Knowledge Bases and Language Models: Complementing Forces

Fabian Suchanek[1,2][0000−0001−7189−2796] and Anh Tuan Luu[2][0000−0001−6062−207X]

[1] Institut Polytechnique de Paris, France
[2] Nanyang Technological University, Singapore

Abstract. Large language models (LLMs), as a particular instance of generative artificial intelligence, have revolutionized natural language processing. In this invited paper, we argue that LLMs are complementary to structured data repositories such as databases or knowledge bases, which use symbolic knowledge representations. Hence, the two ways of knowledge representation will likely continue to co-exist, at least in the near future. We discuss ways that have been explored to make the two approaches work together, and point out opportunities and challenges for their symbiosis.

Keywords: Large Language Models · Knowledge Bases · Databases.

1 Large Language Models

Large Language Models (LLMs) have revolutionized the field of natural language processing in 2023, with the rise of models such as GPT [44], LLaMA [69], and PaLM [11]. As instances of what is called "generative Artificial Intelligence (AI)", LLMs can chat like humans, answer questions, summarize or translate text, write program code, and appear to be able to do just about anything a human can do on text.

LLMs thus have the potential to deeply change our everyday life: they might automatize the work of human professionals in areas where we thought human intelligence was indispensable (such as the work of journalists, legal consultants, authors, programmers, teachers, or researchers [15]). They will pose new societal threats, as they can generate fake news, fake posts, and fake conversations at an unprecedented rate. They might one day pursue their own agenda, threaten users (as it has already happened [39]), or manipulate humans to grant them favors (as hypothesized by Eliezer Yudkowsky in his AI Box thought experiment [76][3]). Already now, they pose legal and ethical challenges in areas such as the copyright of the generated text [4], liability for generated statements [78], explainability of decisions based on such text, the personal data stored in language models (or submitted in prompts), the intellectual property of data used for training, and the remuneration of those who created the training data [20].

[3] https://arminbagrat.com/agi-box/

At the same time, LLMs democratize access to the digital world: novices can now write programs, formulate SQL queries, understand the gist of scientific articles, write, translate, or summarize text, and find answers to just about any question they may have. LLMs will certainly shortly pop up in mobile phones, office software, email clients, and digital assistants, and greatly simplify our life there. In the future, generative AI systems might be granted access to computational tools (so that they can execute code), gain multi-modal capabilities (so that they can see, listen, and speak), become embodied (so that they can physically act), and become ingrained in our life in ways that would be naive to anticipate now [6].

These considerations raise the question whether LLMs will be, or possibly even are, intelligent or even conscious [6]. From a materialistic point of view, one can imagine that a system that is sufficiently complex can indeed show similar emergent properties as the brain, including intelligence and consciousness [58]. However, this is a debate that will have to be pursued not just with computer scientists, but also with philosophers, neuro-scientists, and psychologists. Quite possibly, we might have to come up with new notions of consciousness and intelligence that are suitable for AI. In analogy to Neil deGrasse Tyson's adage, the universe is under no obligation to conform to the words that we humans have invented. It is rather up to us to make our words fit an ever-changing reality — potentially by inventing new ones. For example, when the electric scooter was invented, people did not insist on categorizing it as a kick scooter or as a motorbike. Rather, they invented a new word to describe it, and introduced the legislation to regulate it. We might have to do the same in the case of AI.

2 Structured Data

Despite the successes of LLMs, most knowledge in organizations or companies is stored not in LLMs, but in structured data repositories that use symbolic knowledge representations – such as databases, knowledge bases, XML files, or JSON datasets. Now suppose that we are working at a company that builds an airplane, and that we have a database of all the parts of that airplane. Should we train an LLM to learn the content of that database, and replace the database by the LLM? At the current state of affairs, that would be a terrible idea, for several reasons:

LLMs are probabilistic by nature. While this is convenient for language [77], it is inadmissible for a definite list of items, such as airplane parts. A screw is either part of the airplane or it is not, it should not be there with a certain probability. Even if the LLM tells us that each part is present with a probability of 99%, this would still mean that thousands of parts could be missing, as modern airplanes consist of millions of parts.

LLMs cannot memorize well. Even if we train the models on selected corpora only [26, 13], there is no guarantee that the model will remember what it was trained on without forgetting some facts or inventing others [48, 63, 5, 64]. This is to be expected: language models are machine learning models, i.e., they are designed to generalize, not to memorize.

LLMs may give different answers depending on how we ask. When the same question is asked in different ways or different languages, the answers can be different [71]. This is the reason for the rise of the new discipline called prompt engineering [75].

LLMs may mislead. LLMs will always wrap their wrong answers into a deceptively convincing language: They know how to talk even when they don't know what to say.

Weaknesses accumulate when the query needs reasoning. Many queries on structured data require joins (such as finding the airplane parts that are part of other airplane parts), aggregation (computing the total weight of the airplane), or even reasoning (proving that the total weight of the airplane plus its cargo can be carried by the lift). In such cases, the above weaknesses may accumulate [19].

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 what they contain and how they will reply to queries [71]. They remain black boxes that act at their own discretion.

LLMs cannot be fixed or updated in a straightforward way. LLMs are great only when they work. However, when the model gives an incorrect answer, or when a data point changes, we cannot easily "fix" the model. There are various ways in which the model can be retrained, supplemented by an external memory, or undergo modification of its parameters – but all of these are more complex, more costly, and less reliable than issuing an `UPDATE` statement on a database.

LLMs are costly to train. Databases are more sustainable [1, 55], and faster than LLMs when it comes to retrieving simple 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 [71].

LLMs can be tricked. Through clever prompt engineering (called jailbreaking [32]), LLMs can be made to reveal internal mechanisms, share private data, produce offensive speech, or perform unintended workloads. LLMs thus pose a security risk[4][5][6].

While some of these weaknesses will go away with more (or better) training data, others appear to be here to stay. Thus, at least for now, it seems that structured data repositories have their raison-d'être. Whenever we want to store crisp lists of items, such as commercial products, employees, proteins, or indeed airplane parts, structured data repositories are still the way to go. They are efficient to query, easy to update, amenable to auditing, and 100% deterministic in their

[4] https://simonwillison.net/2023/May/2/prompt-injection-explained/
[5] https://www.jailbreakchat.com/
[6] https://owasp.org/www-project-top-10-for-large-language-model-applications/descriptions/

answers. In simple application cases, a JSON document will work. If we want a fixed schema, large-scale efficiency, and ample software support, a database will be the method of choice. If we need a taxonomy, semantic axioms, reasoning capabilities, and interoperability, a knowledge base lends itself.

By way of introduction, a knowledge base (also known as a KB or knowledge graph) is a labeled directed graph, where the nodes are entities (such as airplane parts, people, organizations, or locations), and the edges are relations between these entities (such as which part belongs to which other part, who works where, or which organization is located in which place) [61]. Similar entities are grouped together into classes (all airplane engineers are in a class "airplane engineers"), and these classes form a taxonomy, where more special classes (such as "airplane engineers") are included in more general classes (such as "people"). In addition, a KB can specify axioms that say, e.g., that every person must have a birth date, that the weight of an airplane part must be a positive numerical value, that the part-of relation is transitive, that people and airplane parts are disjoint, or that a person cannot have more than two parents. KBs can refer to entities of other KBs, and thus reuse what has been defined elsewhere. There are today thousands of publicly available KBs, and these are interlinked in the Semantic Web. KBs can be queried in a formal language called SPARQL, and the responses are efficient, deterministic, and easy to update.

3 LLMs and Structured Data: Complementary Forces

LLMs and structured data repositories store information in fundamentally different ways: in the latter, the information is stored in a symbolic, crisp, accessible way. In a (deep-learning based) LLM, the information is stored in a probabilistic, distributed, opaque, and sub-symbolic way. Each approach has its advantages. Structured data repositories provide a cheaper and more reliable performance than LLMs for simple factoid data and queries. However, they are way less accessible to the user than LLMs, because they require complex query languages. LLMs, in contrast, provide an unparalleled ease of interaction – in a way that is literally very natural (natural language). Furthermore, LLMs store a wealth of informal information that would be cumbersome or outright impossible to store in structured data: commonsense knowledge about objects of everyday life [43], probabilistic knowledge about how things usually are [42], knowledge about processes or hypotheses [56], and the ability to perform casual reasoning.

It thus appears that language models and structured data repositories are complementary: language models excel at general knowledge, and at analyzing and generating natural language text. Structured data, in contrast, is the tool of choice when it comes to storing exact items, and reasoning on them. Again, an analogy with the human brain can be instructive: The human brain is a fantastically powerful and versatile tool. And yet, for some intellectual tasks, humans resort to "external tools", such as paper and pen, or a written list of items. For example, we do not learn the phone numbers of all our friends by heart. We put them in an address book. We do not conduct a proof of a theorem entirely in our heads. We write it down. In the same way, a LLM might make use of external tools when it comes to storing data or to formal reasoning. These external tools can be structured data repositories such as databases and knowledge bases, combined with other symbolic tools such as automated reasoners.

4 Combining LLMs and Structured Data

The question is now how a LLM can interact with symbolic knowledge, so that a user can get the ease of interaction of a LLM combined with the factual accuracy of structured data. There are indeed numerous ways in which LLMs have been combined with symbolic knowledge[7]. One way is to fuse the structured data into the model itself [80, 41]. This, however, requires intimate access to the model architecture, while most LLMs are black boxes. Therefore, most approaches resort to natural language as the vehicle to teach language models.

Among these, some approaches [80, 41] try to instill the knowledge at training or fine-tuning time. However, as we have discussed above, there is no guarantee that the model retains what it has been trained on. Other approaches [12, 14] cross-examine the language model to force it to rethink its answers. Again, there is no guarantee that the reply is correct.

Then there is the philosophy that LLMs should not store factual information at all [71]. There is no need for a LLM to learn "by heart" the coordinates of every city on the planet (as GPT currently does), or the weights of all known molecules. Better outsource that knowledge to a knowledge base. The rationale is that if we remove the factoid information from the LLM, this might considerably reduce the size of the model, while allowing it at the same time to concentrate on what it does best: dealing with language. Indeed, there are tasks where a smaller LLM performs better than a large one [72]. This philosophy would call for LLMs to be trained on carefully selected, but much smaller corpora – although still large enough for the LLM to gain general knowledge about the world.

Among the approaches that follow this philosophy, one of the most intuitive ones is to find the relevant information, and add it as a hidden prompt. This is what LangChain proposes[8], and it appears to be what the Bing AI does[9]. This approach can deal with both textual information and structured data [31]. Other approaches equip language models with the ability to use tools – among others, knowledge graphs. This can be achieved via plugins [40], via Augmented Language Models [38], or via an LLM-SQL bridge [25].

5 Challenges in Combining LLMs and Structured Data

The crux of all approaches that aim to combine LLMs with structured data is that they have to bridge the gap between natural language and symbolic

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

knowledge representations [41]. In some cases this is trivial: If a user asks for the coordinates of the city of Paris, a simple database lookup will do the job. However, most cases are not of that easy form. Let us briefly present the main challenges in bridging the gap between a natural language text and a symbolic knowledge representation of its meaning.

For a start, many named entities do not consist of a single capitalized word (such as "Paris"). Then, it is difficult to distinguish the named entities from the surrounding text. This applies in particular to movie titles ("Have you seen the life of Brian yesterday?"), medical terms ("His Diabetes Mellitus, Type 2 was treated with insulin"), and terms that do not (yet) appear in a dictionary ("The Institut polytechnique de Paris was founded in 2019"). The problem of identifying the items of interest in natural language text is known as **Named Entity Recognition** (NER).

A user who asks for Paris is most likely interested in the capital of France. However, when the user is interested in the books of an author named John Smith, then there are literally hundreds of authors of that name. The same is true if the user asks for employees of a certain name, or proteins with a certain name component. Depending on which meaning we choose, the answer we give from the knowledge base will be different. Thus, we have to find the entity that the user is referring to. This is known as the problem of **Disambiguation**.

The next step is the understanding of the sentence itself: is the user interested in the date that the movie was released, or the date that the movie shows in a local cinema? This is the problem of **Relation Extraction**. Finally, determining the user intent and formulating a query on the database falls in the domain of **Question Answering** [27]. Again, this is usually not as trivial as looking up the coordinates of Paris. Rather, we may have to formulate queries that aggregate, join, or negate ("Which proportion of people in France do not live in its 10 biggest cities?").

In some cases, we might have to go further: we might have to build a symbolic knowledge representation of what is said in an entire text – either to understand the context of the question, or to feed the knowledge contained in the text into the knowledge base. The process of distilling symbolic knowledge from natural language text is known as **Information Extraction** (IE) [73]. It faces formidable challenges beyond those already mentioned: First, not all information in text can be conveniently represented in current symbolic knowledge representations. Non-named entities [43], subjective information [42], and complex information about sequential processes, causality, or hypothetical statements [56], for example, bring us to the edge of what is currently possible. Second, the extracted information has to be in itself coherent, and then logically consistent with the data that has already been stored [62]. Furthermore, errors accumulate: While it is nowadays trivial to map a simple subject-verb-object sentence to a symbolic representation, the deficiencies of NER, disambiguation, relation extraction, information extraction, and symbolic knowledge representation add up, and it is currently beyond reach to build a symbolic representation of a full text at the push of a button.

Finally, the KBs themselves may contain errors or be outdated. They thus require constant **data curation**.

6 Our Work

As we have seen, there is a gap to be bridged between natural language text and symbolic knowledge representations. While the gap is still large, the domain of information extraction has been around for decades and has made that gap at least significantly smaller than it used to be (see [73, 74] for surveys).

We have also contributed our bit to this endeavor [57]. A first step to bridge the gap between LLMs and symbolic representations is to ensure that the language model can represent the input text at all. This requires the **embedding** of the words of the text. We have shown that words that currently have no embedding (so-called *out-of-vocabulary words*) can be embedded efficiently by making their representations similar to the representations of similar words [8]. This approach is called LOVE (for Learning Out-of Vocabulary Embeddings). The next step is to **disambiguate** these words, i.e., to map them to the entities in a knowledge base. We have shown that this can be done with a relatively light-weight model [7], even in the case of non-trivial medical terms. We have also proposed a large benchmark for the disambiguation of acronyms (GLADIS, the General Large Acronym Disambiguation benchmark [9]).

To build knowledge graphs from natural language texts, we need the **combination of different IE subtasks** such as NER, relation extraction, and co-reference resolution. It is well-known that these tasks are correlated and beneficial to each other. However, most of the current studies about knowledge graph construction tend to treat each subtask as a separate task or to apply a sequential pipeline approach, which can lead to cascading errors and obfuscation of the inherent relationship between tasks. To overcome this limitation, we have propose UGFIE [82], a dynamic, graph-based general framework for coupling multiple IE tasks through shared span representations that are refined with context derived from entities, relations, and co-references.

To **deal with a lack of training data** in knowledge graph construction, we have proposed Jointprop [81], a Heterogeneous Graph-based Propagation framework for joint semi-supervised entity and relation extraction. This framework captures the global structured information between individual tasks and exploits interactions within unlabeled data. Specifically, we construct a unified span-based heterogeneous graph from entity and relation candidates and propagate class labels based on confidence scores. We then employ a propagation learning scheme to leverage the affinities between labeled and unlabeled samples.

Another avenue of our research has focused on the **reasoning capabilities** of language models. We have first analyzed the reasoning capabilities of existing (smaller) language models, and catalogued their strengths and weaknesses [21]. We have then addressed one of the weaknesses: **textual inference** in the presence of negation. We have proposed a probabilistic definition of textual inference, and we have then shown that this definition can be used to generate negated training examples from positive training examples. This approach (TINA, for Textual Inference with Negation Augmentation) increases the reasoning performance of these models by up to 20% [23]. A software library for reasoning on text with language models, LogiTorch, complements this work [22].

We have also looked into the evaluation of language models, in particular when it comes to the **quality of the stories that LLMs can generate**. We have first collected quality criteria for stories from the scientific literature in the humanities. We have then generated a large corpus of generated stories with human annotations for these criteria. The resulting benchmark (HANNA, for Human-Annotated Narratives) shows that automated metrics are currently not sufficient to measure the quality of stories [10]. We have also looked into the evaluation of machine learning models in general. Nowadays, such models are no longer evaluated just by their prediction accuracy, but also by their transparency. We have developed an approach that can **explain the decision of a machine-learning model** post-hoc. The central idea is to build not one, but several surrogate models that mimic the behavior of the original model in a human-understandable way (which is why the approach is called STACI, for Surrogate Trees for A posteriori Confident Interpretations) [46]. This work is complemented by an approach for explaining regression models, BELLA (for black-box explanations by local linear models [45]).

The flagship of our work is a **knowledge base called YAGO** (Yet Another Great Ontology) [60, 49, 24, 2, 37, 68]. This KB contains 50 million entities of general interest (such as people, organizations, or locations) and hundreds of millions of facts about them. While earlier versions of YAGO were extracted from Wikipedia, newer versions build on Wikidata. The main distinguishing feature of YAGO is its data quality: it provides a clean taxonomy, human-readable entity names, a manually designed schema, and enforced semantic constraints. The work on YAGO has been complemented by work that extracts commercial products from Web pages by making use of UPC/GTIN codes [65]. Other work **constructs a taxonomy** for a given set of entities using the information from the Web [33–35, 70] .

A KB has to be constantly curated. We have developed several approaches to this end. Some of them [67, 66] can automatically **spot errors in a KB based on the edit history**. Others allow the **alignment of entities** in one KB with the entities in another KB (see [79] for a survey). Our main project here is PARIS (Probabilistic Alignment of Relations, Instances, and Schema) [59], which, despite its age of more than 10 years, remains the state of the art in entity alignment even in the face of neural approaches [30].

A large part of our work has focused on the **incompleteness of KBs** (see [47] for a general introduction): We have shown how to compute a lower bound for the number of missing entities in a KB, based purely on the properties of the entities that exist in the KB [53]. We have also developed a method that can estimate, again only from the existing entities in the KB, whether a property (such as *birth-place*) is present in all entities of a given class in the real world [29].

Another method can estimate whether an entity is missing a property in the KB that it has in the real world [16]. One way to fight this incompleteness is to use rules: If we know that people who are married generally live in the place where their spouse lives, we can use that rule to deduce missing places of residence. With AMIE (for Association rule Mining under Incomplete Evidence) [17, 18, 28], we can **mine such rules automatically from the KB**. These rules have recently been combined with neural methods for link prediction [3].

Finally, we have worked on the **querying of KBs**. We have focused on dynamic KBs, which can be accessed only via functions [52, 51]. We have also shown, rather unconventionally, that in some cases it is more efficient to query a KB via Bash commands rather than loading it into a database system [50]. To query several KBs with aggregation queries, we have developed an algorithm that can approximate the answers for such queries [54].

7 Conclusion

Large language models (LLMs), and generative AI in general, offer fascinating opportunities to simplify our lives and to allow for a more equitable access to the digital world. At the same time, they often have to be complemented by structured data in order to ground their output in reality – at least for now. To allow for such a grounding, we have to bridge the gap between the natural language that is the vehicle of communication with LLMs, and the symbolic representations that are used by structured data repositories such as databases and knowledge bases. This is the challenge of information extraction (IE). While this challenge is decades old, newer IE methods make use of LLMs to analyze natural language text, and to help producing symbolic representations from it [83, 36]. In this way, LLMs themselves may provide the key to overcoming their weaknesses.

Acknowledgement. This work was partially funded by the grant ANR-20-CHIA-0012-01 (“NoRDF”).

References

1. Artificial intelligence is booming—so is its carbon footprint. Bloomberg (2023), https://www.bloomberg.com/news/articles/2023-03-09/how-much-energy-do-ai-and-chatgpt-use-no-one-knows-for-sure
2. Biega, J.A., Kuzey, E., Suchanek, F.M.: Inside YAGO2s: A Transparent Information Extraction Architecture . In: WWW demo track (2013)
3. Boschin, A., Jain, N., Keretchashvili, G., Suchanek, F.M.: Combining Embeddings and Rules for Fact Prediction. In: AIB invited paper (2022)
4. Boyle, J., Jenkins, J.: Intellectual Property: Law & The Information Society (2014)
5. Cao, B., Lin, H., Han, X., Sun, L., Yan, L., Liao, M., Xue, T., Xu, J.: Knowledgeable or educated guess? revisiting language models as knowledge bases. arXiv preprint arXiv:2106.09231 (2021)
6. Chalmers, D.J.: Could a large language model be conscious? arXiv preprint arXiv:2303.07103 (2023)

7. Chen, L., Varoquaux, G., Suchanek, F.M.: A Lightweight Neural Model for Biomedical Entity Linking. In: AAAI (2021)
8. Chen, L., Varoquaux, G., Suchanek, F.M.: Imputing Out-of-Vocabulary Embedding with LOVE Makes Language Models Robust with Little Cost. In: ACL (2022)
9. Chen, L., Varoquaux, G., Suchanek, F.M.: GLADIS: A General and Large Acronym Disambiguation Benchmark. In: EACL (2023)
10. Chhun, C., Colombo, P., Suchanek, F.M., Clavel, C.: Of Human Criteria and Automatic Metrics: A Benchmark of the Evaluation of Story Generation (HANNA). In: COLING (2022)
11. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
12. Cohen, R., Hamri, M., Geva, M., Globerson, A.: Lm vs lm: Detecting factual errors via cross examination. arXiv preprint arXiv:2305.13281 (2023)
13. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314 (2023)
14. Du, Y., Li, S., Torralba, A., Tenenbaum, J.B., Mordatch, I.: Improving factuality and reasoning in language models through multiagent debate. arXiv preprint arXiv:2305.14325 (2023)
15. Eloundou, T., Manning, S., Mishkin, P., Rock, D.: Gpts are gpts: An early look at the labor market impact potential of large language models. arXiv preprint arXiv:2303.10130 (2023)
16. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting Completeness in Knowledge Bases . In: WSDM (2017)
17. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: AMIE: Association Rule Mining under Incomplete Evidence in Ontological Knowledge Bases . In: WWW (2013)
18. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast Rule Mining in Ontological Knowledge Bases with AMIE+ . In: VLDBJ (2015)
19. Golovneva, O., Chen, M., Poff, S., Corredor, M., Zettlemoyer, L., Fazel-Zarandi, M., Celikyilmaz, A.: Roscoe: A suite of metrics for scoring step-by-step reasoning. arXiv preprint arXiv:2212.07919 (2022)
20. Hacker, P., Engel, A., Mauer, M.: Regulating chatgpt and other large generative ai models. In: ACM Fairness, Accountability, and Transparency (2023)
21. Helwe, C., Clavel, C., Suchanek, F.M.: Reasoning with Transformer-based Models: Deep Learning, but Shallow Reasoning. In: AKBC (2021)
22. Helwe, C., Clavel, C., Suchanek, F.M.: LogiTorch: A Pytorch-based library for logical reasoning on natural language. In: EMNLP demo track (2022)
23. Helwe, C., Coumes, S., Clavel, C., Suchanek, F.M.: TINA: Textual Inference with Negation Augmentation. In: EMNLP Find. (2022)
24. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia . In: Artificial Intelligence (2013)
25. Hu, C., Fu, J., Du, C., Luo, S., Zhao, J., Zhao, H.: Chatdb: Augmenting llms with databases as their symbolic memory. arXiv preprint arXiv:2306.03901 (2023)
26. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
27. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al.: Natural questions: a bench-

mark for question answering research. Transactions of the Association for Computational Linguistics **7**, 453–466 (2019)
28. Lajus, J., Galárraga, L., Suchanek, F.M.: Fast and Exact Rule Mining with AMIE 3. In: ESWC (2020)
29. Lajus, J., Suchanek, F.M.: Are All People Married? Determining Obligatory Attributes in Knowledge Bases . In: WWW (2018)
30. Leone, M., Huber, S., Arora, A., García-Durán, A., West, R.: A critical reevaluation of neural methods for entity alignment. PVLDB **15**(8) (2022)
31. Liu, Q., Yogatama, D., Blunsom, P.: Relational memory-augmented language models. TACL (2022)
32. Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., Zhao, L., Zhang, T., Liu, Y.: Jailbreaking chatgpt via prompt engineering: An empirical study. arXiv preprint arXiv:2305.13860 (2023)
33. Luu, A.T., Kim, J.j., Ng, S.K.: Taxonomy construction using syntactic contextual evidence. In: EMNLP (2014)
34. Luu, A.T., Kim, J.j., Ng, S.K.: Incorporating trustiness and collective synonym/contrastive evidence into taxonomy construction. In: EMNLP (2015)
35. Luu, A.T., Tay, Y., Hui, S.C., Ng, S.K.: Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In: EMNLP (2016)
36. Ma, Y., Cao, Y., Hong, Y., Sun, A.: Large language model is not a good few-shot information extractor, but a good reranker for hard samples! arXiv preprint arXiv:2303.08559 (2023)
37. Mahdisoltani, F., Biega, J.A., Suchanek, F.M.: YAGO3: A Knowledge Base from Multilingual Wikipedias . In: CIDR (2015)
38. Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pasunuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., et al.: Augmented language models: a survey. arXiv preprint arXiv:2302.07842 (2023)
39. Nicholson, K.: Bing chatbot says it feels 'violated and exposed' after attack. CBC News (2023), https://www.cbc.ca/news/science/bing-chatbot-ai-hack-1.6752490
40. OpenAI: Chatgpt plugins. https://openai.com/blog/chatgpt-plugins (2023)
41. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. arXiv preprint arXiv:2306.08302 (2023)
42. Paris, P.H., Aoud, S.E., Suchanek, F.M.: The Vagueness of Vagueness in Noun Phrases. In: AKBC (2021)
43. Paris, P.H., Suchanek, F.M.: Non-named entities - the silent majority. In: ESWC short paper track (2021)
44. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
45. Radulovic, N., Bifet, A., Suchanek, F.: Bella: Black box model explanations by local linear approximations. arXiv preprint arXiv:2305.11311 (2023)
46. Radulović, N., Bifet, A., Suchanek, F.M.: Confident Interpretations of Black Box Classifiers (STACI). In: IJCNN (2021)
47. Razniewski, S., Suchanek, F.M., Nutt, W.: But What Do We Actually Know? . In: AKBC workshop (2016)
48. Razniewski, S., Yates, A., Kassner, N., Weikum, G.: Language models as or for knowledge bases. arXiv preprint arXiv:2110.04888 (2021)
49. Rebele, T., Suchanek, F.M., Hoffart, J., Biega, J.A., Kuzey, E., Weikum, G.: YAGO: a multilingual knowledge base from Wikipedia, Wordnet, and Geonames . In: ISWC (2016)
50. Rebele, T., Tanon, T.P., Suchanek, F.M.: Bash Datalog: Answering Datalog Queries with Unix Shell Commands. In: ISWC (2018)

51. Romero, J., Preda, N., Amarilli, A., Suchanek, F.M.: Equivalent Rewritings on Path Views with Binding Patterns. In: ESWC (2020)
52. Romero, J., Preda, N., Suchanek, F.M.: Query Rewriting On Path Views Without Integrity Constraints. In: DataMod workshop (2020)
53. Soulet, A., Giacometti, A., Markhoff, B., Suchanek, F.M.: Representativeness of Knowledge Bases with the Generalized Benford's Law. In: ISWC (2018)
54. Soulet, A., Suchanek, F.M.: Anytime Large-Scale Analytics of Linked Open Data . In: ISWC (2019)
55. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp. arXiv preprint arXiv:1906.02243 (2019)
56. Suchanek, F.M.: The Need to Move Beyond Triples . In: Text2Story workshop (2020)
57. Suchanek, F.M.: A Hitchhiker's Guide to Ontology . In: DESIRES invited paper (2021)
58. Suchanek, F.M.: The Atheist Bible, § 4.5.8. https://suchanek.name/texts/atheism (2023)
59. Suchanek, F.M., Abiteboul, S., Senellart, P.: PARIS: Probabilistic Alignment of Relations, Instances, and Schema . In: VLDB (2012)
60. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago - A Core of Semantic Knowledge . In: WWW (2007)
61. Suchanek, F.M., Lajus, J., Boschin, A., Weikum, G.: Knowledge Representation and Rule Mining in Entity-Centric Knowledge Bases . In: RW invited paper (2019)
62. Suchanek, F.M., Sozio, M., Weikum, G.: SOFIE: A Self-Organizing Framework for Information Extraction . In: WWW (2009)
63. Suchanek, F.M., Varoquaux, G.: On Language Models and Symbolic Representations. In: The Conversation (2022)
64. Sung, M., Lee, J., Yi, S., Jeon, M., Kim, S., Kang, J.: Can language models be biomedical knowledge bases? arXiv preprint arXiv:2109.07154 (2021)
65. Talaika, A., Biega, J.A., Amarilli, A., Suchanek, F.M.: IBEX: Harvesting Entities from the Web Using Unique Identifiers . In: WebDB workshop (2015)
66. Tanon, T.P., Bourgaux, C., Suchanek, F.M.: Learning How to Correct a Knowledge Base from the Edit History. In: WWW (2019)
67. Tanon, T.P., Suchanek, F.M.: Neural Knowledge Base Repairs. In: ESWC (2021)
68. Tanon, T.P., Weikum, G., Suchanek, F.M.: YAGO 4: A Reason-able Knowledge Base . In: ESWC (2020)
69. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
70. Tuan, L.A., Hui, S.C., Ng, S.K.: Utilizing temporal information for taxonomy construction. TACL **4** (2016)
71. Vrandecic, D.: The future of knowledge graphs in a world of large language models. https://www.youtube.com/watch?v=WqYBx2gB6vA (2023)
72. Wei, J., Tay, Y., Le, Q.V.: Inverse scaling can become u-shaped. arXiv preprint arXiv:2211.02011 (2022)
73. Weikum, G., Dong, L., Razniewski, S., Suchanek, F.M.: Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. Foundations and Trends in Databases (2021)
74. Weikum, G., Hoffart, J., Suchanek, F.: Knowledge harvesting: achievements and challenges. Computing and Software Science: State of the Art and Perspectives pp. 217–235 (2019)

75. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
76. Yudkowsky, E.: The ai-box experiment. Singularity Institute (2002)
77. Zhang, H., Song, H., Li, S., Zhou, M., Song, D.: A survey of controllable text generation using transformer-based pre-trained language models. arXiv preprint arXiv:2201.05337 (2022)
78. Zhao, S., Wen, J., Tuan, L.A., Zhao, J., Fu, J.: Prompt as triggers for backdoor attack: Examining the vulnerability in language models. arXiv preprint arXiv:2305.01219 (2023)
79. Zhao, X., Zeng, W., Tang, J., Wang, W., Suchanek, F.M.: An Experimental Study of State-of-the-Art Entity Alignment Approaches . In: TKDE (2020)
80. Zhen, C., Shang, Y., Liu, X., Li, Y., Chen, Y., Zhang, D.: A survey on knowledge-enhanced pre-trained language models. arXiv preprint arXiv:2212.13428 (2022)
81. Zheng, Y., Hao, A., Luu, A.T.: Jointprop: Joint semi-supervised learning for entity and relation extraction with heterogeneous graph-based propagation. arXiv preprint arXiv:2305.15872 (2023)
82. Zheng, Y., Tuan, L.A.: A novel, cognitively inspired, unified graph-based multi-task framework for information extraction. Cognitive Computation pp. 1–10 (2023)
83. Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., Zhang, N.: Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. arXiv preprint arXiv:2305.13168 (2023)