Chapter 13
LLM Adoption Trends and Associated Risks



Zachary Schillaci

Abstract The emergence of *Large Language Models* (LLMs) is expected to impact the job market significantly, accelerating automation trends and posing a risk to traditionally creative-oriented jobs. LLMs can automate tasks in various fields, including design, journalism, and creative writing. Companies and public institutions can leverage generative models to enhance productivity and reduce workforce requirements through machine-assisted workflows and natural language interactions. While technical skills like programming may become less important in certain roles, generative models are unlikely to fully replace programmers due to the need for expertise in code validation and niche development. The enterprise landscape of LLMs comprises providers (organizations training proprietary models), integrators (technology companies fine-tuning LLMs for specific applications), and users (companies and individuals adopting LLM-powered solutions). The applications of the models include conversational search, customer service chatbots, content creation, personalized marketing, data analysis, and basic workflow automation. The regulatory landscape is rapidly evolving, with key considerations including copyright, data security, and liability. Government involvement and informed expertise are recommended to guide governance and decision-making processes in this domain.

13.1 Introduction

Generative AI technologies are currently undergoing a pronounced uptick in financial investment and integration, as significant capital flows are being channeled into both emerging startups and established enterprises alike [1]. As surveyed by McKinsey, research indicates that approximately one-third of organizations now utilize this technology in some capacity for at least one business function [2].

Z. Schillaci (✉)
Effixis SA, Lausanne, Switzerland
e-mail: zachary.schillaci@effixis.ch

© The Author(s) 2024

A. Kucharavy et al. (eds.), *Large Language Models in Cybersecurity*,
https://doi.org/10.1007/978-3-031-54827-7_13

Rather alarmingly, however, of these early adopters, only a minority of surveyed respondents express concern about generative AI-related risks and are actively establishing policies governing employees' use of the technology [2].

Recent advancements and developments in LLMs are the primary drivers behind the current takeoff of generative AI. However, even today's state-of-the-art LLMs are immune to the risks inherent in the technology. The continual and accelerated integration of LLM technology poses many risks for individual companies and the broader public. These risks arise from a convergence of factors, including the technology's limited maturity, its relative novelty to the general population, and its intrinsic complexity. Together, these elements create a complex landscape susceptible to inadvertent and deliberate harm or misuse. This section will discuss the latest adoption trends of LLMs, with a particular eye towards the associated risks and vulnerabilities, will be discussed in detail.

13.2 In-Context Learning vs Fine-Tuning

The potential for generalization within LLMs has been acknowledged by the *Natural Language Processing* (NLP) community for several years. While classical NLP approaches often divided the field into specialized verticals (e.g., parsing, semantics, coreference, translation, classification), pre-trained language models have introduced a horizontal foundational layer that can be applied to nearly any natural language task.

OpenAI researchers showcased the extraordinary "in-context learning" ability of GPT-3 in the pivotal paper "Language Models are Few-Shot Learners" [3]. This capability enables the model to solve various challenges by using just a handful of examples provided within its context window, a method referred to as "few-shot" learning. Notably, it only necessitates a single forward pass of the model, eliminating the need for intermediary gradient updates typical of conventional fine-tuning. Even more strikingly, the model can sometimes respond to queries based solely on an instruction in a single forward pass without any examples, a concept known as "zero-shot" learning. Presently, state-of-the-art instruction fine-tuned models excel at both few- and zero-shot learning, making them well-suited for many tasks. This adaptability is vital for practical applications, offering an alternative to traditional fine-tuning methods, which often demand extensive task-specific data and costly training iterations.

Thanks to recent innovations in efficient fine-tuning techniques—most notably LoRa [4], QLoRA [5], and PEFT [6]—fine-tuning LLMs has emerged as a viable option for certain use cases. In essence, these techniques allow one to adapt the capabilities of a base model for a particular task by training a relatively small number of parameters through gradient descent. Given access to sufficient, high-quality training data, fine-tuning a model can yield improved performance for specific tasks compared to relying only on in-context learning with the same model. Additionally, fine-tuning may considerably cut costs and reduce execution time,

as it often requires fewer, if any, tokens for instructions and examples in the prompt, unlike in-context learning. The reduction in execution time is especially noteworthy, considering that the self-attention mechanism—a central component of the transformer architecture—has a complexity that scales quadratically with the length of the input sequence.

Concerning the safety implications, fine-tuning poses a more pronounced risk of misaligning models and circumventing safety guardrails. A recent study demonstrated that fine-tuning a base model on only a handful of adversarially designed training examples can effectively bypass the model's safety alignment [7], which is not generally feasible with in-context learning or prompt engineering alone. As the researchers demonstrated, this technique even works with OpenAI's API for fine-tuning GPT-3.5 Turbo, despite the built-in safety mechanisms [7]. Furthermore, the research suggests that even non-adversarial fine-tuning with benign intentions can also inadvertently degrade the safety alignment of models [7]. The possibility of fine-tuning proprietary models directly from service providers, combined with the easy access to open-source models and training techniques on platforms such as HuggingFace, raises considerable concerns about the possibility of misuse. Though prompt engineering is far more accessible to the average user, the complexity of fine-tuning models is steadily decreasing with some services, like OpenAI's fine-tuning interface, offering what are essentially "no-code" solutions. This has effectively lowered the barrier to entry from machine learning engineers and data scientists to motivated individuals with only novice programming experience.

13.3 Adoption Trends

Since OpenAI's disruptive release of ChatGPT in November 2022, online chatbot services have dominated the public's use of LLMs. This includes notable platforms such as Microsoft's Bing Chat (based on ChatGPT), Google's Bard, and Anthropic's Claude. ChatGPT has become the fastest-growing consumer software application ever, amassing a staggering 100 million users within 2 months of its launch [8]. Though its adoption is beginning to slow, Similarweb ranks openai.com—the domain of ChatGPT—within the top 25 most visited websites globally as of the time of writing.

The leading LLM providers offer state-of-the-art, proprietary models generally capable of solving various NLP tasks. This encompasses everything from routine business operations to creative writing and software programming. These models are also made accessible as an API for software development purposes, albeit in a limited capacity. In either scenario, however, end users need the underlying model, limiting customization possibilities. Furthermore, transmitting sensitive data to external services via a web application or an API renders these solutions unsuitable for more security-conscious sectors. In this context, similar options exist to enable local hosting and serving of LLMs in a manner akin to the major cloud providers.

The following discussion, therefore, pertains to both cloud-based and locally-hosted solutions.

With the release of ChatGPT and similar services, many enterprises and individuals have quickly realized the power of general-purpose LLMs in assisting with various day-to-day tasks. For some, ChatGPT has become an essential tool in copywriting or software programming. Indeed, research suggests that posts to the popular software forum Stack Overflow have dropped dramatically since the release of ChatGPT, implying that many are turning to LLMs as a first resort for solving programming issues [9].

McKinsey reports that most organizations are using generative AI tools for marketing and sales, product and service development, and service operations, of which the common tasks involve crafting first drafts, summarizing documents, identifying trends in data, and interacting with chatbots [2]. Another application that's gaining traction to solve the problem of knowledge management is retrieval-augmented generation. This method involves linking a LLM with an external knowledge source, typically a vector database consisting of semantic embeddings[1]—such as those produced by Sentence Transformer models [10]. This connection enables the LLM to access an up-to-date, specialized knowledge repository and can drastically reduce the likelihood of hallucinations. Such applications are relatively easy to set up with off-the-shelf components and provide a straightforward means to customize an LLM for specific domains without the necessity of fine-tuning. Lastly, so-called agents are perhaps the most exciting and worrying usage of LLMs, wherein language models are connected to external tools and APIs to answer a wide range of user questions and instructions autonomously. AutoGPT is perhaps the most well-known example of an agent-oriented LLM framework, which describes itself as an "experimental open-source attempt to make GPT-4 fully autonomous" [11] and emerged as a top-trending repository on GitHub upon its release.

The emerging applications of LLMs are multifaceted and continuously evolving. The following presents a non-exhaustive list of the primary areas driving the technology's adoption:

- **Classical NLP Tasks**: This encompasses a wide variety of natural language processing functions such as question-answering, summarization, *Named Entity Recognition* (NER), text classification, machine translation, sentiment analysis, and text generation. Prior to today's state-of-the-art language models, handling each specific task would typically require a dedicated model trained on a specific dataset of examples. However, a single state-of-the-art LLM can now solve many of these problems through much more accessible prompting techniques.

[1] An embedding is a high-dimensional numeric vector representation that encapsulates the core attributes of an input, such that "similar" inputs tend to have closely situated embeddings. These embeddings are commonly obtained using deep learning methods and are widespread in computer vision and natural language processing. In NLP, vector embeddings can convey the semantic significance of words, sentences, or extended texts.

- **Programming**: This category includes tasks like code generation, code infilling (as with GitHub Copilot), static code analysis, documentation writing, and converting text into query languages (e.g., text-to-SQL).
- **Knowledge Management and Information Retrieval**: This consists of connecting a domain- or company-specific knowledge base to an LLM, usually via a retrieval-augmented generation pipeline, to provide a store of up-to-date, grounded information. Relevant entries in the knowledge base are passed as context along with user questions to guide model generation and prevent hallucinations.
- **Chatbots**: Personalized chatbots are being adopted for various needs, including customer support, internal knowledge management, document question-answering, and more. This also covers online services from direct providers such as OpenAI's ChatGPT and Google's Bard.
- **Agents**: Semi-autonomous LLM agents with integrated tool functionality and API connectivity. Though far from fully functional, these infant technologies pose a risk for harm and misuse, which will be discussed in the following section. Agents may or may not be configured as chatbots.

Through all of these use cases, a common LLM application framework is beginning to emerge in the field based primarily on in-context learning and, to a lesser extent, fine-tuning. While the landscape is still undergoing rapid change, certain patterns are beginning to become key components in the new field of *LLM Operations* (LLMOps). These elements encompass data ingestion pipelines, data transformation through embedding models, data indexing with vector databases, orchestration using frameworks like LangChain [13] and LlamaIndex [14], and traditional software methods, such as caching, logging, and validation, that have been adapted specifically for LLM applications [12].

13.3.1 LLM Agents

One of the more surprises of recent LLM adoption is the trend towards function calling and tool utilization. Just as the unanticipated programming capabilities of GPT-2 astounded researchers at OpenAI, the robust API calling competencies of contemporary LLMs surprised many in the field. Upon reflection, this progression seems predictable, given that the training sets of modern foundation models likely encompass all publicly available API documentation from the data collection period.

Shortly after the introduction of ChatGPT, users discovered the potential of using LLMs within an agentic framework, granting the model access to one or more functions, typically linked to APIs. This had previously been studied with the ReAct [15] approach, whereby structured reasoning and action of an LLM is accomplished through careful prompt engineering. Open-source frameworks such as LangChain [13] and HuggingFace Agents [16] were quick to integrate these approaches for mass use.

Following the success of these techniques, others ventured further by fine-tuning base LLMs on the downstream task of tool use and API calling, resulting in innovations such as Meta's Toolformer [17] and Microsoft's Gorilla LLM [18]. OpenAI, in turn responded with the release of ChatGPT plugins, augmenting the base capabilities of ChatGPT with an array of external APIs and, several months later, an update to its developer-facing API, exposing the fine-tuned plugin models for general use. OpenAI has also released a dedicated agent called Code Interpreter—available to ChatGPT Plus users—that accepts file uploads and has access to a restricted Python environment, allowing it to solve mathematical problems, conduct data analysis, and generate plots. Recent updates to ChatGPT Plus have built upon the agent paradigm, enabling Code Interpreter, Bing web browsing, and DALL-E image generation as default capabilities for ChatGPT-4.

13.4 Potential Risks

The *Open Worldwide Application Security Project* (OWASP)—a key organization in the open-source software security community—has outlined a comprehensive list detailing the principal vulnerabilities of LLM-powered applications [19]. These range from prompt injection attacks, where crafty inputs can manipulate the LLM, to supply chain vulnerabilities that can arise from using third-party components [19]. Prompt injection attacks are particularly pernicious because they can come from either a malicious user input, if direct access can be obtained, or a contaminated data source inserted into the prompt. The latter could be executed by, for example, embedding the attack within a document or website from which the LLM sources in a retrieval-augmented generation pipeline.

Other major concerns include insecure output handling, which can expose backend systems and enable remote code execution, training data poisoning that may introduce biases and security risks, and model theft, leading to loss of competitive advantage and sensitive information [19]. Insecure output handling is a pervasive issue in many LLM applications, given that it affects any solution that relies on using model output to initiate actions, execute code, or modify data. Building a robust post-processing pipeline to vet and sanitize LLM outputs against all potential threats is challenging, especially with the concomitant threat of prompt injection.

Furthermore, the OWASP outlines threats such as sensitive information disclosure, denial of service attacks, insecure design in plugins, excessive agency in LLM-based systems, and over-reliance on these models [19].

Agentic LLM systems are vulnerable to all these threats and, therefore, worthy of particular concern and scrutiny. Autonomous agents with high-level system privileges and broad tool access, such as AutoGPT [11] and its more malicious variants, have an even wider attack surface that makes responsible use effectively impossible. Granting an LLM, agentic or otherwise, access to arbitrary code execution in a public-facing application poses a major cybersecurity risk. The

widely popular framework LangChain [13] was, up until recently, susceptible to exactly this threat, as documented by the National Vulnerability Database (CVE-2023-29374) [20]. Other applications that directly execute LLM outputs as code, of which there are surely many in use now, are similarly affected. Stopping short of this, other applications using agentic LLMs can wreak havoc on systems if restricted permissions and environments are not properly implemented.

Beyond the security vulnerabilities inherent in the development of LLM applications, as outlined by OWASP, a separate class of threats stemming from deliberate misuse exists. Malicious entities may employ sophisticated prompt engineering or adversarial fine-tuning of models to achieve harmful objectives. Such risks are especially acute within the open-source domain, where actors can more easily bypass built-in safety filters and guardrails of aligned models. Currently, the options to deter such malevolent utilization are limited, and the technical barrier to entry into this domain is progressively diminishing.

These vulnerabilities and risks collectively highlight the need for preventative awareness campaigns, robust security measures, and careful oversight of open-source developments to prevent potential misuse, data breaches, and other severe consequences.

References

1. Goldman Sachs. Ai investment forecast to approach $200 billion globally by 2025, August 2023. Accessed: 2023-08-24.
2. McKinsey & Company. The state of ai in 2023: Generative ai's breakout year, August 2023. Accessed: 2023-08-24.
3. Tom B. Brown et al. Language models are few-shot learners, 2020.
4. Edward J. Hu et al. Lora: Low-rank adaptation of large language models, 2021.
5. Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.
6. Haokun Liu et al. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, 2022.
7. Xiangyu Qi et al. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023.
8. Krystal Hu. ChatGPT sets record for fastest growing user base: Analyst note, 2023. Accessed: 2023-08-21.
9. Maria del Rio-Chanona, Nadzeya Laurentsyeva, and Johannes Wachs. Are large language models a threat to digital public goods? evidence from activity on stack overflow, 2023.
10. Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.
11. Significant Gravitas. Auto-gpt, 2023. An experimental open-source attempt to make GPT-4 fully autonomous.
12. Matt Bornstein and Rajko Radovanovic. Emerging architectures for llm applications, 2023. Accessed: 2023-08-17.
13. Harrison Chase. Langchain, 10 2022. If you use this software, please cite it as below.
14. Jerry Liu. Llamaindex, 11 2022. If you use this software, please cite it as below.
15. Shunyu Yao et al. React: Synergizing reasoning and acting in language models, 2023.

16. Thomas Wolf et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 10 2020. Association for Computational Linguistics. Software available at https://github.com/huggingface/transformers.
17. Timo Schick et al. Toolformer: Language models can teach themselves to use tools, 2023.
18. Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023.
19. OWASP. OWASP Top Ten for Large Language Model Applications, 2013. Accessed: 2023-08-17.
20. National Vulnerability Database. CVE-2023-29374, 2023. 28.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

