmu.edu/F20/document/readings/Bidirectional%20Recurrent%20Neural%20Networks.pdf.

Serafeim, G., and A. Yoon. 2022. "Which Corporate ESG News Does the Market React To?" *Financial Analysts Journal* 78 (1): 59–78.

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. "Alpaca: A Strong, Replicable Instruction-Following Model." Stanford Center for Research on Foundation Models. https://crfm.stanford.edu/2023/03/13/alpaca.html.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. "LLaMA: Open and Efficient Foundation Language Models." Meta AI. https://arxiv.org/pdf/2302.13971.pdf.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." Google, 31st Conference on Neural Information Processing Systems. https://arxiv.org/pdf/1706.03762.pdf.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2023. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." Google Research. https://arxiv.org/pdf/2201.11903.pdf.

Wigglesworth, R. 2023. "Markets Are Becoming Less Efficient, Not More, Says AQR's Clifford Asness." *Financial Times* (14 December).

Wu, Shijie, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. "BloombergGPT: A Large Language Model for Finance." https://arxiv.org/pdf/2303.17564.pdf.

Zhao, Frank. 2021. "U.S. Filings: No News Is Good News." S&P Global Market Intelligence (May). www.spglobal.com/marketintelligence/en/documents/us-filings-no-news-is-good-news.pdf.