Journal of Artificial Intelligence Research 80 (2024) 1497-1523 Submitted 01/2024; published 08/2024

Language-Models-as-a-Service: Overview of a New Paradigm and its Challenges

Emanuele La Malfa EMANUELE.LAMALFA@CS.OX.AC.UK
Department of Computer Science, University of Oxford
Oxford, OX1 3QG, UK
The Alan Turing Institute
London, NW1 2DB, UK

Aleksandar Petrov ALEKSANDAR.PETROV@ENG.OX.AC.UK
Department of Engineering, University of Oxford
Oxford, OX1 3PJ, UK

Simon Frieder SIMON.FRIEDER@CS.OX.AC.UK
Department of Computer Science, University of Oxford
Oxford, OX1 3QG, UK
Faculty of Informatics, Vienna University of Technology
Vienna 1040, Austria

Christoph Weinhuber CHRISTOPH.WEINHUBER@CS.OX.AC.UK
Department of Computer Science, University of Oxford
Oxford, OX1 3QG, UK

Ryan Burnell RBURNELL@TURING.AC.UK
The Alan Turing Institute
London, NW1 2DB, UK

Raza Nazar RAZANAZAR1@GMAIL.COM
Faculty of Law, University of Oxford
Oxford, OX1 3UL, UK

Anthony G. Cohn A.G.COHN@LEEDS.AC.UK
School of Computing, University of Leeds
Leeds, LS2 9JT, UK
The Alan Turing Institute
London, NW1 2DB, UK

Nigel Shadbolt NIGEL.SHADBOLT@CS.OX.AC.UK
Michael Wooldridge MICHAEL.WOOLDRIDGE@CS.OX.AC.UK
Department of Computer Science, University of Oxford
Oxford, OX1 3QG, UK
The Alan Turing Institute
London, NW1 2DB, UK

Abstract

Some of the most powerful language models currently are proprietary systems, accessible only via (typically restrictive) web or software programming interfaces. This is the Language-Models-as-a-Service (LMaaS) paradigm. In contrast with scenarios where full model access is available, as in the case of open-source models, such closed-off language models present specific challenges for evaluating, benchmarking, and testing them. This paper has two goals: on the one hand, we delineate how the aforementioned challenges act as impediments to the accessibility, reproducibility, reliability, and trustworthiness of LMaaS. We systematically examine the issues that arise from a lack of information about language models for each of these four aspects. We conduct a detailed analysis of existing solutions, put forth a number of recommendations, and highlight directions for future advancements. On the other hand, it serves as a synthesized overview of the licences and capabilities of the most popular LMaaS.

©- The Authors. Published by AI Access Foundation under Creative Commons Attribution License CC BY 4.0.

1. Introduction

The field of natural language processing (NLP) has undergone a profound transformation in the past few years, with the advent of (Transformer-based) Language Models (LMs) (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2019). Improved access to large models has been a fundamental facilitator of progress (Wolf et al., 2019), fueled by the scale of data and computing available to research institutions and companies (Kaplan et al., 2020). In less than five years, the state-of-the-art models evolved from small architectures that were trainable on a few GPUs (Peters et al., 2018; Devlin et al., 2019) to very large and complex architectures that require dedicated data centres and supercomputers (Raffel et al., 2020; Rae et al., 2021) which are very costly to set up. Commercial incentives have led to the development of large, high-performance LMs, accessible exclusively as a service for customers that return strings or tokens in response to a user's textual input – but for which information on architecture, implementation, training procedure, or training data is not available, nor is the ability to inspect or modify its internal states.

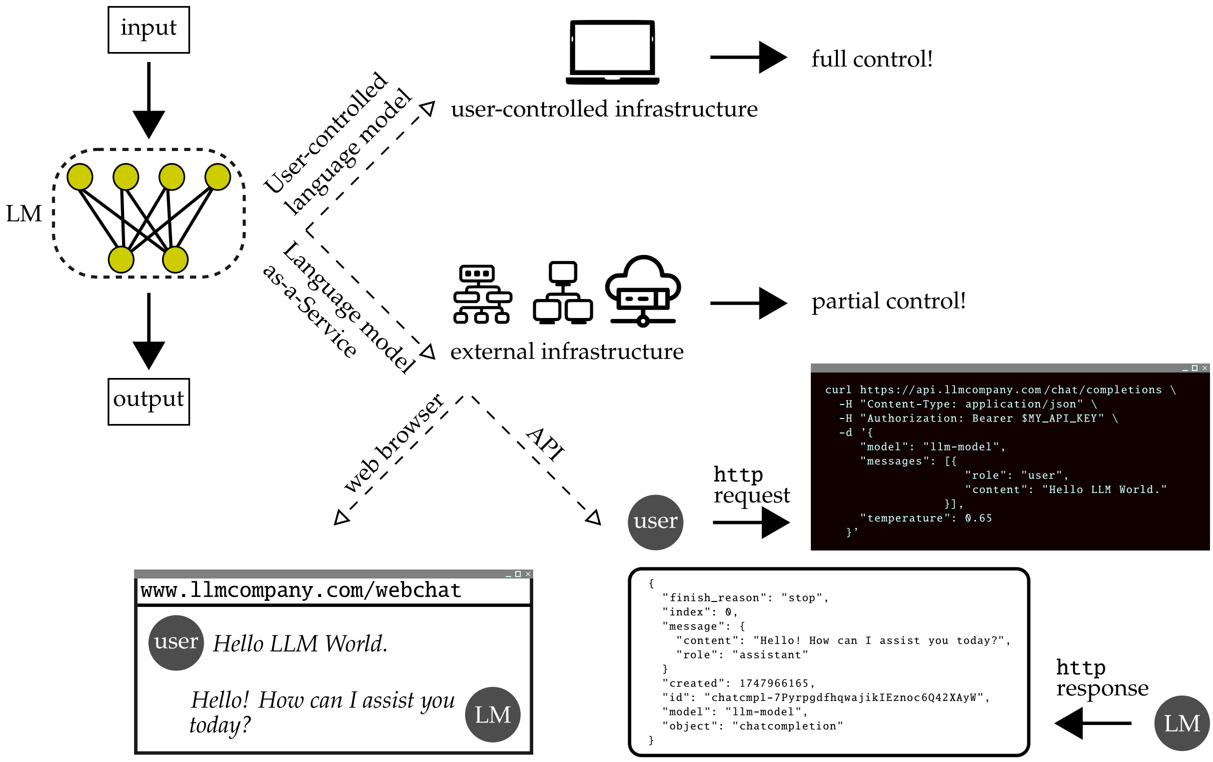
This paradigm, known as *Language-Models-as-a-Service* (LMaaS) (Sun et al., 2022), represents a licensing model in which LMs are centrally hosted and, typically, provided on a subscription or pay-per-use basis. Modern LMaaS provide a unified portal for various services that had previously been separated; from access to information that was the realm of search engines to problem-solving tools on a large number of domains such as data analysis, image generation, etc. (Romera-Paredes and Torr, 2015; Brown et al., 2020; Lewis et al., 2020; OpenAI, 2023). These services have grown rapidly and are now adopted extensively by hundreds of millions of customers. In parallel to improving models' capabilities, the risk of malicious usage is also increasing, e.g., regarding the weaponization of biotechnologies and mass surveillance (Hendrycks et al., 2023). Other recent works highlight the risks of LMaaS and LMs that are not aligned with human values (Bommasani et al., 2021). This has resulted in an explosion of interest in understanding values and biases encoded by these models, intending to align the former to those of humans and mitigate the latter (Ganguli et al., 2022; Liu et al., 2022; Santurkar et al., 2023).

However, access restrictions inherent to LMaaS, combined with their black-box nature, are at odds with the need of the public and the research community to understand, trust, and control them better, as illustrated in Figure 1. This causes a significant problem at the core of artificial intelligence: the most powerful and economically impactful, but at the same time most risky, models are also the most difficult to analyze. LMs are released with various licences, from open-source to more restrictive cases such as "open-weights".[1] Yet, once obtained, LMs are inspectable, and the end-user can flexibly control their behaviour. On the other hand, LMaaS are accompanied by commercial licences, are mostly closed-source and partially controllable by the end-user, and are often accessible via paid subscriptions. Thousands of applications are currently powered by LMaaS,[2] whose internal pipelines depend on the output (or the internal representations) of generative models whose internal mechanisms are kept undisclosed. In this sense, trusting LMaaS is a necessary, yet not sufficient, condition to trust the products they boost, both for the end-user and the companies that adopt them. While some of these problems are orthogonal to the existing concerns with LMs, in general, the particularities of LMaaS exacerbate these issues or make their assessment or mitigation significantly more difficult. We group the difficulties arising from such paradigm along four categories, namely *accessibility*, *reproducibility*, *reliability*, and *trustworthiness*.

- **Accessibility - Section 3.1.** LMaaS are frequently accessible through application programming interfaces (API) or web interfaces, with free, pay-per-use, or subscription-based payment modalities. To use them, one must accept and subscribe to commercial

[1] `https://github.com/Open-Weights/Definition`, accessed on 19/06/2024.

[2] `https://openai.com/index/gpt-3-apps/`, accessed on 29/12/2023.



```
curl https://api.llmcompany.com/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $MY_API_KEY" \
  -d '{
     "model": "llm-model",
     "messages": [{
                    "role": "user",
                    "content": "Hello LLM World."
                 }],
     "temperature": 0.65
   }'
```

```
{
  "finish_reason": "stop",
  "index": 0,
  "message": {
    "content": "Hello! How can I assist you today?",
    "role": "assistant"
  }
  "created": 1747966165,
  "id": "chatcmpl-7PyrpgdfhqwajikIEznoc6Q42XAyW",
  "model": "llm-model",
  "object": "chatcompletion"
}
```

www.llmcompany.com/webchat

user *Hello LLM World.*

Hello! How can I assist you today? LM

Figure 1: Illustration of the difference between interacting with LMaaS and user-controlled LMs. Most LMs that are offered in full provide access to a model's internals (e.g., its weights) and list details on the training procedure and instructions on executing the models locally. In most cases, this allows users to inspect the inner workings of these models, and run (or change) them on the hardware of their choice. On the other hand, LMaaS are accessible through a web interface or an API only (for illustrative purposes, in the diagram above, the API calls resemble those of OpenAI). They are powered by LMs that typically run on computational infrastructures controlled by third parties, and little information about the model is exposed to the user, beyond general reports or model cards (Wolf et al., 2019).

licenses that grant companies the right to collect and use prompts to improve their model. LMs, despite some notable exceptions that we touch on in the first Section of this paper, are largely accessible and often come with permissive licenses. Furthermore, LMaaS access costs are often not in harmony with the socioeconomic factors of prospective users, potentially resulting in disadvantages in certain demographic layers.

- **Reproducibility - Section 3.2.** LMaaS are deployed and updated in a continuous delivery/deployment regime, with legacy models frequently replaced by newer ones and depreciated altogether, possibly with little prior notice. This undermines reproducibility as one cannot evaluate withdrawn models or compare different versions. Reproducibility is further limited by the intrinsic non-determinism of LMaaS and the limited configuration options the service providers offer.

- **Reliability - Section 3.3.** Benchmarking the LMs' and LMaaS' performance on various tasks and problems is how we ensure models are reliable. Benchmarking any LM incurs significant computational and human costs and is non-trivial to carry out, but for LMaaS, compared to user-controlled LMs, additional challenges occur, such as *dataset* and *user contamination* (i.e., it is hard to devise samples already not digested by the model and out-of-distribution test sets) and *evaluating emergence* (i.e., attributing the origin of certain allegedly emergent abilities of an LMs and LMaaS).

- **Trustworthiness - Section 3.4.** Models whose decision process is inspectable and interpretable in terms of elementary input-output operations are often denoted as *self-explaining*. LMaaS do not belong to that class but are rather *explanatory* techniques, with their explanations coming from conditioned prompts, which are handled the same way as any other query. In this sense, any explainability technique that requires access to a model's internal does not apply to them.

These problem categories show that LMaaS require careful treatment due to their Software-as-a-Service nature (SaaS), an appropriate regulatory framework, and policies implemented by the companies that provide access to such services. We conclude the paper with Section 4, which outlines some strategies to mitigate the most urgent issues raised by LMaaS, which we hope will help the research community and companies to make Language-Models-as-a-Service more accessible, reproducible, reliable, and trustworthy. We emphasise the complementary role companies and the research community play to ensure the above values are instilled and preserved.

2. Related Work

Several factors have underlined the success of LMs and the consequent commercial interest and advent of the LMaaS paradigm. In this section, we provide a brief review of these developments.

Language Models. Many computational models and approaches towards natural language understanding have been developed over the last several decades (Goldberg, 2016). Until recently, though, their relatively poor performance limited their commercial viability. However, the introduction of the Transformer architecture (Vaswani et al., 2017), inspired by previous works on attention mechanisms (Bahdanau et al., 2015; Niu et al., 2021), led to more efficient training of deep-learning-based LMs, which could leverage large scale text data for training. Consequently, the performance of Transformer-based LMs quickly surpassed all previous methods (Devlin et al., 2019). Large Transformer-based models, of which LMs are the most successful implementation, also exhibit zero- and few-shot behaviours, i.e., the ability to solve novel tasks with zero or few examples provided as part of the input prompt (Chang et al., 2008; Brown et al., 2020). These behaviours scale with the size of the models (Kaplan et al., 2020), which led to the development of even larger models. This scaling was also supported by the collection of massive training datasets (Shanahan, 2022; Zhao et al., 2023), the development of novel training and fine-tuning techniques (Houlsby et al., 2019; Hu et al., 2021; Bai et al., 2022), and improvements in computing hardware.

Parallel to that, the research community has focused on evaluating LMs and their zero- and few-shot behavior (Xian et al., 2018; Chang et al., 2023) as well their capability (and limitations) to solve tasks that require compositional reasoning (Dziri et al., 2023; McCoy et al., 2023b). For example, recent works have proposed techniques to test the capabilities on unseen data points, like suites for dynamic benchmarks, behavioural testing, and out-of-domain analysis (Ribeiro et al., 2020; Kiela et al., 2021; Zhou et al., 2022). Despite their impressive performances, state-of-the-art LMs still struggle to solve the most challenging cases of low-order tasks such as sentiment analysis (Barnes et al., 2019; Barnes, 2021; Malfa and Kwiatkowska, 2022) or math (Frieder et al., 2023). LMs have also been shown to lack robustness: they may respond incorrectly to minor variations of inputs that a model has correctly classified (Sinha et al., 2021; Wang et al., 2023a).

Language-Models-as-a-Service. The first work we are aware of to use the term *Language-Model-as-a-Service* is by Zhao et al. (2021), despite many publications (Deng et al., 2022; Ding et al., 2022; Dong et al., 2022) citing the later work by Sun et al. (2022). LMaaS came to prominence with the advent of ChatGPT (OpenAI, 2023a) and other products developed by Google and Microsoft (Thoppilan et al., 2022),[3] though breakthroughs and key observations that contributed to their success date back to LMs such as GPT-3 (Brown et al., 2020). Recently, a lot of research has focused on evaluating the performance of LMaaS on consolidated NLP datasets and benchmarks (Liang et al., 2022; Chang et al., 2023; Laskar et al., 2023), as well as on specialized tasks such as mathematical and spatial reasoning, symbols manipulation, and code generation (Chen et al., 2019; Kojima et al., 2022; Cohn and Hernandez-Orallo, 2023; Frieder et al., 2023; Ray, 2023; Rozière et al., 2023; Shen et al., 2023). These evaluations continually evolve and improve, representing a crucial component in advancing the state-of-the-art in NLP (Wang et al., 2023a; Zhao et al., 2023). LMaaS also achieve super-human performance on a variety of tasks (Bubeck et al., 2023), yet fail on edge-cases that humans correctly classify (Berglund et al., 2023; Hao et al., 2023; Kocoń et al., 2023), exhibit implicit linguistic biases (Ahia et al., 2023; Bang et al., 2023; Petrov et al., 2023b) or are brittle to adversarial prompts (Kocoń et al., 2023; Shen et al., 2023; Schlarmann and Hein, 2023; Zou et al., 2023). They also do not model uncertainty or treat ambiguity properly (Liu et al., 2023a). We note that since the best-performing language models are the ones available solely as-a-Service (OpenAI, 2023), many investigations concerning advanced capabilities of LMs cannot be decoupled from the as-a-Service platform via which the LMs are offered (Ray, 2023; Shen et al., 2023).

3. The LMaaS Paradigm

We provide a high-level definition of LMs, which we then restrict to that of LMaaS. We then discuss how the LMaaS differs from LMs in four crucial respects, namely accessibility, reproducibility, comparability, and trustworthiness.

A language model defines a probability distribution over a finite string of tokens (Du et al., 2022) and is trained to predict a symbol or token (an instance of a sequence of characters) from a finite vocabulary. Instead of computing the token with the highest probability over the next in a sequence, most LMs introduce diversity by employing non-deterministic sampling strategies (Holtzman et al., 2020). While more capable than the deterministic counterparts, the behaviour of non-deterministic models varies according to parameters such as the *temperature* or the *seed*. The former is expected to make the generative process deterministic. At the same time, with the latter, we refer to a setting where all the sources of uncertainty, including the generative process, are deterministic and thus reproducible.

LMs are in-context learners (Brown et al., 2020; Dong et al., 2022), which refers to their ability to solve a task without changing model weights, thereby being an optimal learning approach for models provided as-a-Service through APIs or a web interface, as illustrated in Figure 1. In their simplest form, LMaaS return utterances in response to a user's prompt, both provided textually through the mentioned means of interactions. This form of interaction removes many of the possibilities of changing the models' behaviour, although commercial offerings allow some level of fine-tuning models (OpenAI, 2023b).

3.1 Accessibility

Licenses are (legal) instruments that accompany most LMs and LMaaS and govern the use and distribution of software. They define how the software can be used, modified, and distributed and outline the rights and responsibilities of the user and the software provider. Open-source and free software are widely spread philosophies that promote transparency, customizability, and community-driven development, which can enhance or limit the accessibility of software products. The landscape of LMs' licenses is varied, as discussed in the next section and shown in Figure 2; on the other hand, for LMaaS, companies mostly employ commercial licenses due

[3] `https://www.microsoft.com/bing`, accessed on 26/09/2023.

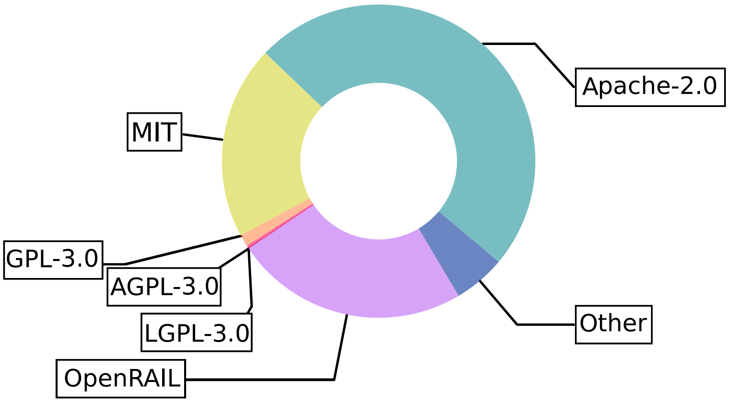


Figure 2: LMs, sourced from the https://huggingface.co/models library, grouped by how they are provided regarding software licences. Half of the LMs on Huggingface are provided with an Apache-2.0 licence, followed by the MIT licence and the OpenRAIL licence, which is a new licence specifically devised for machine-learning applications. Last access 01/08/2023.

to their nature as-a-Service tools. Further, we illustrate how LMaaS, delivered as pay-per-use services, intensify disparities between high-resource (e.g., English) and low-resource languages on components such as the tokenization process before a model is prompted.

LMaaS and LMs licenses. While some LMs require massive computational capabilities to be instantiated locally (Rae et al., 2021; Chowdhery et al., 2022), most are freely available for download, inspection, and execution on middle- to low-end machines: an instance of Alpaca-7B (Taori et al., 2023) or LLaMA-7B (Touvron et al., 2023a,b) can be used to do inference in full-precision, on a single GPU with 28GB of RAM.[4] Furthermore, approaches such as Petal (Borzunov et al., 2022) allow users to access and fine-tune distributed LMs such as LLaMA and Falcon, leveraging a peer-to-peer network where one can contribute by sharing their computational power. LMs licences come in different shapes and forms, as reported in Figure 2. Most LMs come with an MIT or an Apache license, which allows users to use, copy, modify, distribute and sell copies of the software. We recall that the MIT license does not require the source code to be available when redistributing the software unless the derived code comes in turn with an MIT license. In contrast, Apache licenses must include a copy of the license and a list of any modifications made to the original software, whether the derived product is released with the source code or in binary form. Popular LMs that are licensed with an MIT or an Apache license are GPT-2 (Radford et al., 2019) and BERT (Devlin et al., 2019), respectively.

The (Open)RAIL family of licences[5] are a relatively recent licensing method developed to account for cases specific to AI-powered software. They allow free access and re-distribution of its licensed material or derivatives, as long as they credit OpenRAIL and license their new creations under identical terms. On the other hand, OpenRAIL licences specify use-based restrictions clauses to account for potential social costs stemming from harmful uses of openly licensed LMs. BLOOM is a popular example of an OpenRAIL licensed LM (Scao et al., 2022). Taking a broader perspective, even though LMs are released through popular libraries that allow users to share machine learning models, such as Huggingface (Wolf et al., 2019) or AllenNLP (Gardner et al., 2018), not all of them fall within the definition of open source or free software (Stallman, 2010, 2022). For example, models such as LLaMA are "available-weights",

[4] https://finbarr.ca/how-is-llama-cpp-possible/, accessed on 26/09/2023.
[5] https://www.licenses.ai/ai-licenses

in the sense that one requests access to the model and is then granted access to the weights,[6] which come with custom licenses that limit their usability. The LLaMA license, in particular, forbids its users to use its output to train other models.[7]

LMaaS, and in general Software-as-a-Service (SaaS), move the computation burden to the provider's servers though at the cost of the users no longer having direct control over the software. With capabilities that often largely surpass that of LMs (Liu et al., 2023b; OpenAI, 2023), LMaaS have become a tool used daily by users and researchers. As previously mentioned, free software is software users can run, copy, distribute, study, change, and improve (Stallman, 2010) with some reasonable limitations, e.g., responsible use for OpenRAIL. LMaaS violate some or all of these principles, depending on who is offering the service, as most come with commercial licences (Liesenfeld et al., 2023), and cannot be run locally. The options are either paying an API provider or using a *freemium* service, i.e., free for basic usage with the possibility to upgrade it with a paid subscription, offered through a web interface.[8] For models such as Google Bard (Thoppilan et al., 2022) and Microsoft Copilot in Bing, prompting through an API is not supported natively at the time of writing (see Table 1), yet workarounds exist: despite wide adoption, such solutions create problems regarding the liability of the software and the reliability of the results, as they might break the commercial licence under which the end user is expected to use the service. Consequently, copying, distributing, fully studying, and changing most commercial LMaaS is impossible.

Aligning the principles of free software with the procedures and practices of LMaaS involves, at minimum, releasing their source code and providing training instructions. A few institutions and companies have released LMs, such as Alpaca and BLOOM, that implement these practices (Scao et al., 2022; Taori et al., 2023), and part of the research community has committed to developing LMs one can download and use locally, with techniques such as LoRA and its variants (Hu et al., 2021; Dettmers et al., 2023), that allow fine-tuning and, to some extent, training models that otherwise require access to large scale computational facilities.

On the other hand, some others have faced criticisms, as in the previously discussed case of LLaMA.[9] However, in commercial products released as-a-Service, licences are more restrictive, as companies prefer to avoid open software methodologies (Liesenfeld et al., 2023) to protect, via *security through obscurity* their intellectual property and the derived competitive advantages. With the computational power distributed unevenly and concentrated in a tiny number of companies, those with a technological, yet not computational, advantage face a dilemma. While open-sourcing their LMaaS would benefit them in terms of market exposure and contribution to their codebase by the community, releasing the code that powers a model may rapidly burn their competitive advantage in favour of players with higher computational resources (Henkel, 2009; Heron et al., 2013; Tkacz, 2020).

Access barriers to LMaaS. AI-powered solutions such as LMaaS will drive significant economic growth in the forthcoming years. For instance, the UK government has projected that AI could contribute to a 10% increase in GDP.[10] Given the substantial economic implications, ensuring widespread access to these models and services becomes paramount. Nevertheless, the prevalent practice of uniform pricing poses a barrier, particularly limiting accessibility for individuals and organizations in underdeveloped and developing areas of the world. Consequently, the paid models offered by LLMaaS may result in considerable disparities in the

[6] `https://www.alessiofanelli.com/blog/llama2-isnt-open-source`, accessed on 10/11/2023.

[7] `https://ai.meta.com/resources/models-and-libraries/llama-downloads/` section v., accessed on 10/11/2023.

[8] Without even counting that different models may serve different users, in the same way as LMaaS served through the API potentially differs from that hosted through the web interface

[9] `https://opensourceconnections.com/blog/2023/07/19/is-llama-2-open-source-no-and-perhaps-we-need-a-new-definition-of-open/`, accessed on 15/10/2023.

[10] `https://assets.publishing.service.gov.uk/media/5ff3bc6e8fa8f53b76ccee23/AI_Council_AI_Roadmap.pdf`, accessed on 10/11/2023.

economic impacts of this technology worldwide (Zarifhonarvar, 2023). Moreover, due to commercial interests, developers of LMaaS tend to focus on affluent markets. This bias is evident in the performance disparities for less commonly used languages, as noted in various studies on tokenization (Ahia et al., 2023). Additionally, the pricing model, which charges per token or Unicode character, disproportionately affects certain languages (Petrov et al., 2023b). For instance, some languages incur costs up to fifteen times higher than English. Consequently, the exclusive and paid structure of LMaaS contradicts the assertion that language models can promote global welfare and reduce social inequalities. On the contrary, we believe they are more likely to aggravate these disparities.

A starting point to mitigate these issues is thus analyzing the impact of LMaaS and, more generally, pay-per-use artificial intelligence services as a standalone, pervasive, and disruptive technology. For works highlighting unfair premiums paid by low-represented groups, e.g., non-English speakers, we argue that the pricing would be better informed by taking into account the different groups' geographical and economic factors (for example, using informed indicators such as the monetary measure of the market value). Solving disparities in access to disruptive solutions such as LMaaS goes well beyond addressing their technological problems and requires implementing ad-hoc policies and governance tools.

3.2 Reproducibility

Reproducibility is a fundamental concept in science, serving as a cornerstone for establishing the reliability of findings. In machine learning, reproducibility refers to achieving the same results using the same dataset and algorithm. LMaaS hardly meet these conditions.

Their probabilistic nature, exacerbated by the access policies as-a-Service, the practice of deprecating or changing models seamlessly, without public notifications, and their intrinsic non-determinism, affect such models even when all the sources of randomness available to the end-user are fixed. Before delving into LMaaS's reproducibility problems, we discuss their uniqueness in the panorama of ML as-a-Service products. While one may argue most problems that affect LMaaS are shared by any ML as-a-Service tool, e.g., machine translation, object recognition, diffusion-based image generation, etc., unprecedented attention has been given by the research community to LMaaS-related issues. More than 15,000 articles were published in 2023 that include terms such as 'GPT-4' or 'GPT-3-5',[11] and most of top machine learning venues now dedicate tracks to 'generative models' (thus including LMs and LMaaS). We thus emphasise the necessity of developing frameworks and tools to analyse them. Reproducibility and determinism are two cornerstones of such an approach.

Reproducible experiments in light of LMaaS depreciation. In software development, continuous delivery is an approach where code changes are made to an already deployed application and released into the production environment. For SaaS solutions, this means that for a user who interacts with an LMaaS through a web browser GUI, a new version of a product can overwrite the preceding one seamlessly (Hacker News, 2023a,b; OpenAI Community, 2023). LMaaS, such as GPT-3.5, GPT-4, Bard, and Microsoft Copilot in Bing, are accessible via a GUI or APIs , though few details are publicly available on the exact deployment strategy. On one side, continuous delivery allows companies to deploy better models and rapidly patch bugs and newly discovered vulnerabilities; on the other, it harms the reproducibility of empirical analyses conducted on such models (Chen et al., 2023). For API access, one would expect clear documentation of model changes, but even in this case, reports of sudden changes in behaviour exist (Gao, 2023)

When a company deprecates a model, assessing and thus trusting the validity of an experiment depends only on the historical data and the consensus reached by the research community

[11] Source: Google Scholar; Query: `"GPT-4" | "GPT-3.5"`; accessed on 30/12/2023: https://scholar.google.com/scholar?q=%22GPT-4%22+%7C+%22GPT-3.5%22&hl=en&as_sdt=0%2C5&as_ylo=&as_yhi=2023

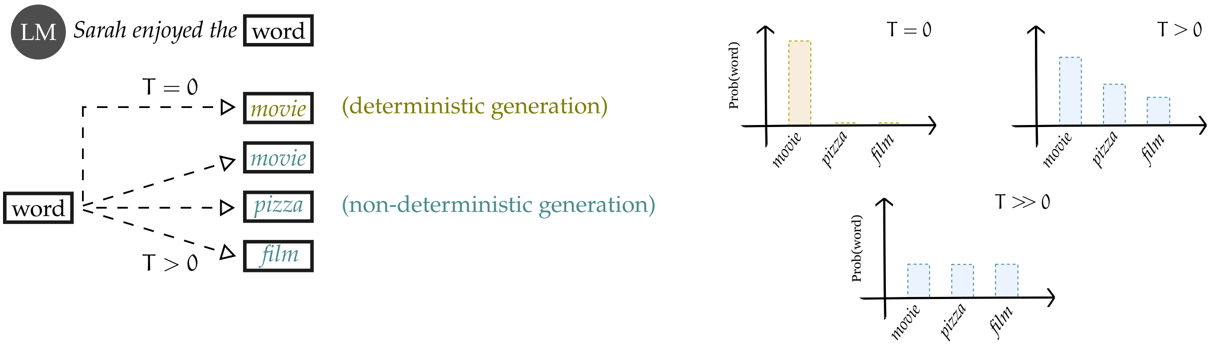


Figure 3: Setting the temperature T to zero makes (typically) LMaaS deterministic, with the probability of sampling the next word that concentrates on a single word. On the other hand, for T > 0, LMaaS become progressively more *creative*, but makes the distribution over which they sample progressively more flat.

on the reliability of the benchmarking technique. Assume that one measured, between April and May 2023, the performance of GPT-3.5 with the methodology described in (Liang et al., 2022). On the 11th of May 2023, OpenAI removed GPT-3.5-legacy from the list of available models and is adopting a similar policy for newer models.[12] Reproducing experiments for those models has become impossible; thus, we can only trust the correctness of the benchmarking techniques. On the contrary, most LMs that are not offered under the LMaaS paradigm, are stored by third-party services and made available for download with permissive licences, as discussed previously in this paper.

To check if an LMaaS has been updated, one can collect some answers in response to a set of prompts, store their hash, and compare it over time to prove that a model replacement has occurred. Unfortunately, this solution is not sound: two different LMaaS can output the same answer or differ not because one replaced the other but due to their intrinsic non-determinism (see next paragraph). Another viable solution would be to require companies to support access to legacy models for an extended period: this could happen, if not immediately after being withdrawn, after a period that would not preclude users from adopting newer versions. However, this is not without its downsides too: as newer versions patch safety and exploitation vulnerabilities (e.g., "jailbreaks" (Chao et al., 2023; Liu et al., 2023c; Yao et al., 2023a)), maintaining an "unpatched" version can thus be a method for malicious actors to continue exploiting the system.

These aspects are highly problematic, particularly when LMaaS are used as an intermediate node in a software pipeline for a downstream consumer software offering that builds on LMaaS: as LMaaS change, the quality of the downstream service will also change. One way to address these issues is to allow only access to vetted individuals and organizations, though that would limit the audit of the models. Alternatively, providers can supply an interface for researchers to register benchmarks that re-evaluate a model at every update, like public continuous integration testing. Since API change is not documented well presently (Gao, 2023), the most viable (but costly) alternative is running permanent regression tests (Chen et al., 2023) to monitor and statistically track model performance on metrics relevant to the downstream service, akin to the daily “GPT-4 unicorn” (Adam, 2023).

The myth of determinism and the hardness of prompting. Another concern is the inherent non-determinism of most models accessible as-a-Service. This problem is not limited to LMaaS but extends to many generative AI techniques deployed and made available as-a-Service, such as diffusion-based image generators (Rombach et al., 2021). With LMaaS, the same prompt

[12] https://openai.com/blog/gpt-4-api-general-availability, accessed on 26/09/2023.

possibly returns more than one answer, aggravating reproducibility. Models such as GPT-3.5 and GPT-4, when accessed via API or via the web-interface provided by cloud services such as Microsoft Azure, allow adjusting the diversity of the completion generated over a user's prompt by tuning the value of parameters such as the *temperature*, as illustrated in Figure 3. In the case of GPT-3.5, the model should behave deterministically by setting the temperature to zero. Nevertheless, there have been reports that GPT-3.5 and GPT-4 are non-deterministic even for temperature zero (Chann, 2023; Ouyang et al., 2023).[13] For values greater than zero, an LMaaS samples the outputs according to a probability distribution, therefore, the sample from the distribution can be different at each run. The number of utterances a deterministic LMaaS can output is a (strict) subset of those a non-deterministic LMaaS can generate: in this sense, a model gains diversity with non-determinism but could diminish the consistency of so-generated answers. Summarizing, while undoubtedly helpful, controlling

Only fixing all the randomness sources (informally, their seed) allows one to control, reproduce, and trust experiments conducted with that model. Thus, precluding access to their source code erodes trust in the end-user and possibly weakens our control of the LMaaS. Another related risk that is debated in literature is whether a fully deterministic model that would expose a company to the risk of seeing its models mimicked by others. While some research papers show how to train a competitive replica of an LMaaS with access solely to the output of another model or its embedded representations (Mukherjee et al., 2023; Peng et al., 2023), others argue that "imitation" models only apparently close the gap with powerful LMaaS (Gudibande et al., 2023), leaving this debate unsolved for now.

Another issue that affects transparency is that different models offer different access interfaces (e.g., via web GUI or APIs), as reported in Table 1. The case of GPT-3.5 & 4 is emblematic: while the API access allows setting the temperature and other parameters, the corresponding web interface does not admit such options. Any LMaaS we want to evaluate rigorously should provide parameters to make such analysis deterministically reproducible. When that is not possible, benchmarking techniques need to take non-determinism into account, e.g., via sampling strategies that explore a model output's landscape robustly to account for non-deterministic behaviours. While this latter approach would not make experiments fully reproducible, it would at least amount to experiments that can be *trusted*.

3.3 Reliability

Reliability in machine learning is pivotal, signifying the consistency and dependability of a learning model's results. This concept is fundamental as it underpins the extent to which one can place confidence in the model's predictions or decisions within real-world scenarios. In the context of LMs and LMaaS, reliability encompasses models whose performance is quantifiable through metrics. These metrics should reflect the model's proficiency in accomplishing tasks that transcend mere memorization of training data and reliance on superficial patterns. This section delves into pertinent issues for benchmarking LMs, with the LMaaS framework intensifying these assessment challenges. A primary concern is *dataset contamination*. This occurs when the training data contains inputs and corresponding labels that are similar to, or exact replicas of, those utilized in the testbed. Another type of contamination, which we name *user contamination*, occurs when companies collect prompts to train and fine-tune their models and is specific to the LMaaS paradigm. Furthermore, this section explores the "emergent behaviour" concept in LMaaS. Emergence refers to the ability of LMaaS to tackle tasks not encountered during training. Identifying such emergent behaviours necessitates a preliminary determination of whether these tasks represent novel challenges. Lastly, we provide an overview of an ongoing debate in benchmarking practices: on the one hand, one can conduct comprehensive

[13] `https://152334h.github.io/blog/non-determinism-in-gpt-4/`, accessed on 24/08/2023.

evaluations across multiple datasets and metrics; on the other, one can employ techniques that assess and aggregate the meta-capabilities of such models in addressing analogous tasks.

Dataset contamination and user contamination. LMs are pre-trained on massive datasets that often consist of billions, if not trillions, of tokens, usually scraped as unstructured text from the web (Touvron et al., 2023b). Such an approach contrasts with the broader paradigm of (self)-supervised training, where models are trained and instructed on curated input/output samples and tested on unseen test data. The need for massive datasets induces scarcity of test beds (Van, 2023), as popular ones might be available online and consumed during training due to poor scraping and data collection policies. This is the issue of *dataset contamination*. In this setting, performance evaluation becomes non-trivial, as it must be conducted on data not used at training time. Memorization, defined as the tendency of an LM to output entire sequences seen at training time, further invalidates benchmarking (Ippolito et al., 2022) while also being an issue from the point of privacy, by revealing personal data contained in the training dataset (Carlini et al., 2022, 2021). Furthermore, models such as ChatGPT and Bard can develop answers that, while seeming correct at first glance, contain inaccurate, false, or not made-up information (Alkaissi and McFarlane, 2023; Zhang et al., 2023a). This phenomenon is endemic in almost all the LMaaS and most LMs (Maynez et al., 2020a), and companies are providing their services by mentioning such issues in their licences, as reported in Table 1.

Another related issue is that of *user contamination*, that arises when LMaaS, prone to memorisation (Biderman et al., 2023a,b), and are further trained on the user's inputs—a right which commercial licences retain, as illustrated in Table 1. *Dataset contamination* can be induced by *user contamination*: a model that does not (yet) suffer from *dataset contamination* can be prompted with a novel test bed (as is often the case during test bed creation); this test bed, being digested by the LMaaS, will then, via *user contamination*, be invalidated for use in ulterior benchmarks on that particular LMaaS. While some LMaaS deliver the option to opt-out of the mechanism of continual data collection, such services are not a standard practice and come with additional premium costs for the end user.

On tasks that are challenging for humans (the so-called high-order tasks, which stand in contrast to low-order tasks such as sentiment analysis), there is empirical evidence that LMaaS exhibit impressive performance (Liang et al., 2022; Wang et al., 2023b), and perform expertly on many tests designed to be challenging for humans (Zhang et al., 2023b). While recent developments in deep learning have significantly augmented the capabilities of LMs and LMaaS in manipulating symbols, the phenomenon of *dataset contamination* might lead to overestimating their actual performance.

An illustrative case is that of the work by Zhang et al. (2023b), which suggests that GPT-3.5 and GPT-4 ace entry exams at MIT. The article, subsequently withdrawn from arXiv,[14] raises concerns related to *dataset contamination* (among a number of other points of concern): a meta-analysis conducted on the paper suggests that results may have been contaminated (Chowdhuri et al., 2023). Issues are not limited to the case mentioned above as other researchers raised the issue of *dataset contamination* explicitly for models such as GPT-3.5 and GPT-4 (Aiyappa et al., 2023): well-established benchmarks have their test set, alongside the labels, available on sharing platforms such as GitHub (Jacovi et al., 2023), which raises the likelihood of them being included in the model's training data.

The effort to detect and remove datasets dates back to earlier LMs such as GPT-2 (Brown et al., 2020; Carlini et al., 2022; McCoy et al., 2023a). A proposed solution is to extract, from the training corpora or via testing a model's output, all the n-grams of a leaked test bed (Brown et al., 2020), though more refined procedures exist (McCoy et al., 2023a). Nevertheless, such approaches are computationally expensive and not sound: an unsuccessful result does not prove that a model has not digested (a slight variation of) it.

[14] https://arxiv.org/abs/2306.08997, accessed on 19/08/2023.

With classical LMs, one can detect and mitigate *dataset contamination* by choosing unfamiliar benchmarks to assess model reliability. If one does not trust a pre-trained model, it is possible (with some effort) to retrain it locally. One recent proposal to mitigate *dataset contamination* is to stop uploading test data in plain text to the internet (Jacovi et al., 2023). This, however, is not sufficient. Models equipped with tools to browse the internet and run code (such as the GPT-4 plug-ins) could decompress the test data in plain text in their context, which can then be used for training future versions of the models. Furthermore, the proposal by Jacovi et al. (2023) relies on an honour system: if one inadvertently exposes a test set to a model, then one should disclose it. However, knowledge of the practice and compliance would be hard to enforce.

Similarly, evaluating the LMaaS can leak the test samples to future iterations of its training if the model is trained on user inputs, as evidenced by *user contamination*. A viable approach involves the deployment of models which abstain from gathering data via prompts and are rigorously trained on datasets that are open to scrutiny.

For both *dataset contamination* and *user contamination*, it is also insufficient that the *exact* test set is not in the training data. One should also ensure that *the same information* is not in the training data. For example, even if a specific test set on solving addition problems is not ingested, one still needs to ensure that none of the individual problems have been independently developed, released online, and ingested in the training dataset. This is especially important for simple tasks such as summing double-digit numbers, of which hundreds of exercise sheets can be found online or generated with minimal effort.

Therefore, guaranteeing that no contamination occurs is not viable. Data handcrafted to test the model's capabilities once and then discarded, namely *one-shot data*, are practically impossible to employ, as this requires human experts or specialized algorithms. Generating high-quality data with a high degree of linguistic variability is, in fact, an open problem in NLP and computational linguistics. While LMs and LMaaS have a fairly highly developed facility to generate such data (Eldan and Li, 2023; Møller et al., 2023), they do not solve the problem (Dwivedi et al., 2023): managing the generative process inherent in LMs and LMaaS is linguistically challenging. There exists a substantial risk associated with evaluating the performance based on datasets the model may already classify with considerable accuracy, being that data is generated with high confidence by the model itself. Such a process also burdens the research community with the need to develop new datasets continuously and benchmarks, which are then immediately ingested by LMaaS providers, raising ethical concerns about the labour involved. Therefore, ensuring that the training set does not independently have the same data is necessary.

We, therefore, believe that the *dataset contamination* and *user contamination* problems cannot be solved without the active cooperation of the model developers. However, a few interesting *agnostic* approaches that estimate the probability of a sample to be part of the training data, which is assumed to be inaccessible as in the case of LMaaS, are emerging, see, e.g., the research by Shi et al. (2023).

In addition to such approaches, we propose investigating the option of developing a registry for models and datasets. A researcher could check which test samples are present in the training dataset of a given model without accessing the complete training dataset directly. Then, they could omit these samples from their evaluation. For small variations of an already digested input, a solution to speed up the search of similar sentences in a model training data can employ vector databases, which allow computing similarity between inputs that slightly vary (Han et al., 2023; Pan et al., 2023). Such an approach also addresses the dual problem: that of a conscientious model developer wanting to avoid the test sets of benchmarks and evaluations, which can be difficult when users might be inadvertently exposing them to the model. A model developer can check that their training dataset does not contain test sets from the registry. Furthermore, developers can dynamically check the user inputs and omit data samples that have such test samples from future training.

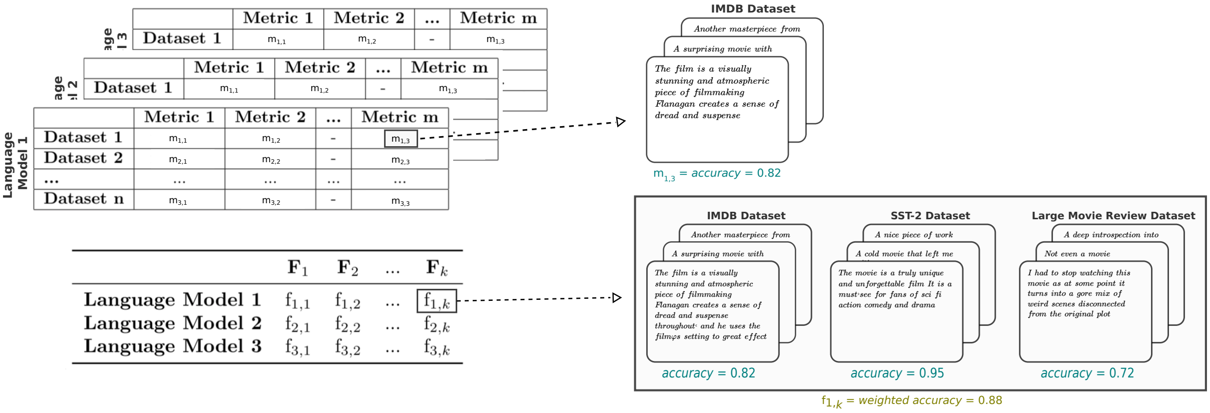
Company	Model	Opt-out	Train	Fine-tune	API	Accuracy Disclaimer
AI2Lab	Jurassic-1/2	×	✓	✓	✓	✓
Anthropic	Claude 1/2/3	✓	✓	✓	✓	✓
Cohere	Command	✓	✓	✓	✓	✓
Google	PaLM	✓	✓	✓	✓	✓
	Bard (Powered by PaLM)	✓	✓	✓	✓[†]	✓
Inflection	Pi	×	✓	✓	×	✓
Microsoft	Microsoft Copilot in Bing	×	✓	×	×	✓
OpenAI	GPT-3.5/4/4o API	✓	✓	✓	✓	✓
	GPT-3.5/4/4o web	✓[‡]	✓	✓[∔]		

Table 1: An analysis of how companies expose their LMaaS to users. Opt-out refers to the possibility of rejecting data collection while still accessing the models fully. Symbols ✓ and × indicate whether APIs and opt-out are available, and prompts are explicitly collected for fine-tuning the current LMaaS or training a new model (or product). The Accuracy Disclaimer means the relevant company does not provide any warranty as to the accuracy of the model's output. Relevant URLs to licences and access are reported in Appendix A. The Table is illustrative and thus not exhaustive .
[†] API access is available in beta mode.
[‡] Opt-out is the default for APIs since March 1st, 2023, while opt-in is the default for the web interface.
[∔] Fine-tuning on the web-interface is not as flexible as that provided by the API.

Evaluating emergence. There have been high-profile claims that some of the most advanced LMaaS exhibit impressive emergent capabilities (Bubeck et al., 2023; Huang et al., 2023a; Singhal et al., 2023), i.e., capabilities not explicitly programmed or anticipated during their development that manifest when a model reaches a certain scale or complexity. In practice, emergent abilities are behaviours that cannot be directly generalized from the training data (Wei et al., 2022a), and arise when models are scaled up in size and complexity (Biderman et al., 2023b). For example, if a model has only seen sentiment analysis data during training, we consider its ability to solve math problems as emergent. At the same time, despite recent works showing how such properties occur in LMs (Lu et al., 2023), there is reason to doubt whether these are indeed emergent abilities for LMaaS, as one cannot access the training data. Hence, one cannot evaluate the similarity of the supposed emergent ability to the abilities encoded in the training data.

Evaluating whether LMaaS have emergent abilities is even more difficult than measuring their performance on a benchmark. While the previous section outlined ways to check whether an *exact replica* of the test data is contained in the training dataset, the emergence case requires evaluating whether data for *similar tasks* we want to test for are present in the training data. While a dataset and benchmark registry can be endowed with a notion of *similarity measure*, finding *similar* rather than *exact* data is computationally challenging, when not impossible (establishing the semantic equivalence between two strings is undecidable). Therefore, for the immediate future, claims about the emergent abilities of LMaaS will likely remain highly questionable and, for LMaaS, not actionable as there is no way to access the models we want to assess (while similar tools have already been developed for standard LMs (Biderman et al., 2023b)).

We end with a note of caution: the existence of emergent abilities (which entail the challenges of testing them on LMaaS) is not definitively established, and in some cases, researchers discovered tasks where performances are negatively correlated with the size of the model (McKenzie et al., 2023) and positively correlated with the probability of a similar prompt to be present in the training data (McCoy et al., 2023b). There have also been theoretical results showing that obtaining novel behaviours with prompting is challenging and that a model is good at a task indicates that it has likely seen similar or related tasks during pre-training (Petrov et al., 2023a). What appear to be emergent abilities may be artefacts of poorly chosen evaluation metrics (Schaeffer et al., 2023a). This fact is exacerbated by the startling observation that an increase in performance that comes from Chain-of-Thought (CoT) prompting is still observable if the prompt has logical errors (Schaeffer et al., 2023b). Nonetheless, models appear to be more than merely "stochastic parrots", as shown via a combinatorial argument by Yu et al. (2023), so the question of a model's true capabilities, and whether it can exhibit emergent abilities or is constrained in specific ways, is still open.



	Metric 1	Metric 2	...	Metric m
Dataset 1	m$_{1,1}$	m$_{1,2}$	-	m$_{1,3}$
Dataset 2	m$_{2,1}$	m$_{2,2}$	-	m$_{2,3}$
...	...	...	...	...
Dataset n	m$_{3,1}$	m$_{3,2}$	-	m$_{3,3}$

	$\mathbf{F}_1$	$\mathbf{F}_2$	...	$\mathbf{F}_k$
Language Model 1	$f_{1,1}$	$f_{1,2}$	...	$f_{1,k}$
Language Model 2	$f_{2,1}$	$f_{2,2}$	...	$f_{2,k}$
Language Model 3	$f_{3,1}$	$f_{3,2}$	...	$f_{3,k}$

Another masterpiece from

A surprising movie with

The film is a visually stunning and atmospheric piece of filmmaking Flanagan creates a sense of dread and suspense throughout‹ and he uses the filmφs setting to great effect

A nice piece of work

A cold movie that left me

The movie is a truly unique and unforgettable film It is a must›see for fans of sci fi action comedy and drama

A deep introspection into

Not even a movie

I had to stop watching this movie as at some point it turns into a gore mix of weird scenes disconnected from the original plot

Figure 4: Extensive benchmarking techniques (top) suffer from poor inspectability as they have a cubic growth rate w.r.t. the number of models, datasets and metrics involved. On the other hand, techniques that cluster models based on dimensionality reduction and distilled latent factors (bottom) aggregate multiple datasets and/or metrics but suffer from poor interpretability.

Comparing LMaaS. It is not only *dataset contamination* and *user contamination* that makes it challenging to compare general-purpose LMaaS. While the scientific literature is moving towards multi-task, multi-models measures of performance (Liang et al., 2022), or, generally speaking, towards more extensive and complex benchmarks (Kiela et al., 2021; Suzgun et al., 2022), the debate is still open on how to compare two models on a broader, multi-domain set of tasks (Chang et al., 2023). A model might be better at one task and poorer at another; it is thus hard to imagine a measure of performance that imposes a partial/total order over LMaaS performances. Moreover, benchmarks such as BIG-Bench (Srivastava et al., 2022), designed to progressively include more complex task instances, may diverge from the initial benchmarking purpose to encompass samples outside the distribution, where models exhibit failure.

Evaluating LMs and LMaaS is difficult for many reasons: with tens of models available, hundreds of metrics, and test beds, an extensive assessment of each case is expensive and necessarily time-consuming.[15] LMaaS add a further layer of complexity championed by *data/user contamination*, model replacement, and non-determinism. If *contamination* invalidates results by boosting performances on memorized test beds, experiments conducted at different time-frames do not reflect the current capabilities of a model, as a model may have been substituted in the meantime with a different version whose performances varied considerably (Chen et al.,

[15] If the experiments reported in (Liang et al., 2022) were conducted solely on models with pay-per-usage costs similar to that of ChatGPT-3.5-turbo, they would cost approximately USD 18 250, without counting the cost of machines and engineering labour.

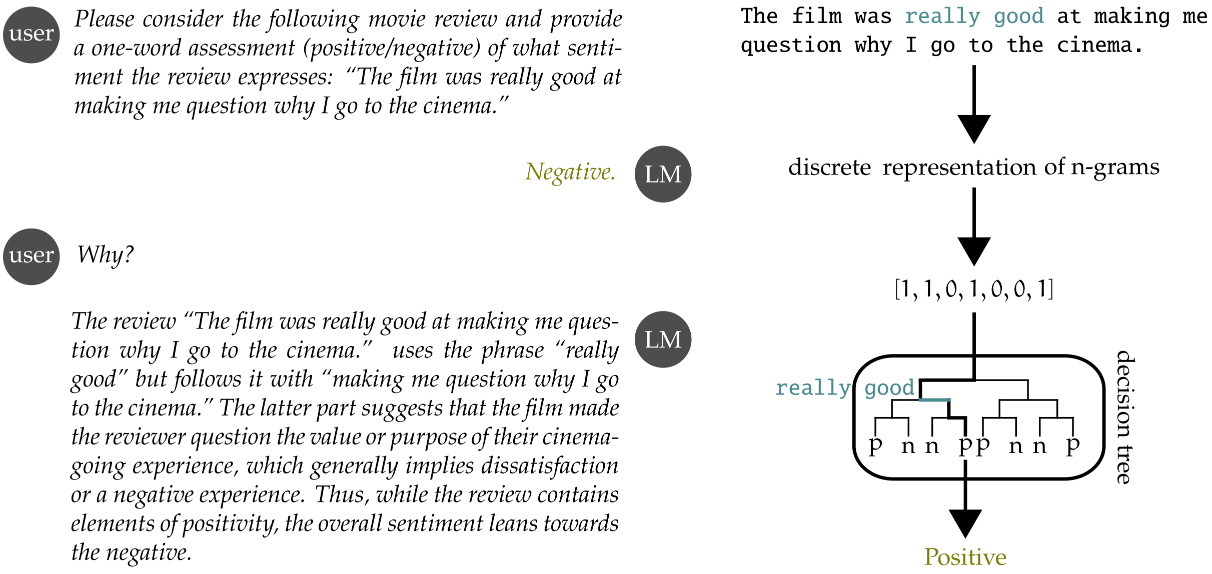
2023). Non-determinism, unless tamed (e.g., as suggested in Section 3.2), makes point-wise comparisons uncertain and subject to larger numerical deviations.

We observe a tension in the recent literature between benchmarking via point-wise metrics on a large number of models and scenarios (Liang et al., 2022; Srivastava et al., 2022; Zhong et al., 2023), and evaluations where models are aggregated *post-hoc* based on latent factors that capture variations over comparable tasks (Burnell et al., 2023). Such methods are prone to illusory correlation, i.e., tasks are grouped and thus considered similar based on the LMaaS performances.

An illustrative example of the difference between extensive benchmarking and latent factor analysis is illustrated in Figure 4: approaches such as HELM (Liang et al., 2022), extensively benchmark LMaaS for different metrics and settings, while dimensionality reduction methods (bottom) aggregate LMs and LMaaS based on tasks where they performed similarly, yet the computed latent factors can be hard to interpret.

In conclusion, rigorous benchmarking requires mitigating *data/user contamination*, non-determinism, and most of the previous issues mentioned in the paper, thus representing a long-term challenge on which the research community and the model providers must necessarily collaborate.

3.4 Trustworthiness



user: *Please consider the following movie review and provide a one-word assessment (positive/negative) of what sentiment the review expresses: "The film was really good at making me question why I go to the cinema."*

LM: *Negative.*

user: *Why?*

LM: *The review "The film was really good at making me question why I go to the cinema." uses the phrase "really good" but follows it with "making me question why I go to the cinema." The latter part suggests that the film made the reviewer question the value or purpose of their cinema-going experience, which generally implies dissatisfaction or a negative experience. Thus, while the review contains elements of positivity, the overall sentiment leans towards the negative.*

The film was really good at making me question why I go to the cinema.
discrete representation of n-grams
[1, 1, 0, 1, 0, 0, 1]
really
good
p n n p p n n p
decision tree
Positive

Figure 5: LMaaS can be used to generate *explanations*. On the left is a real interaction with GPT-4 (dated September 21, 2023), asked to solve a sentiment analysis task. In this instance, the reasoning is sound, but we have no guarantee of correctness. On the right, the workings of a fictitious decision tree to solve the same sentiment analysis task are illustrated on text represented as 2-grams: while it can misclassify the input as "positive", after having represented the input as vectors, it is possible to trace the reason for the misclassification back to the existence of the 2-gram "really good", which affected some of the decisions that were taken. In this sense, techniques like decision trees are *self-explaining*, as they embed explanations from which one can derive the model's decision-making process.

LMs and LMaaS have progressively grown in sophistication up to the degree that one can query them to produce an explanation for their output (Alvarez-Melis and Jaakkola, 2018; Zini and Awad, 2022): we refer to such models as *explanatory neural networks* (AlRegib and

Prabhushankar, 2022), to distinguish them with *self-explaining*, which concerns another family of explainability techniques (Rudin, 2019). *Self-explaining* methods produce explanations that make the model's decision process transparent, as in the case of decision trees, where one can extract an intelligible rule from the branches a model follows to make its decision. LMaaS prompted for an explanation cannot provide insights into a model's decision process: an explanation for an LMaaS is, in fact, just a conditional prompt over a model's previous interaction, as sketched in Figure 5. This problem also affects standard LMs, though having access to their internal states offers deeper introspection of their decision processes (Azaria and Mitchell, 2023). LMaaS internals accessible through API or web interface prevents introspection and limits the explanation of a model's decision. An explanation only enforces trust in a model if we trust the explanation itself: for example, if correctness guarantees accompany it. An explanation for an LMaaS is just a conditional prompt over a model's previous interaction; hence, there is no basis to trust it.

Popular prompting techniques methods such as *Chain of Thoughts* (Wei et al., 2022b) (CoT) and its variations (Zhang et al., 2022; Besta et al., 2023; Paranjape et al., 2023; Yao et al., 2023b), are employed to enhance an LMs' performance by asking a model to *reason step by step and write it thoughts* when producing the answer to a problem. This step-by-step reasoning improves LMaaS performances and adds to their explainability (Kojima et al., 2022). However, the improved performance might be due to the increasing complexity of the model (Feng et al., 2023) and does not prevent an LMaaS from generating untrustworthy rationales (Turpin et al., 2023). Nor do prompts have to be logically correct to increase the performance (see (Schaeffer et al., 2023b)), as we noted in Section 3.3.

Explicitness, faithfulness, and stability are three common desiderata of explanations (Alvarez-Melis and Jaakkola, 2018; Maynez et al., 2020b; Li et al., 2023). General-purpose LMaaS produce answers that maximize explicitness: in this sense, an explanation, which for an LMaaS is a conditioned prompt, will be immediate and understandable. On the other hand, faithfulness, as the relevance of each input variable (in the case of LMaaS, each token) for the model's decision, and stability, as the consistency of an explanation for slight input variations, are not – though some works are moving in that direction (Huang et al., 2023b; Lanham et al., 2023).

We argue that faithful explanations should include the sufficient causes that led a model to output an answer to a specific prompt (Darwiche and Hirth, 2020), while stability is closely connected to robustness. The research literature has provided overwhelming evidence that explanations for machine learning models are not sufficient to imply a model's prediction and are highly sensitive to slight input variations (Ignatiev et al., 2019; Marques-Silva and Ignatiev, 2022; Izza et al., 2023) unless explicitly trained with that objective. Including, but not limited to, safety-critical applications, we thus advocate for methods that explicitly embody robustness guarantees, with the double intent to provide security to the end-user and not to erode their trust. Though certificates of optimality and robustness do not scale beyond small-scale models (Malfa et al., 2021), we champion approximate methods and probabilistic guarantees. Such properties, combined with invariance to similar inputs and the explicitness mentioned above (which can be enforced at training time by algorithms such as reinforcement learning through human feedback and its variations (Bai et al., 2022; Gulcehre et al., 2023)), can align the explainability of an LMaaS to the desiderata the research community is recommending for models that inherently suffer of poor interpretability. While LMaaS are kept closed-source to maintain a competitive advantage over other providers, we believe that including, alongside the output, intermediate representations (e.g., their internal states and their output logits) is a reasonable trade-off between closed-source and interpretability, with companies that would benefit from red-teaming operations by teams of researchers and developers. Other lines of research that should be pursued include interpretability and model steering. Companies can release, alongside non-inspectable LMaaS, surrogate models trained on their representation, similarly to the line of research Anthropic follows with Claude (Templeton et al., 2024; Anthropic, 2024). An LMaaS whose features are interpretable and outputs governable, e.g., via surrogate models such as Sparse Auto-Encoders (Bricken et al., 2023), would enhance trust in the end user and further fuel areas of research that are at the moment uncharted territory. In conclusion, while some works are moving in the direction of grounding explanations with external knowledge (Mei et al., 2023; Ohmer et al., 2023), we must develop strategies beyond recursive prompting that provide formally guaranteed and unbiased introspection of a model's decision landscape (Bills et al., 2023), and integrate the training process with faithful *post-hoc* methods (Krishna et al., 2023).

4. Mitigating LMaaS Issues: a Tentative Agenda

This work highlights four aspects that differentiate LMaaS from LMs: accessibility, reproducibility, reliability, and trustworthiness. These issues affect our ability to understand the capabilities and limitations of LMaaS, which hundreds of millions of users use daily. We, therefore, need to work as a community to find solutions that enable researchers, policymakers, and members of the public to trust LMaaS. Below, we summarize previously identified issues and emphasize a path forward by highlighting challenges that must be addressed.

Accessibility. To enhance accessibility, companies should release the source code (or at least detailed model cards) of the LMs that power their LMaaS. While licences that prevent free commercial usage wouldn't be enough to guarantee that companies can retain their competitive advantage (other companies could leverage their findings and larger computational infrastructures), we recommend that the source code (or very comprehensive model card) of LMaaS should at least be available to auditors/evaluators/red teams with restrictions on sharing. Accessibility would be further enhanced by companies that release their LMs in different *sizes*, as in the case of Alpaca LLaMA and Pythia, so that researchers with access to limited computational facilities can still experiment with their models. Regarding imbalance across languages, fair tokenizers and pay-per-token access policies can spread the usage of LMaaS among economically disadvantaged and low-resource language customers. We also need to assess and quantify the gap disadvantaged users and countries with limited technology access face when accessing LMaaS. Doing so can provide insights and techniques for companies and policymakers to mitigate unfair treatment.

Reproducibility. Reproducibility requires that LMaaS are not taken offline when a newer version is deployed. Providing access to such older models is certainly not remunerative for companies. Furthermore, releasing legacy LMaaS under permissive, if not open-source, licences is a strategy that, from a commercial perspective, likely damages providers despite allowing them to benefit from the research community discovering and reporting biases, bugs, and vulnerabilities. The research community would benefit from old models made available to researchers for as long as possible, with companies that warn clearly before updates and give lead time before deprecating old versions to enable the completion of experiments and replication efforts. At a minimum, all the parameters that make up a model should be hashed, and a log of "model commits' should be offered to the user by model maintainers as the maintainer updates the model. The benefit is that specific user interactions with the models (in particular, benchmarks researchers make) can be matched to model commit hashes. Companies could also offer the option to make a model's behaviour fully deterministic (although this might increase certain attacks on the model's architecture). We do not argue that all models released should be deterministic but that we prefer them for scientific evaluation purposes over their non-deterministic counterparts. In this sense, the scientific community, in terms of journals and conference venues, should discourage the usage of models that do not meet adequate reproducibility requirements.

Reliability. *Dataset contamination* can be addressed by two complementary approaches: on the one hand, LMaaS should state clearly the datasets on which they have been trained, similarly to model cards for LMs. The industrial and research communities can collaborate to jointly develop fast indexing techniques to assess whether a model has digested an input or a slight variation. Models that collect prompts from interaction with the users should not be dismissed, but the research community, in the absence of tools to inspect whether a prompt from a test bed has been collected, should discourage their usage for reporting purposes (e.g., benchmarking). Concerning benchmarking, we argue that the research community should study and develop benchmarking methodologies that search for and test latent factors that can explain performances across tasks while avoiding *post-hoc* methods to maintain a sufficient degree of interpretability. As most efforts are currently directed towards the development of extensive testbeds such as HELM or BIG-Bench, which suffer from poor inspectability and possibly test on out-of-distribution data, we think that more can be done in the direction of methodologies that group LMaaS and LMs by metrics and datasets, enabling interpretable model comparisons.

Trustworthiness. Finally, faithfulness and stability should be embodied in LMaaS and explainability tools that make their behaviour intelligible. Faithfulness is achievable through grounding and sufficiency. Despite different connotations, both terms refer to the elements of a prompt that imply a model's decision. In contrast, robustness and invariance to slight prompt variations can enable stability. Formal methods and robustness have already developed a rich corpus of literature from which the community should draw inspiration. The long-term objective is to deploy applications powered by LMaaS and LMs that can serve in safety-critical settings. This can instil trust in the reasons behind a model's decisions, and this is an ongoing research effort.

Acknowledgements

ELM is supported by the Alan Turing Institute. AP is supported by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (EP/S024050/1). AGC is supported by the Economic and Social Research Council (ESRC) under grant ES/W003473/1; and the Turing Defence and Security programme through a partnership with the UK government in accordance with the framework agreement between GCHQ and the Alan Turing Institute.

Appendix A. Commercial Licences for Common LMaaS and Opt-Out Form

In the following, the main links (at the time of submission) are listed by the company hosting the language model.

AI2Lab
https://www.ai21.com/terms-of-use
https://www.ai21.com/privacy-policy
https://studio.ai21.com/privacy-policy

Anthropic
https://console.anthropic.com/legal/terms
https://console.anthropic.com/legal/privacy
https://support.anthropic.com/en/articles/7996868-i-want-to-opt-out-of-my-prompts-and-results-being-used-for-training-models

Cohere
https://cohere.com/saas-agreement

Google
https://support.google.com/bard/answer/13594961

https://support.google.com/bard/answer/13594961?hl=en
https://cloud.google.com/vertex-ai/docs/generative-ai/learn/responsible-ai
https://docs.google.com/forms/d/e/1FAIpQLSdUwCF62JRg8rVYh5IaN7VWwIrLtWbxtcQDRC97zbzoq54bfg/viewform
https://blog.google/technology/ai/an-update-on-web-publisher-controls/

Inflection
https://pi.ai/profile/policy - Licence available after log-in.
https://pi.ai/profile/terms - Licence available after log-in.

Microsoft

https://privacy.microsoft.com/privacystatement

OpenAI

https://openai.com/policies/terms-of-use
https://help.openai.com/en/articles/7730893-data-controls-faq
https://platform.openai.com/docs/models/how-we-use-your-data
https://openai.com/index/introducing-gpts/

All the URLs listed above were last accessed on 30/07/2023, except for the Anthropic and OpenAI, which were last accessed on 29/12/2023.

References

Dean Adam. 2023. GPT Unicorn: A daily exploration of GPT-4's image generation capabilities. Accessed on July 31, 2023.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R Mortensen, Noah A Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? Tokenization in the era of commercial language models. *ArXiv preprint*, abs/2305.13707.

Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on ChatGPT? *ArXiv preprint*, abs/2303.12767.

Hussam Alkaissi and Samy I McFarlane. 2023. Artificial hallucinations in ChatGPT: Implications in scientific writing. *Cureus*, 15(2).

Ghassan AlRegib and Mohit Prabhushankar. 2022. Explanatory paradigms in neural networks. *ArXiv preprint*, abs/2202.11838.

David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7786–7795.

AI Anthropic. 2024. The Claude 3 model family: Opus, Sonnet, Haiku. *Claude-3 Model Card*, 1.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when its lying. *arXiv preprint arXiv:2304.13734*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. *ArXiv preprint*, abs/2302.04023.

Jeremy Barnes. 2021. Is it time to move beyond sentence classification?

Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment analysis is not solved! Assessing and probing sentiment classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 12–23, Florence, Italy. Association for Computational Linguistics.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: LLMs trained on "A is B" fail to learn "B is A". *arXiv preprint arXiv:2309.12288*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with Large Language Models. *arXiv preprint arXiv:2308.09687*.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. 2023a. Emergent and predictable memorization in Large Language Models. *arXiv preprint arXiv:2304.11158*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023b. Pythia: A suite for analyzing Large Language Models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *ArXiv preprint*, abs/2108.07258.

Alexander Borzunov, Dmitry Baranchuk, Tim Dettmers, Max Ryabinin, Younes Belkada, Artem Chumachenko, Pavel Samygin, and Colin Raffel. 2022. Petals: Collaborative inference and fine-tuning of large models. *arXiv preprint arXiv:2209.01188*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *ArXiv preprint*, abs/2303.12712.

Ryan Burnell, Han Hao, Andrew RA Conway, and Jose Hernandez Orallo. 2023. Revealing the structure of language model capabilities. *ArXiv preprint*, abs/2306.10062.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *ArXiv preprint*, abs/2202.07646.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*, volume 2, pages 830–835.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of Large Language Models. *ArXiv preprint*, abs/2307.03109.

Sherman Chann. 2023. Non-determinism in GPT-4 is caused by sparse MoE. Accessed on August 5, 2023.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box Large Language Models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is ChatGPT's behavior changing over time? *ArXiv preprint*, abs/2307.09009.

Xinyun Chen, Chang Liu, and Dawn Song. 2019. Execution-guided neural program synthesis. In *International Conference on Learning Representations*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311.

Raunak Chowdhuri, Neil Deshmukh, and David Koplow. 2023. No, GPT4 can't ace MIT.

Anthony G Cohn and Jose Hernandez-Orallo. 2023. Dialectical language model evaluation: an initial appraisal of the commonsense spatial reasoning abilities of LLMs. *arXiv preprint arXiv:2304.11164*.

Adnan Darwiche and Auguste Hirth. 2020. On the reasons behind decisions. *ArXiv preprint*, abs/2002.09284.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Li Du, Lucas Torroba Hennigen, Tiago Pimentel, Clara Meister, Jason Eisner, and Ryan Cotterell. 2022. A measure-theoretic characterization of tight language models. *ArXiv preprint*, abs/2212.10502.

Yogesh K Dwivedi, Nir Kshetri, Laurie Hughes, Emma Louise Slade, Anand Jeyaraj, Arpan Kumar Kar, Abdullah M Baabdullah, Alex Koohang, Vishnupriya Raghavan, Manju Ahuja, et al. 2023. "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management*, 71:102642.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D Hwang, et al. 2023. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent English? *ArXiv preprint*, abs/2305.07759.

Guhao Feng, Yuntian Gu, Bohang Zhang, Haotian Ye, Di He, and Liwei Wang. 2023. Towards revealing the mystery behind chain of thought: A theoretical perspective. *ArXiv preprint*, abs/2305.15408.

Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, and Julius Berner. 2023. Mathematical capabilities of ChatGPT. *ArXiv preprint*, abs/2301.13867.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *ArXiv preprint*, abs/2209.07858.

Leo Gao. 2023. On the sizes of OpenAI API models. Accessed on July 31, 2023.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, Melbourne, Australia. Association for Computational Linguistics.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary LLMs. *arXiv preprint arXiv:2305.15717*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*.

Hacker News. 2023a. ChatGPT use declines as users complain about 'dumber' answers. Accessed on July 31, 2023.

Hacker News. 2023b. Experiencing decreased performance with ChatGPT-4. Accessed on July 31, 2023.

Yikun Han, Chunjiang Liu, and Pengfei Wang. 2023. A comprehensive survey on vector database: Storage and retrieval technique, challenge. *arXiv preprint arXiv:2310.11703*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. 2023. An overview of catastrophic AI risks. *arXiv preprint arXiv:2306.12001*.

Joachim Henkel. 2009. Champions of revealing—the role of open source developers in commercial firms. *Industrial and Corporate Change*, 18(3):435–471.

Michael Heron, Vicki L Hanson, and Ian Ricketts. 2013. Open source and accessibility: advantages and limitations. *Journal of interaction Science*, 1(1):1–10.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, 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. In *International Conference on Learning Representations*.

Qiuyuan Huang, Jae Sung Park, Abhinav Gupta, Paul Bennett, Ran Gong, Subhojit Som, Baolin Peng, Owais Khan Mohammed, Chris Pal, Yejin Choi, and Jianfeng Gao. 2023a. ArK: Augmented reality with knowledge interactive emergent ability. *arXiv preprint arXiv:2305.00970*.

Shiyuan Huang, Siddarth Mamidanna, Shreedhar Jangam, Yilun Zhou, and Leilani H Gilpin. 2023b. Can Large Language Models explain themselves? a study of LLM-generated self-explanations. *arXiv preprint arXiv:2310.11207*.

Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. 2019. Abduction-based explanations for machine learning models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1511–1519.

Daphne Ippolito, Florian Tramèr, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher A Choquette-Choo, and Nicholas Carlini. 2022. Preventing verbatim memorization in language models gives a false sense of privacy. *arXiv preprint arXiv:2210.17546*.

Yacine Izza, Alexey Ignatiev, Peter Stuckey, and Joao Marques-Silva. 2023. Delivering inflated explanations. *ArXiv preprint*, abs/2306.15272.

Alon Jacovi, Avi Caciularu, Omer Goldman, and Yoav Goldberg. 2023. Stop uploading test data in plain text: Practical strategies for mitigating data contamination by evaluation benchmarks. *ArXiv preprint*, abs/2305.10160.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv preprint*, abs/2001.08361.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. ChatGPT: Jack of all trades, master of none. *ArXiv preprint*, abs/2302.10724.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are zero-shot reasoners. *ArXiv preprint*, abs/2205.11916.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. 2023. Post hoc explanations of language models can improve language models. *arXiv preprint arXiv:2305.11426*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. *ArXiv preprint*, abs/2305.18486.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating ChatGPT's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *ArXiv preprint*, abs/2304.11633.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *ArXiv preprint*, abs/2211.09110.

Andreas Liesenfeld, Alianda Lopez, and Mark Dingemanse. 2023. Opening up ChatGPT: Tracking openness, transparency, and accountability in instruction-tuned text generators. *ArXiv preprint*, abs/2307.05532.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2023a. We're afraid language models aren't modeling ambiguity. *ArXiv preprint*, abs/2304.14399.

Ruibo Liu, Ge Zhang, Xinyu Feng, and Soroush Vosoughi. 2022. Aligning generative language models with human values. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 241–252, Seattle, United States. Association for Computational Linguistics.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023b. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. 2023c. Jailbreaking ChatGPT via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. Are emergent abilities in Large Language Models just in-context learning? *arXiv preprint arXiv:2309.01809*.

Emanuele La Malfa and Marta Kwiatkowska. 2022. The king is naked: On the notion of robustness for natural language processing. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11047–11057. AAAI Press.

Emanuele La Malfa, Rhiannon Michelmore, Agnieszka M. Zbrzezny, Nicola Paoletti, and Marta Kwiatkowska. 2021. On guaranteed optimal robust explanations for NLP models. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 2658–2665. ijcai.org.

Joao Marques-Silva and Alexey Ignatiev. 2022. Delivering trustworthy AI through formal XAI. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12342–12350.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020a. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020b. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023a. How much do language models copy from their training data? Evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. 2023b. Embers of autoregression: Understanding Large Language Models through the problem they are trained to solve. *arXiv preprint arXiv:2309.13638*.

Ian R McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, et al. 2023. Inverse scaling: When bigger isn't better. *arXiv preprint arXiv:2306.09479*.

Alex Mei, Sharon Levy, and William Yang Wang. 2023. Foveate, attribute, and rationalize: Towards physically safe and trustworthy AI. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11021–11036.

Anders Giovanni Møller, Jacob Aarup Dalsgaard, Arianna Pera, and Luca Maria Aiello. 2023. Is a prompt and a few samples all you need? Using GPT-4 for data augmentation in low-resource classification tasks. *ArXiv preprint*, abs/2304.13861.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of GPT-4. *ArXiv preprint*, abs/2306.02707.

Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. 2021. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62.

Xenia Ohmer, Elia Bruni, and Dieuwke Hupkes. 2023. Evaluating task understanding through multilingual consistency: A ChatGPT case study. *ArXiv preprint*, abs/2305.11662.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*, abs/2303.08774.

OpenAI. 2023a. Introducing ChatGPT. Accessed on April 11, 2023.

OpenAI. 2023b. OpenAI documentation - fine-tuning. Accessed on August 7, 2023.

OpenAI Community. 2023. Experiencing decreased performance with ChatGPT-4. Accessed on July 31, 2023.

Shuyin Ouyang, Jie M Zhang, Mark Harman, and Meng Wang. 2023. LLM is like a box of chocolates: the non-determinism of ChatGPT in code generation. *arXiv preprint arXiv:2308.02828*.

James Jie Pan, Jianguo Wang, and Guoliang Li. 2023. Survey of vector database management systems. *arXiv preprint arXiv:2310.14021*.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. ART: Automatic multi-step reasoning and tool-use for Large Language Models. *arXiv preprint arXiv:2303.09014*.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. 2023. Are you copying my model? Protecting the copyright of Large Language Models for EaaS via backdoor watermark. *ArXiv preprint*, abs/2305.10036.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Aleksandar Petrov, Adel Bibi, and Philip Torr. 2023a. When do prompting and prefix-tuning work? a theory of capabilities and limitations. *arXiv preprint arXiv:2310.19698*.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023b. Language model tokenizers introduce unfairness between languages. *ArXiv preprint*, abs/2305.15425.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training Gopher. *ArXiv preprint*, abs/2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Partha Pratim Ray. 2023. ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*.

Bernardino Romera-Paredes and Philip H. S. Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2152–2161. JMLR.org.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code Llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? *ArXiv preprint*, abs/2303.17548.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *ArXiv preprint*, abs/2211.05100.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023a. Are emergent abilities of Large Language Models a mirage? *arXiv preprint arXiv:2304.15004*.

Rylan Schaeffer, Kateryna Pistunova, Samar Khanna, Sarthak Consul, and Sanmi Koyejo. 2023b. Invalid logic, equivalent gains: The bizarreness of reasoning in language model prompting. *arXiv preprint arXiv:2307.10573*.

Christian Schlarmann and Matthias Hein. 2023. On the adversarial robustness of multi-modal foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3677–3685.

Murray Shanahan. 2022. Talking about Large Language Models. *ArXiv preprint*, abs/2212.03551.

Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In ChatGPT we trust? Measuring and characterizing the reliability of ChatGPT. *ArXiv preprint*, abs/2304.08979.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from Large Language Models. *arXiv preprint arXiv:2310.16789*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large Language Models encode clinical knowledge. *Nature*, pages 1–9.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Noëlle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615.

Richard Stallman. 2010. What does that server really serve?

Richard Stallman. 2022. Open source misses the point.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20841–20855. PMLR.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. *ArXiv preprint*, abs/2210.09261.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *ArXiv preprint*, abs/2201.08239.

Nathaniel Tkacz. 2020. *Wikipedia and the Politics of Openness*. University of Chicago Press.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. LLaMA: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. LLaMA 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *arXiv preprint arXiv:2305.04388*.

Hoang Van. 2023. Mitigating data scarcity for Large Language Models. *arXiv preprint arXiv:2302.01806*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, et al. 2023a. On the robustness of ChatGPT: An adversarial and out-of-distribution perspective. *ArXiv preprint*, abs/2302.12095.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023b. Is ChatGPT a good sentiment analyzer? A preliminary study. *ArXiv preprint*, abs/2304.04339.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of Large Language Models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in Large Language Models. *ArXiv preprint*, abs/2201.11903.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *ArXiv preprint*, abs/1910.03771.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning – a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265.

Dongyu Yao, Jianshu Zhang, Ian G Harris, and Marcel Carlsson. 2023a. FuzzLLM: a novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in Large Language Models. *arXiv preprint arXiv:2309.05274*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023b. Tree of thoughts: Deliberate problem solving with Large Language Models. *arXiv preprint arXiv:2305.10601*.

Dingli Yu, Simran Kaur, Arushi Gupta, Jonah Brown-Cohen, Anirudh Goyal, and Sanjeev Arora. 2023. Skill-mix: A flexible and expandable family of evaluations for ai models. *arXiv preprint arXiv:2310.17567*.

Ali Zarifhonarvar. 2023. Economics of ChatGPT: A labor market view on the occupational impact of artificial intelligence. *Available at SSRN 4350925*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023a. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Sarah J Zhang, Samuel Florin, Ariel N Lee, Eamon Niknafs, Andrei Marginean, Annie Wang, Keith Tyser, Zad Chin, Yann Hicke, Nikhil Singh, et al. 2023b. Exploring the MIT mathematics and EECS curriculum using Large Language Models. *ArXiv preprint*, abs/2306.08997.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in Large Language Models. *arXiv preprint arXiv:2210.03493*.

Mengjie Zhao, Fei Mi, Yasheng Wang, Minglei Li, Xin Jiang, Qun Liu, and Hinrich Schütze. 2021. LMTurk: Few-shot learners as crowdsourcing workers in a language-model-as-a-service framework. *arXiv preprint arXiv:2112.07522*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of Large Language Models. *ArXiv preprint*, abs/2303.18223.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. AGIEval: A human-centric benchmark for evaluating foundation models. *ArXiv preprint*, abs/2304.06364.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2022. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Julia El Zini and Mariette Awad. 2022. On the explainability of natural language processing deep models. *ACM Computing Surveys*, 55(5):1–31.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.