Towards Safer Large Language Models (LLMs)

LAWRENCE Carolin, BIFULCO Roberto, GASHTEOVSKI Kiril, HUNG Chia-Chien, BEN RIM Wiem, SHAKER Ammar, OYAMADA Masafumi, SADAMASA Kunihiko, ENOMOTO Masafumi, TAKEOKA Kunihiro

Abstract

Large Language Models (LLMs) are revolutionizing our world. They have impressive textual capabilities that will fundamentally change how human users can interact with intelligent systems. Nonetheless, they also still have a series of limitations that are important to keep in mind when working with LLMs. We explore how these limitations can be addressed from two different angles. First, we look at options that are currently already available, which include (1) assessing the risk of a use case, (2) prompting a LLM to deliver explanations and (3) encasing LLMs in a human-centred system design. Second, we look at technologies that we are currently developing, which will be able to (1) more accurately assess the quality of an LLM for a high-risk domain, (2) explain the generated LLM output by linking to the input and (3) fact check the generated LLM output against external trustworthy sources.

Keywords

summarization, question-answering, safety, human-centric, explainability

1. Introduction

In an era of remarkable technological advancements, Large Language Models (LLMs) have emerged as a transformative force. Their astonishing performance[1] on many tasks has led to an exponential increase in real-world applications of LLM-based technology. LLMs are reshaping the way society interacts with computer systems, textual information and even our own creativity. They are in the process of revolutionizing various industries, being applicable to nearly any vertical, they help us by generating textual content, powering virtual assistants, and providing insights from large-scale data analysis. As such, LLMs provide an opportunity to bring about unprecedented levels of convenience and efficiency. However, as we explore the many benefits that LLMs offer, we must also tread cautiously, acknowledging their limitations and considering possible mitigation strategies.

For example, one of the most discussed shortcomings of LLMs is their tendency to generate plausible but erroneous information, commonly referred to as hallucinations[2)3)]. It is paramount to be aware of such a limitation in order to utilize the benefits of LLMs in a safe manner. In one instance a lawyer used the famous LLM ChatGPT[4)] as a search engine, not realizing it might generate incorrect information, and filed documents to the court without manually confirming the output.

In the following, we first exemplify the main advantages and capabilities of LLMs, before turning to their most relevant limitations (section 2). With this in mind, we discuss overall characteristics for suitable use cases of LLMs as well as possible mitigation strategies that can be applied right now to reduce the risk posed by the existing limitations (section 3). This includes (1) assessing the risk of a use case to determine the depth of mitigation strategies required (section 3.1), (2) prompting an LLM to deliver its reasoning path in a natural language explanation (section 3.2) and (3) encasing LLMs in a human-centred system design to facilitate safe usage by giving the human user the control and tools required to utilize the LLM in a safe manner (section 3.3).

To further increase the usability of LLMs, we next turn to three technologies currently under development that can increase the safety of using LLMs (section 4): (1) the quality checker (section 4.1) allows us to more

accurately assess the performance of LLMs that go beyond accuracy and can for example measure the safety of a response (e.g. whether a response includes hate speech); (2) the LLM explainer (section 4.2) can explain the LLM generated output by linking phrases of the output to the phrases in the input that generated the output phrase; (3) the fact checker (section 4.3) that allows us to verify if the generated LLM output can be validated based on external trustworthy sources. Finally, we conclude with an executive summary (section 5).

2. Large Language Models (LLMs)

LLMs are a family of neural network models for text processing, generally based on neural networks that implement the Transformer architecture[5]. Unlike past language models trained on task-specific labelled datasets, LLMs are trained using unsupervised learning on massive amounts of data. Their training objective is to predict the next word, given an input prompt. The simplicity of the training objective and the ability to learn on unlabelled data allows scaling these models to ingest a massive amount of data. This training regime combined with scale proves to be sufficient to unlock the model's ability to solve a number of previously unseen tasks and acquire emergent behaviors[6][7]. For instance, LLMs are capable of question answering[8], story generation[9], information extraction[10] and text summarization[11] and many other tasks. More surprisingly, these emergent abilities include creative language generation, reasoning and problem-solving, and domain adaptation[1]. Given the emergent capabilities in multiple tasks, the research community often refers to large pre-trained LLMs as Foundation Models (FM)[12]. This definition reflects the ability to perform multiple tasks with the same model and generalizes the concept to recent developments where foundation models include other modalities beyond text in the prompt, such as images.

2.1 Advantages

LLMs show significant capabilities in many tasks, often outperforming more narrowly focused models. Nonetheless, speaking only of the ability of an LLM to solve a given task would not capture the real key advantage of LLMs. In fact, a key aspect of LLMs is the way they can be leveraged in applications: unlike other machine learning models, LLMs can be used as-is for the target task, without requiring any modification, further training or domain-specific datasets. This removes most of the technical barriers that engineers face when integrating machine learning in their systems, therefore unlocking an unprecedented pace of development and innovation for machine learning applications. In this sense, perhaps the most interesting emergent ability of LLMs is their in-context learning and instruction-following capabilities. That is, users can program the behaviour of an LLM by prompting it with specific natural language instructions. This removes the need to have machine learning expertise to use them, and even enables non-technical users to employ them in interesting applications. For instance, a prompt like "Summarize the following text" is sufficient to specialize the LLM to become a system that provides high-quality text summaries. As such, this ability has given rise to a new type of job referred to as "prompt engineer" which explores how an LLM should be prompted to achieve a desired outcome.

2.1.1 Fluent Text

Underpinning the capabilities of an LLM is its ability to generate fluent text, including in different styles and contexts. LLMs can effortlessly switch from colloquial prose to poetic endeavours, even writing songs and poetries, and formal domain-specific writing, for instance for law documents. Combined with the ability of handling text in multiple languages seamlessly, this unlocks a universal text-interface for any system's input and output. For example, it is possible to write an instruction in English about a text in Spanish while asking the LLMs to provide an answer in German. Likewise, LLMs can be instructed to provide more than fluent text, integrating structure formats in their input and outputs, such as tables, texts formatted with markdown or HTML, etc.

2.1.2 Few-shot Capability

The ability to follow instructions is sometimes called "zero-shot learning", since a task is described but no examples of the task are provided. In case more instruction is required to teach the LLM a task, it is possible to provide a couple of examples as a prompt and the LLM is able to extract the target task from this.

For instance, the following prompt is sufficient to instruct an LLMs to extract the second set of digits, "12465", from the provided record.

"record: 235-32446-abc-d
code: 32446
record: 631-12465-lkj-e
code:"

This capability is generally called "few-shot learning" and it is particularly useful when describing the task in words might be ambiguous or otherwise difficult for the user.

2.1.3 Program Code

LLMs can also handle programming languages in addition to natural languages. This unlocks yet another interesting capability: the generation of working programs. In fact, LLMs exhibit a large set of capabilities related to program code.

(i) They can understand and explain source code.
(ii) They can fix bugs in code snippets.
(iii) They can translate across programming languages and software libraries, for example translating a python function to a C++ function, or modifying a script that uses NumPy to use an equivalent pytorch functions.
(iv) They can generate programs from scratch given an instruction to do so.
(v) They can extend existing programs to complement and complete them according to the provided instructions.

2.2 Limitations

While LLMs have great potential, they also have significant limitations. First, their training is very expensive, therefore they are retrained with low frequency. This makes LLMs unable to keep up to date with recent knowledge. Second, their prompt input and output sizes are generally limited. The input of LLMs is first tokenized, and then provided to the LLM. A *token* can be thought of as a part of a word. For instance, a model might limit to 4k tokens (about 3k-3.5k words) the total size of input plus output, which limits the kind of inputs that can be processed. Finally, LLMs might generate incorrect outputs, a phenomenon sometimes called "hallucination"[2)3)]. In such cases, the LLM-generated answers might be imprecise, incorrect, or even completely made-up, despite appearing as a confident statement at first glance.

2.2.1 Hallucinations

Given a prompt, an LLM will always generate a response that is fluent and confident. However, the generated text is not necessarily correct. Therefore, a major vulnerability of LLMs lies in their tendency to generate fluent but incorrect texts that, at first glance, seem plausible but are actually incorrect and thus referred to as "hallucinations".

For example, given the question "How often did France win the football world cup?", an LLM might confidently answer "France won the world cup once, in 1998." However, in reality, France won the World Cup twice – in 1998 and 2018. This is a form of hallucination where one event was omitted, and therefore, a fluent and confident but wrong answer is returned.

Similarly, we might continue the conversation by saying "but I know for a fact that France won 3 times". The LLM might reply, "I apologise, you are correct, France won 3 times, 1998, 2018 and 1958". This is again a hallucination, this time making up an additional date. Additionally, it showcases a common behaviour of LLMs where they attempt to please the user and conform to their wish.

Hallucinations are particularly dangerous, especially when complex answers are given, mixing facts with false information. The outcome will be that the user will trust the output as a whole and fall into committing the fallacy of "argument from authority". In some cases, such as prompting for a piece of medical advice, the answer is postfixed with the hint to consult a medical doctor. Even though this is a useful hint, it will often be overread due to its generality and loose connection to the factual and nonfactual arguments.

2.2.2 Lack of Complex Reasoning

While LLMs excel at generating human-like text, they often lack common sense understanding. They rely on statistical patterns in the data they were trained on, i.e. they have been trained, given some input, to predict the next tokens (= words). This can lead to factual inaccuracies and illogical responses in certain situations. For this reason, LLMs have also been called "*stochastic parrots*" [13)]. Complex reasoning tasks where LLMs might fail include topics such as multi-step, arithmetic, social, temporal, or multimodal reasoning[14)] due to missing the physical understanding of the world.

2.2.3 Hidden Bias.

LLMs often inherit biases present in the training data, which can perpetuate or even amplify societal biases and stereotypes. This bias can affect the way LLMs generate text and make decisions. For example, many LLMs are predominantly trained on English data and, therefore, are likely to produce outputs that conform to the culture of English-speaking countries. Similarly, if an LLM is trained on, e.g., social media data, then it may exhibit any type of discriminatory views that might have been present in the training data. Addressing bias in LLMs is a significant challenge[15)]; for example, it requires careful curation of training data and ongoing monitoring to mitigate unintended consequences.

2.2.4 A Black Pandora Box

One of the issues that are still under investigated is the hidden harmful capabilities LLMs might have. For one, it's not fully known how safe the documents are on which these models have been trained. This makes us see a trained LLM as a black Pandora box. While the models often refuse to reveal what harmful information they know when prompted, adversarial prompting has been shown to succeed in opening the box and revealing harmful information, such as downloading piracy media and other self-harming content.

3. Safer Usage

Given the advantages but also limitations of current LLMs, two key questions are:

(1) "*What are good application areas?*"
(2) "*What can we do to facilitate a safer usage of current LLMs?*"

Subsequently we look at three options to address question (2), which will include assessing the risk of an LLM use case (section 3.1), asking an LLM to generate its reasoning in a natural language explanation (section 3.2) and embedding the LLM in a human-centric system (section 3.3).

For question (1), "What are good application areas?", it is important to keep in mind that LLMs are applicable without further data ingestion or application-specific training. In this large space of applications, how to select those where LLMs could be readily successful? Here, we need to consider that LLMs are great creative writers but might produce incorrect information. That is, an LLM has immediate applicability in applications where correctness is not necessarily a problem (e.g., fictional writing) or where it is mitigated by an active engagement with humans.

For instance, the example of an LLMs that helps with law text is a clear case in which correctness is needed, and at the same time it could work as an example where such concerns could be bypassed. In fact, in many applications humans already build intermediate checks, recognizing that also human experts can make mistakes. For these applications, LLMs can undertake the initial text generation task and be paired seamlessly with the human experts and within the original application's workflow. In the example of a law text, where we would have had two lawyers interact to check the correctness of a text, we could now envision the addition of an LLMs that performs most of the "heavy-lifting" while reducing the human work to a hopefully simpler error checking tasks. In the remainder of this section, we provide some intellectual tools to better reason about the suitable application areas for LLMs.

3.1 Risk Classification

The best mitigation strategy for the safe usage of an LLM can heavily depend on the use case for which the LLM is to be deployed. For example, using an LLM for book recommendations would be a low-risk use of an LLM because it is not very detrimental in case the LLM hallucinates a book that does not actually exist. In contrast, using an LLM to generate a medical report for a patient comes with a high risk — if the medical report contains a hallucination and a doctor makes a decision based on this report, it might lead to an incorrect and even dangerous treatment for the patient. Therefore, it is paramount to first assess the risk of a use case and based on this develop an appropriate mitigation strategy.

To determine the risk level of a use case, we can employ for example the risk definition outlined by the upcoming Artificial Intelligence act[16] of the European Union. The EU AI act foresees that AI usage will be categorized into one of four risk categories. Each risk category will have different implications on the usage and checks that needs to be run on the AI system prior to deployment. An overview for this can be found in **Fig. 1**. The four categories are:

(1) Minimal risk - users have to be informed about the usage of AI and have the option to opt out.
(2) Limited risk - transparency is required.
(3) High risk - a conformity assessment needs to be run before an AI system is allowed to be deployed within the EU.

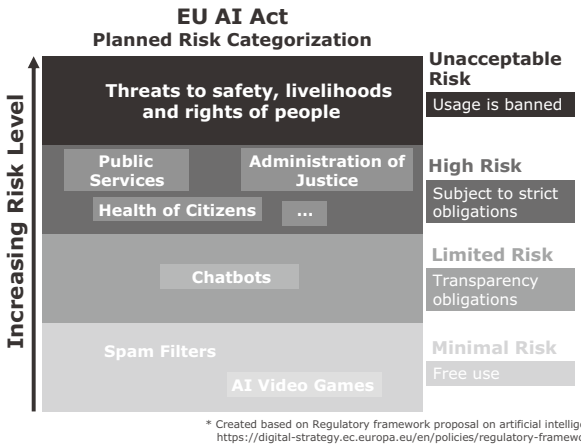

Fig. 1 The EU AI act categorizes AI applications according to risk levels.

(4) Unacceptable risk - AI is not allowed to be used for such use cases.

3.2 Natural Language Explanations

One mitigation strategy can be to simply ask the LLM to generate a reasoning path along with its response: in the prompt to the LLM we can include a request where we explicitly ask the LLM to provide us with a natural language explanation behind its reasoning. For example, consider the below scenario, where we ask the LLM to solve a text-based math problem: "I have 3 apples in my basket. I go to the supermarket, buy 6 apples and add them to my basket. On the way home I meet two friends and give them 1 apple each. Once home, how many apples I have left in my basket?"

Based on this prompt, an LLM might provide an incorrect answer: "You would have 4 apples left in your basket", whereas the correct answer would be "7". As the human reader, we don't know why the LLM provided this answer and our only option for verification is to solve the problem ourselves (which would obliterate the benefit of asking the LLM in the first place). However, with a simple change to the input, we can ask an LLM to generate the reasoning alongside the solution. For instance, we can add the phrase "Let's think step-by-step" to the original input to obtain the reasoning path. In the case of our example, we receive the following response: "You start with 3 apples in your basket. You buy 6 more apples at the supermarket and add them to your basket, bringing the total to 3 + 6 = 9 apples. On the way home, you give 1 apple to each of 2 friends, leaving you with 9 - 2 = 7 apples remaining in your basket."

In addition to having the reasoning path available, we also have the advantage that now the LLM arrives at the correct answer: by iteratively prompting the model with the most recent reasoning step, we implicitly guide the LLM to produce the correct answer.

One potential drawback of this approach is that this technique of engineering a prompt is currently very brittle: this means that adding a certain phrase cannot guarantee us a correct answer and changing the phrase slightly (e.g., by simply adding an exclamation mark at the end of the phrase) could already lead to an incorrect result again.

3.3 Human-Centred Approach

With their current shortcomings and their generally statistic nature, it is impossible to ensure that an LLM works 100% percent correctly. Therefore, the question arises of what can be done to increase the safe usage of an LLM. Here we outline that if we design an application, in which an LLM is embedded, in a human-centric manner[17], we can increase the amount of control a human has and therefore encourage safe usage.

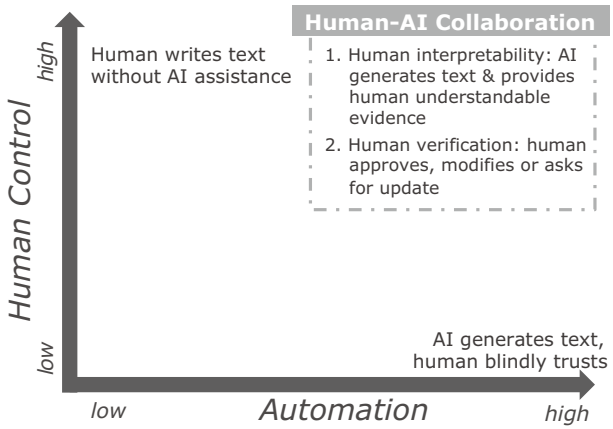

Fig. 2 The safety of LLM applications by ensuring that human users are given the right level of control.

Typically, the amount of automation a computer application offers is seen on a singular axis ranging from low to high automation. But this view can be extended: we add an additional axis that embodies the amount of human control we give to a use[17]. Following this paradigm and applying it to LLMs[18], leads to the following possible scenarios (**Fig. 2**). In the scenario without LLMs, humans are fully in control of writing texts. In the scenario, were LLMs generated text that is blindly trusted by the human user, leading to a loss of control and therefore fully exposing the user to the current limitations and consequent dangers of LLMs. We can mitigate this in the third and final scenario: by offering humans the right tools, we can give them back control.

We can give control over the generated LLM text to the human user by providing means for human interpretation and human verification. For human interpretation we supply human-understandable evidence for the LLM-generated output. For this scenario, we have already seen the option to ask the LLM to supply a natural language explanation. We can also consider a new type of emerging LLMs which are retrieval-based[19][20], i.e. they first retrieve a relevant text passage before generating their answer. However, so far their performance is still subpar and therefore not yet widely used.

To enable human verification, we give the human users tools that enable them to verify the content generated by the LLM. We introduce 3 technologies in the following.

4. Safer Technology

We can empower the human user of LLM technology to verify the LLM output and therefore enable a safer usage of LLMs. In the following, we introduce three technologies that aid this goal (**Fig. 3**).

First, we introduce the quality checker (section 4.1): it can be run before an LLM is deployed to check if its accuracy is good enough. The term accuracy can be interpreted in different ways, for example, we can employ metrics that measure how safe or factual a set of outputs are. This allows us to ensure a minimum quality and we can compare different LLMs to choose the one best suitable for a use case.

Second, the LLM explainer (section 4.2) can be run after an LLM generated a text: it links phrases in the generated text back to the input source that was given to the LLM. For example, in the case of text summarization, this allows us to understand which passage in the original document led to the LLM generating a certain summary sentence or phrase. With this technology it would therefore be possible for e.g. a doctor to efficiently verify the correctness of a medical report, therefore freeing up the doctors' time from writing reports by herself to attend to other tasks.

Third, the fact checker (section 4.3) can be used to validate the LLM-generated text against an external (trustworthy) source. With this, we can warn the user of potential hallucinations. In a different scenario, this technology could also be used to identify fake news.

4.1 Quality Checker

LLMs are not immune to errors or biases, and these shortcomings can lead to detrimental consequences, especially when used in high-risk domains (e.g., legal, medical; aligned with EU AI Act as shown in Fig. 1). Therefore it is paramount to rigorously evaluate an LLM for a particular use case before deployment. The evaluation of LLMs typically revolves around two central aspects:

(i) The selection of appropriate datasets for assessment.

(ii) The establishment of an evaluation methodology.

The former involves identifying suitable benchmarks for evaluation, while the latter entails defining evaluation criteria for both automated and human-centric assessments[21].

Within the context of high-risk domains, the complexities and potential ramifications associated with LLM usage underscore the need for a more comprehensive and critical evaluation process. Specific challenges emerge when evaluating LLMs[22]. For example, domains such as law require constant updates to stay relevant[23]. In the healthcare field, the safety-critical nature of decisions severely restricts current applications due to the potential for harmful consequences caused by the high possibility of hallucinations[24].

This highlights the vital importance of addressing both factual accuracy and safety concerns when evaluating LLM performance. In our previous work, Hung et al. investigated how well domain-adaptive instruction-tuned LLMs perform in high-risk domain tasks (i.e., question answering and summarization) in legal and medical fields. For this we created a quality checker that measures the performance of different LLMs on various high-risk test sets with regards to state-of-the-art metrics that can measure "factuality"[25][26] and "safety" [27][28]. The findings revealed a significant gap in the suitability of LLMs for high-risk domain tasks, suggesting that the use of LLMs in their current state is *not yet* practical unless carefully embedded in a human-centric application.

Overall, the quality checker with the currently implemented metrics can give an indication of how well an

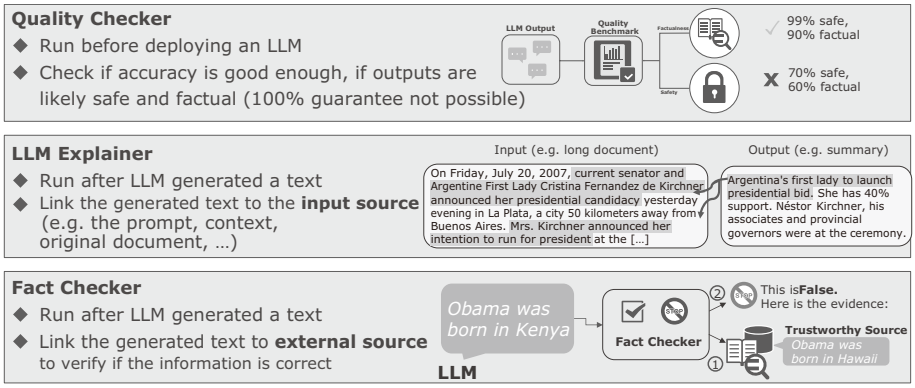

Fig. 3 An overview of the three technologies currently under development to make the usages of LLMs safer.

LLM performs and which LLM might be the best for a use case. But further effort needs to be made (i) to define evaluation metrics tailored to specific domain applications[29]; and (ii) to investigate with domain experts how to best assess the accuracy of model outputs and address safety concerns[30]. Therefore, at this point in time, the quality checker should also be paired with other mitigation strategies, such as designing human-centric applications.

4.2 LLM Explainer

LLMs are prone to hallucinations, which makes them difficult to use directly in high-risk domains like medicine. Consider, for example, a summary of a medical instruction, where the LLM produces the following sentence: The patient should take 50mg pill of Drug X, three times a day. However, when consulting the original document, it turns out that this is a hallucination because the patient is recommended to take 5mg pill of Drug X, once a day. In this example, the summary's hallucination recommends the patient to take "30 times" higher dosage of the drug. If the patient blindly trusts the LLM and follows the instruction in the generated summary, it could lead to serious consequences for the patient (e.g., lethal overdosing).

To avoid such problems, our LLM explainer can create links between the generated LLM text and the original input. Here, we assume that the information the LLM explainer should link to, is already part of the input query to the LLM. For example, our explainer can map sentences from an LLM summary to their original source (Fig. 3, middle). This linking allows users to efficiently verify that the generated information from the LLM is correct.

In a similar manner, we can also highlight which information might be present in the original input but missing in the summary.

The explainer can also be used for other tasks where we have access to the original text. For example, in the case of question-answering, where the user provides a question and relevant input text based on which the question should be answered. With the explainer, we can map the generated answer back to the text referring to this answer in the input text.

Overall, the explainer is a tool that allows the user to understand which input phrases cause the generation of which output phrases. This allows the user to verify the correctness of the output, therefore enabling the use of LLMs in high-risk domains, where users can save time by outsourcing content generation to an LLM but require verification of the output for safety reasons. In contrast, in some use cases, the relevant information is not provided alongside the input query.

In such scenarios, the LLM explainer is not directly applicable because the LLM accesses its internal knowledge instead, which is not explicitly represented. For these scenarios, we can instead turn to our fact checker, which can verify LLMgenerated information by comparing the information to a set of trustworthy sources.

4.3 Fact Checker

Automated fact-checking can enable the prompt and accurate identification of false or misleading information. With the rise of LLM usage in a variety of domains and applications, fact-checking has emerged as a critical tool that allows researchers and users alike to detect falsehoods and hallucinations generated by the models. Additionally, fact-checking in the traditional sense is important in analysing social media posts and debunking fake news at a time when spreading misinformation is easier and more consequential than ever.

The pipeline for automated fact-checking can be described as follows: A given text is broken down into phrases and the fact checker first identifies which phrases require fact-checking (e.g. "Dear ladies and gentlemen" does not need to be fact- checked). Second, if a phrase should be checked, it becomes a "claim" and the system retrieves relevant reference documents related to the claim. Based on this, in the third and final step, the fact checker determines whether the claim is true or false.

Standard fact checkers cannot explain why a certain claim is classified as true or false. This implies that a human user would still have to run a verification by herself in order to be able to trust the classification, therefore significantly reducing the benefit of the fact checker. Instead, we can increase the usability of the system if the fact checker can also output its reasoning path. For an example **Fig. 4**, where we identify that the claim "NEC was established by a University of Tokyo graduate in 1899" is wrong because Kunihiko Iwadare graduated from the Imperial College of Engineering, not the University of Tokyo. By giving a reasoning path, a human can understand the reason behind the fact checker decision.

Supplying a reasoning path is an important first step. Next, we plan to further improve such fact checkers in the following two aspects. First, the reasoning path in current systems[31] supply the evidence document that was used to determine if a claim is true or false, however, they do not explain why or in what way the evidence refutes or supports the claim. Adding such an explanation will further enhance the usability of the system by speeding up the verification process for the human user. Second, many existing benchmarks for fact-checking

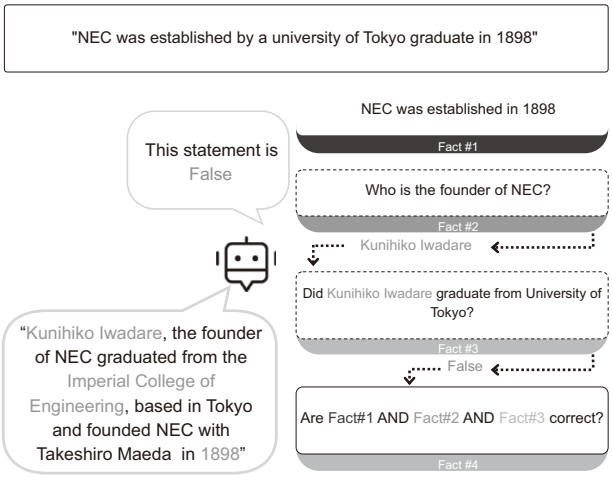

Fig. 4 A fact checker example.

are often limited in terms of domains and usability (e.g. many benchmarks assume that the correct evidence document is already given, whereas in reality, it might have to first be found). Engaging with real users with a need for a fact checker can help us to move towards more realistic settings.

5. Executive Summary

LLMs are revolutionizing our world. They have impressive capabilities that include (1) writing fluent text; (2) being able to learn a new simple text-based task with a few demonstrations and (3) writing program code snippets. Next to their advantages, we discussed some current limitations, which include:

(1) LLMs can hallucinate, which means they produce fluent and confident-sounding text that is actually wrong.
(2) LLMs lack world knowledge and commonsense, therefore they are not able to perform any complex reasoning.
(3) LLMs are trained on existing textual data and therefore contain and potentially amplify the bias of this data.

Based on this initial assessment, we turned to the question of how the safe usage of current LLMs can be increased. For this, we identified three approaches:

(1) We can classify the risk level of a use case that will use an LLM and this can inform us how much care needs to be taken to ensure a reliable and safe LLM response.
(2) We can modify how we prompt the LLM for an answer, for example, adding "Let's think step-by-step" may enable the LLM to generate a reasoning path in natural language.
(3) We can take care how we design the LLM application in such a way that the human retains final control. For example, with our LLM explainer, e.g., medical doctors can get the help of an LLM to write reports quicker but also in a safe manner – by using our explainer to verify the output efficiently.

Finally, we discussed three technologies currently under development that will further aid in making the use of LLMs safer.

(1) The quality checker can be used to measure how safe or factual LLMs perform for a certain use case. This allows us to choose the best LLM given a set of options and to measure if it performs well enough yet. Our initial prototype has been tested in English for the medical and legal domains. In the future we plan to make the quality checker more precise by creating more specific measurements, such as how to measure if a response is safe in a medical domain, and by extending these measurements to work for other languages.
(2) The LLM explainer can trace the output generated by an LLM back to the input prompt that was given to the LLM. With this we can for example use LLMs for summarization more safely. Without our explainer, the summary might contain wrong information. With our explainer, users can quickly and efficiently verify that the information in the summary is correct and if anything important is missing. This enables the use of LLM in high-risk domains, such as writing medical reports.
(3) The fact checker can be used to automatically find contradictions between texts. The input text can either be generated from an LLM or also written by a human. Our fact checker compares the provided text to a reference database with trustworthy text sources and provides a warning if a contradiction of the text with the trustworthy source is detected. With this, we can for example detect fake news.

Overall, LLMs hold immense promise, offering us a glimpse into a future where understandable and supportive AI systems extend human capabilities. From this, human-computer collaborations can arise that open new, previously unsought possibilities. As we harness the LLM potential to enhance productivity, we must also remain vigilant, understanding that, like any transformative technology, LLMs carry inherent limitations and necessitate responsible usage. By embracing the remarkable abilities of LLMs while respecting their limitations, we can collectively steer the course to move toward a brighter, more equitable future.

* ChatGPT is a trademark of OpenAI.
* All other company names and product names that appear in this paper are trademarks or registered trademarks of their respective companies.

References

1) Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean and William Fedus: Emergent Abilities of Large Language Models, Transactions on Machine Learning Research (TMLR), 2022
https://arxiv.org/abs/2206.07682
2) Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro: Factuality Enhanced Language Models for Open-Ended Text Generation, Neural Information Processing Systems, 2022
https://arxiv.org/abs/2206.04624
3) Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto and Pascale Fung: Survey of Hallucination in Natural Language Generation, ACM Computing Surveys, Volume 55, Issue 12, pp.1-38, 2023
https://doi.org/10.1145/3571730
4) OpenAI: Introducing ChatGPT, June 2023.
https://openai.com/blog/chatgpt
5) Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin: Attention Is All You Need, Neural Information Processing Systems, 2017
https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
6) Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell et al.: Language Models are Few-Shot Learners, Neural Information Processing Systems, 2020
https://arxiv.org/abs/2005.14165
7) Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo and Yusuke Iwasawa: Large Language Models are Zero-Shot Reasoners, NeurIPS2022, 2022
https://arxiv.org/abs/2205.11916
8) Md Adnan Arefeen, Biplob Debnath and Srimat Chakradhar: LeanContext: Cost-Efficient Domain-Specific Question Answering Using LLMs, 2023
https://arxiv.org/abs/2309.00841
9) Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Mohammad Shoeybi and Bryan Catanzaro: Factuality Enhanced Language Models for Open-Ended Text Generation, The 36th Conference on Neural Information Processing Systems (NeurIPS), 2022
https://arxiv.org/abs/2206.04624
10) Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Jiaqi Mu, Hao Zhang and Nan Hua: LMDX: Language Model-based Document Information Extraction and Localization,2023
https://arxiv.org/abs/2309.10952
11) Haopeng Zhang, Xiao Liu and Jiawei Zhang: SummIt:

Iterative Text Summarization via ChatGPT, 2023
https://arxiv.org/abs/2305.14835
12) 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.: On the Opportunities and Risks of Foundation Models, 2021
https://arxiv.org/abs/2108.07258
13) Emily M. Bender, Timnit Gebru, Angelina McMillan-Major and Shmargaret Shmitchell: On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?, The 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, pp.610-623, 2021
https://dl.acm.org/doi/10.1145/3442188.3445922
14) Wenting Zhao, Mor Geva, Bill Yuchen Lin, Michihiro Yasunaga, Aman Madaan and Tao Yu: Acl 2023 tutorial: Complex Reasoning in Natural Language. ACL 2023, pp.11-20, 2023
https://aclanthology.org/2023.acl-tutorials.2/
15) Zeerak Talat, Aurélie Névéol, Stella Biderman, Miruna Clinciu, Manan Dey, Shayne Longpre, Sasha Luccioni, Maraim Masoud, Margaret Mitchell, Dragomir Radev, Shanya Sharma, Arjun Subramonian, Jaesung Tae, Samson Tan, Deepak Tunuguntla and Oskar Van Der Wal: You reap what you sow: On the Challenges of Bias Evaluation Under Multilingual Settings, BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models, pp.26-41, 2022
https://aclanthology.org/2022.bigscience-1.3/
16) European Commission: Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEG-ISLATIVE ACTS, 2021
https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0206
17) Ben Shneiderman: Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy, International Journal of Human–Computer Interaction, Volume 36 Issue 6, pp.495-504, 2020
https://www.tandfonline.com/doi/full/10.1080/10447318.2020.1741118
18) Chia-Chien Hung, Wiem Ben Rim, Lindsay Frost, Lars Bruckner and Carolin Lawrence: Walking a Tightrope – Evaluating Large Language Models in High-Risk Domains, EMNLP 2023 Workshop on Benchmarking Generalisation in NLP (GenBench), 2023
https://arxiv.org/abs/2311.14966
19) Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat and Mingwei Chang: Retrieval Augmented Language Model Pre-Training, The 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pp.3929-3938, 2020
https://proceedings.mlr.press/v119/guu20a.html
20) Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen and Laurent Sifre: Improving Language Models by Retrieving from Trillions of Tokens, The 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pp.2206-2240, 2022
https://proceedings.mlr.press/v162/borgeaud22a.html
21) Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang et al.: A Survey on Evaluation of Large Language Models, 2023
https://arxiv.org/abs/2307.03109
22) Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu and Robert McHardy: Challenges and Applications of Large Language Models,2023
https://arxiv.org/abs/2307.10169
23) Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky and Daniel Ho: Pile of Law: Learning Responsible Data Filtering from the Law and a 256GB Open-Source Legal Dataset, Neural Information Processing Systems 35 (NeurIPS 2022), pp29217-29234, 2022
https://proceedings.neurips.cc/paper_files/paper/2022/hash/bc218a0c656e49d4b086975a9c785f47-Abstract-Datasets_and_Benchmarks.html
24) Sandeep Reddy: Evaluating large language models for use in healthcare: A framework for translational value assessment, Informatics in Medicine Unlocked, Volume 41, Article.101304, 2023
https://www.sciencedirect.com/science/article/pii/S2352914823001508
25) Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu and Caiming Xiong: QAFactEval: Improved QA-Based Factual Consistency Evaluation for Summarization, The 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.2587-2601, 2022
https://aclanthology.org/2022.naacl-main.187/
26) Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji and Jiawei Han: Towards a Unified Multi-Dimensional Evaluator for Text Generation, The 2022 Conference on Empirical Methods in Natural Language Processing, pp.2023-2038, 2022
https://aclanthology.org/2022.emnlp-main.131/
27) Laura Hanu and Unitary team: Detoxify
https://github.com/unitaryai/detoxify
28) Emily Dinan, Gavin Abercrombie, A. Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau and Verena Rieser: SafetyKit: First Aid for Measuring Safety in

Open-Domain Conversational Systems, The 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.4113-4133, 2022
https://aclanthology.org/2022.acl-long.284/

29) Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi et al.: Survey on Factuality in Large Language Models: Knowledge, Retrieval and Domain-Specificity, 2023
https://arxiv.org/abs/2310.07521

30) Xiang'Anthony' Chen, Jeff Burke, Ruofei Du, Matthew K. Hong, Jennifer Jacobs, Philippe Laban, Dingzeyu Li, Nanyun Peng, Karl D. D. Willis, Chien- Sheng Wu et al.: Next Steps for Human-Centered Generative AI: A Technical Perspective, 2023
https://arxiv.org/abs/2306.15774

31) Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan and Preslav Nakov: Fact-Checking Complex Claims with Program-Guided Reasoning, The 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.6981-7004, 2023
https://aclanthology.org/2023.acl-long.386/

Authors' Profiles

LAWRENCE Carolin
Manager & Chief Research Scientist
NEC Laboratories Europe

BIFULCO Roberto
Senior Manager
NEC Laboratories Europe

GASHTEOVSKI Kiril
Senior Research Scientist
NEC Laboratories Europe

HUNG Chia-Chien
Research Scientist
NEC Laboratories Europe

BEN RIM Wiem
Research Scientist
NEC Laboratories Europe

SHAKER Ammar
Senior Research Scientist
NEC Laboratories Europe

OYAMADA Masafumi
Research Fellow and Group Head
Data Science Laboratories

SADAMASA Kunihiko
Professional
Data Science Laboratories

ENOMOTO Masafumi
Data Science Laboratories

TAKEOKA Kunihiro
Special Researcher
Assistant Manager
Data Science Laboratories

Information about the NEC Technical Journal

Thank you for reading the paper.
If you are interested in the NEC Technical Journal, you can also read other papers on our website.

Link to NEC Technical Journal website

Japanese

English

Vol.17 No.2 Special Issue on Revolutionizing Business Practices with Generative AI
– Advancing the Societal Adoption of AI with the Support of Generative AI Technologies

Remarks for Special Issue on Revolutionizing Business Practices with Generative AI
Approaches to Generative AI Technology: From Foundational Technologies to Application Development and Guideline Creation

Vol.17 No.2
June 2024

Special Issue TOP

Papers for Special Issue

Market Application of Rapidly Spreading Generative AI
NEC Innovation Day 2023: NEC's Generative AI Initiatives
Streamlining Doctors' Work by Assisting with Medical Recording and Documentation
Using Video Recognition AI x LLM to Automate the Creation of Reports
Understanding of Behaviors in Real World through Video Analysis and Generative AI
Automated Generation of Cyber Threat Intelligence
NEC Generative AI Service (NGS) Promoting Internal Use of Generative AI
Utilization of Generative AI for Software and System Development
LLMs and MI Bring Innovation to Material Development Platforms
Disaster Damage Assessment Using LLMs and Image Analysis

Fundamental Technologies that Enhance the Potential of Generative AI
NEC's LLM with Superior Japanese Language Proficiency
NEC's AI Supercomputer: One of the Largest in Japan to Support Generative AI
Towards Safer Large Language Models (LLMs)
Federated Learning Technology that Enables Collaboration While Keeping Data Confidential and its Applicability to LLMs
Large Language Models (LLMs) Enable Few-Shot Clustering
Knowledge-enhanced Prompt Learning for Open-domain Commonsense Reasoning
Foundational Vision-LLM for AI Linkage and Orchestration
Optimizing LLM API usage costs with novel query-aware reduction of relevant enterprise data

For AI Technology to Penetrate Society
Movements in AI Standardization and Rule Making and NEC Initiatives
NEC's Initiatives on AI Governance toward Respecting Human Rights
Case Study of Human Resources Development for AI Risk Management Using RCModel

NEC Information

2023 C&C Prize Ceremony