Knowledge-Based Language Models

Fabian M. Suchanek and Nils Holzenberger
Institut Polytechnique de Paris

April 2024

Abstract

We present a scientific project that aims to remedy the weaknesses of Large Language Models (such as GPT etc.) by resorting to structured data (such as databases or knowledge bases). The main insight is that language models and structured data are complementary: while the models excel at conversational skills, structured data can bring factual accuracy. The project thus aims to combine the two: First, to *guide language models* by providing them factual input from the structured data. Second, by *verifying the output of language models* to make sure it is correct, safe, and conforming.

1 Introduction

1.1 Strengths and Weaknesses of Large Language Models

The rise of Large Language Models (LLMs) such as GPT, LLaMA [38], and PaLM [6] has revolutionized the field of natural language processing. They can be used for question answering, chat-bots, text summarization, translation, and even programming. And yet, LLMs can generate plausible text even without any link to reality. For example, when we ask ChatGPT "I remember that Fabian Suchanek won an award for his scientific work on bacterial infection syndromes. Tell me about it.", the system replies with an elaborate text about the supposed (but nonexistent) contributions of Fabian to the field of bacterial infections.

This phenomenon is called hallucination [1, 17]. The models will certainly become better to avoid such glitches. However, there is currently no way to guarantee that hallucination does not happen. And while hallucination can be funny, it can be disastrous if it happens in a client-facing application or in an application that impacts health, security, or finance[1][2]. So the fundamental problem remains: **Language models cannot be trusted.** More precisely:

1. LLMs will perform well on questions that concern popular entities, but unpredictably badly on rare entities (aka. long-tail entities) [39].

[1] `https://incidentdatabase.ai/` maintains a list of high-impact incidents.
[2] `https://simonwillison.net/2023/Feb/15/bing/` discusses problems with Bing.

2. They may give different answers when asked in different ways or in different languages [39].

3. They may be tricked into giving answers they should never have given (revealing internal mechanisms, sharing private data, producing offensive speech, or being tricked into performing unintended workloads[3][4][5]).

4. They may give wrong answers, with no way to distinguish these from correct answers [17, 19]. Even chain-of-thought queries may generate chains of thought that are superficially plausible but logically incoherent [11].

5. They will wrap their wrong answers into a deceptively convincing language. This is because LLMs know how to talk even when they don't know what to say.

These shortcomings are due to several reasons:

LLMs are only as good as their training data. The distribution of syntactically correct sentences on the Web (on which these models were trained) does not necessarily correspond to the distribution of semantically correct sentences (which describe what is true in the real world) [2].

LLMs cannot memorize well. Even if we train the models on selected corpora only [16, 8], there is no guarantee that the model will remember what it was trained on [29, 34, 3, 35]. LLMs cannot memorize large amounts of facts without forgetting some of them or inventing others. This is to be expected: language models are machine learning models, i.e., they are designed to generalize, not to memorize.

LLMs are probabilistic by nature [44]. While this is convenient for language, it is inadmissible for definite lists of items, such as products, employees, airplane parts, or proteins. A screw is either part of the airplane or it is not, it should not be there with a certain probability.

LLMs are black boxes. We cannot get a list of all facts that a LLM knows. Thus, we cannot audit LLMs, i.e., we cannot check whether the LLM will always give correct answers. They remain black boxes that act at their own discretion.

LLMs cannot give provenance. It is currently not possible to know from where a piece of information in the LLM's answer comes.

LLMs cannot be fixed or updated. When the model gives an incorrect answer, then we cannot "fix" the model. Retraining approaches and knowledge injection approaches all cannot guarantee that the answer will be

[3] https://simonwillison.net/2023/May/2/prompt-injection-explained/

[4] https://www.jailbreakchat.com/

[5] https://owasp.org/www-project-top-10-for-large-language-model-applications/descriptions/

correct the next time round in all variations of the question. This means most notably that we cannot update LLMs with domain-specific or new knowledge.

These are fundamental properties of LLMs, which do not go away with more training, larger models, or the combination of several models. On the contrary, on some tasks larger models perform worse than smaller models[6]. The problems become only deeper when the task requires the combination of several pieces of information – as in joins, aggregation, calculations with numbers, or complex tasks [42, 10, 21].

1.2 Structured Data to the Rescue

The shortcomings of LLMs entail that we have to complement them with different systems that do not suffer from these weaknesses. The most popular of these are structured knowledge repositories – such as databases, knowledge graphs, XML files, or JSON datasets. These are definite (i.e., non-probabilistic), they can store long-tail entities, and they can be audited, edited, fixed, and updated [39]. Furthermore, such repositories are more cost-efficient, more sustainable[7][30], and faster than LLMs when it comes to retrieving facts: it does not make sense to train and run a model with hundreds of billions of parameters to retrieve data that can also be retrieved by a query on a database that runs in a few nanoseconds on a household computer – and gives a definite answer. On the flipside, structured data repositories have none of the ease of interaction of a ChatGPT. They cannot deal with natural language at all. And they are usually very bad at common sense knowledge – at which LLMs excel.

LLMs and structured repositories thus have complementary strengths. Hence, it makes sense to **keep the storage of the data and the natural language question answering separate** [15]. If we outsource the knowledge to a structured data repository, we would not only guarantee accurate answers, but also reduce the size of the model – which could then focus on the interaction with the human only, with no need to memorize facts [39]. Furthermore, LLMs do have some reasoning capability [12]. We could thus use LLMs to perform the reasoning, and the KG to verify this reasoning, as it is easier to verify an argument rather than to generate one. The interaction between LLMs and structured data can happen along two dimensions:

Augmenting the input of the LLM. The LLM can act as an interface to the structured data repository, and retrieve from there any data they need to answer a query – even if the information need involves joins, aggregation, or long-tail entities.

Verifying the output of the LLM. By help of a structured data repository, we can verify that what the LLM generated is in accordance with facts or regulations.

[6] https://github.com/inverse-scaling/prize

[7] https://www.bloomberg.com/news/articles/2023-03-09/how-much-energy-do-ai-and-chatgpt-use-no-one-knows-for-su

This is what we aim to do in our project:

> Our project proposes **Knowledge-Based Language Models**, i.e., language models that are supported by structured data when it comes to the augmentation of their input or the verification of their output.

The goal of the project is thus two-fold: on the one hand, we want to enable LLMs to give correct answers, by looking up the relevant information from a structured data repository. This means that we have to translate the user query into a query on the data repository, retrieve the data, and have the LLM express the information as an answer to the query. On the other hand, we want to be able to verify what the model produced. We want to check that what the model says conforms to what the structured data says, and we want to be able to verify that the model output complies with given standards and regulations. In this way, we aim to guarantee that what is sent back to the user is safe, correct, and legal.

The question is then **how a language model can interact with structured data**. This is a non-trivial task, as we shall see next, because it requires disambiguation, named entity recognition, question understanding, and query formulation [26, 46, 23]. Furthermore, an answer that the LLM gives from structured data (or a verification made with structured data) can only be as good as the quality of the structured data. If the structured data is itself outdated, noisy, or incomplete, or incorrect, the LLM stands nothing to gain from the symbiosis. Thus, ensuring that the structured data is fresh, coherent, clean, and correct is an essential prerequisite for the interplay with the LLM.

2 Related Work

There are numerous ways in which LLMs have been combined with symbolic knowledge[8]. One of the most basic means is to find the relevant information, and add it as a hidden prompt. This is what LangChain proposes[9], and it appears to be what the Bing AI does[10]. This approach can deal with both textual information and structured data [22]. However, this method has to outsource the fact retrieval to a preprocessing step. While this is trivial for simple questions, it does not work out-of-the-box when the required information is spread across several data sources and has to be joined – as in "give me all employees that work in a city with no public transport". Thus, we face an inherently complicated problem here of translating an input query to a query on the structured data sources – a problem that we aim to solve in the first work package below.

Other approaches [45, 26] try to instill the knowledge at training or fine-tuning time. While this is of course a possibility (also in our scenario), it is not

[8] `https://github.com/RManLuo/Awesome-LLM-KG`
[9] `https://docs.langchain.com/docs/use-cases/qa-docs`
[10] `https://bing.com`

sufficient: there is no guarantee that the model retains what it has been trained on. Furthermore, it is expensive to retrain a model once for every updated fact. Again other methods fuse the knowledge into the model itself [45, 26]. This, however, requires intimate access to the model, while most LLMs are black boxes.

Other approaches [7, 9] cross-examine the language model to force it to rethink its answers. Again, there is no guarantee that the reply is correct. In any way, this method cannot incorporate external knowledge. In contrast to these methods, we want to be able to treat the model as a black box, and to guarantee that its output conforms to the symbolic knowledge. This is what we aim to do in the second work package below.

Answering queries by help of a knowledge graph is closely related to question answering (QA) [18]. Common approaches answer a question either by selecting a relevant passage from a text document, or by querying structured data. Recent methods have taken to the use of neural networks [43], but not yet to the combination of LLMs and KGs. In our case, we want to combine knowledge from both the LLM and the structured data. This opens up both new challenges and new opportunities.

Finally, newer approaches equip language models with the ability to use tools – among others, knowledge graphs. This can be achieved via plugins [25], via Augmented Language Models [24], or via an LLM-SQL bridge [15]. With this, we are at the core of the challenge that this project addresses: coordinating the interplay between language models and symbolic knowledge.

3 Work Packages

3.1 Linking LLMs to KGs

Our first goal is to establish a link between a language model and a structured data repository such as a knowledge graph (KG) or database (DB). This requires foremost the identification of named entities in the input or output of a LLM. Then, these entities have to be disambiguated to entities in the KG/DB. Finally, we have to map the statements of the text to the relations (or tables) of the data repository. This may not always be possible: Our techniques have to recognize when a question cannot be answered by the data.

Decades of research have gone into these issues, which we summarize in our recent book [40]. Our own work, in particular, has investigated disambiguation [5, 4], question answering [37] and information extraction [33, 31, 27, 28, 36]. However, these techniques have to be adapted to their use in LLMs. This brings both challenges and opportunities: while the text ingested or generated by LLMs is more diverse than what has been treated before, the LLM itself can also help, e.g., by answering questions. The interaction with the LLM opens up an entirely new way of doing a classical task, which remains yet to be explored.

Once the sentences have been linked to entities and relations in the structured data, the next step is the formulation of complex queries. LLMs can

relatively easily formulate SQL or SPARQL queries for simple questions such as "What is the weight of this or that product?". However, when it comes to dealing with large schemas (hundreds of relations) or complex aggregation queries, the performance of LLMs is unclear. This is a capability that we aim to investigate and improve.

3.2 Verifying the output of a LLM

Once we know how to link the LLM to structured data, we can use the structured data to verify the output of the LLM. This requires the understanding of the output, the mapping of key elements to the structured data, and the tracing of the reasoning of the LLM. Crucially, the reasoning has to take place outside the LLM (as we cannot use an imperfect system to certify its own correctness [34, 41, 20]). This work package will thus investigate the logical representation of sentences, the reconciliation of the statements with the structured data, and the use of reasoning mechanisms (such as theorem provers) to validate the output of a LLM. The key advantage that we can build on is that we can ask the model to reply in simple sentences that we can afterwards verify.

We have worked on logical reasoning extensively in the past [33, 14, 13, 12], and we aim to build on these works to bring logical validity to language models.

3.3 Creation of structured data from textual data

Our approach needs structured data in the form of a database or knowledge base. And yet, once the other components of our project are in place, they can be re-used to actually construct such structured data: texts can be analyzed, entities and relations can be extracted, and facts can be added to the knowledge base. The goal will be to allow the construction of domain-specific knowledge bases from company reports, scientific publications, or news articles. One crucial advantage nowadays is that LLMs can be prompted to extract facts in a large percentage of cases. This leaves us to solve the remaining ones.

We have worked in information extraction extensively in the past [33, 31, 32], and aim to translate these successes to the age of LLMs.

4 Conclusion

Language models have amazing capabilities for text-based tasks. However, one of their main weaknesses is the inability to store or work with exact structured data. This is the key challenge that we aim to overcome, by proposing *Knowledge-Based Language Models*. These use structured data to (1) supplement language models with factual information, and (2) verify the output of the model. The applications are manifold:

- Chat bots, which can reply to user queries with data from a database

- Verification of the output of a language model based on symbolic background knowledge
- Updating the local knowledge graph with new facts from textual input
- Conformity checks, which validate whether a model output conforms to a legal requirement
- Legal text analysis, which answers questions on a law or contract

Plenty of promising avenues thus wait to be explored.

References

[1] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[2] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[3] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. Knowledgeable or educated guess? revisiting language models as knowledge bases. *arXiv preprint arXiv:2106.09231*, 2021.

[4] Lihu Chen, Gaël Varoquaux, and Fabian M. Suchanek. A Lightweight Neural Model for Biomedical Entity Linking. In *AAAI*, 2021.

[5] Lihu Chen, Gaël Varoquaux, and Fabian M. Suchanek. GLADIS: A General and Large Acronym Disambiguation Benchmark. In *EACL*, 2023.

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[7] Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. Lm vs lm: Detecting factual errors via cross examination, 2023.

[8] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[9] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023.

[10] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality, 2023.

[11] Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*, 2022.

[12] Chadi Helwe, Chloé Clavel, and Fabian M. Suchanek. Reasoning with Transformer-based Models: Deep Learning, but Shallow Reasoning. In *AKBC*, 2021.

[13] Chadi Helwe, Chloé Clavel, and Fabian M. Suchanek. LogiTorch: A Pytorch-based library for logical reasoning on natural language. In *EMNLP demo track*, 2022.

[14] Chadi Helwe, Simon Coumes, Chloé Clavel, and Fabian M. Suchanek. TINA: Textual Inference with Negation Augmentation. In *EMNLP Find.*, 2022.

[15] Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. Chatdb: Augmenting llms with databases as their symbolic memory, 2023.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[17] 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*, 55(12):1–38, 2023.

[18] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[19] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2022.

[20] Z. Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. Deductive verification of chain-of-thought reasoning. 2023.

[21] Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Exposing attention glitches with flip-flop language modeling, 2023.

[22] Qi Liu, Dani Yogatama, and Phil Blunsom. Relational memory-augmented language models. *TACL*, 2022.

[23] Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. Large language model is not a good few-shot information extractor, but a good reranker for hard samples!, 2023.

[24] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. Augmented language models: a survey, 2023.

[25] OpenAI. Chatgpt plugins. `https://openai.com/blog/chatgpt-plugins`, 2023.

[26] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

[27] Pierre-Henri Paris, Syrine El Aoud, and Fabian M. Suchanek. The Vagueness of Vagueness in Noun Phrases. In *AKBC*, 2021.

[28] Pierre-Henri Paris and Fabian M. Suchanek. Non-named entities - the silent majority. In *ESWC short paper track*, 2021.

[29] Simon Razniewski, Andrew Yates, Nora Kassner, and Gerhard Weikum. Language models as or for knowledge bases. *arXiv preprint arXiv:2110.04888*, 2021.

[30] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[31] Fabian M. Suchanek, Georgiana Ifrim, and Gerhard Weikum. Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents . In *SIGKDD short paper track*, 2006.

[32] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago - A Core of Semantic Knowledge . In *WWW*, 2007.

[33] Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. SOFIE: A Self-Organizing Framework for Information Extraction . In *WWW*, 2009.

[34] Fabian M. Suchanek and Gaël Varoquaux. On Language Models and Symbolic Representations. In *The Conversation*, 2022.

[35] Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. Can language models be biomedical knowledge bases? *arXiv preprint arXiv:2109.07154*, 2021.

[36] Aliaksandr Talaika, Joanna Asia Biega, Antoine Amarilli, and Fabian M. Suchanek. IBEX: Harvesting Entities from the Web Using Unique Identifiers . In *WebDB workshop*, 2015.

[37] Thomas Pellissier Tanon, Marcos Dias de Assunção, Eddy Caron, and Fabian M. Suchanek. Demoing Platypus – A Multilingual Question Answering Platform for Wikidata . In *ESWC demo track*, 2018.

[38] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[39] Denny Vrandecic. The future of knowledge graphs in a world of large language models. `https://www.youtube.com/watch?v=WqYBx2gB6vA`, 2023.

[40] Gerhard Weikum, Luna Dong, Simon Razniewski, and Fabian M. Suchanek. *Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases*. Foundations and Trends in Databases, 2021.

[41] Yixuan Weng, Minjun Zhu, Shizhu He, Kang Liu, and Jun Zhao. Large language models are reasoners with self-verification. *arXiv preprint arXiv:2212.09561*, 2022.

[42] Steven Wolfram. Wolfram alpha as the way to bring computational knowledge superpowers to chatgpt. `https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/`, 2023.

[43] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering, 2022.

[44] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337*, 2022.

[45] Chaoqi Zhen, Yanlei Shang, Xiangyu Liu, Yifei Li, Yong Chen, and Dell Zhang. A survey on knowledge-enhanced pre-trained language models. *arXiv preprint arXiv:2212.13428*, 2022.

[46] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv preprint arXiv:2305.13168*, 2023.