Language Models As or For Knowledge Bases

Simon Razniewski[a], Andrew Yates[a,b], Nora Kassner[c] and Gerhard Weikum[a]

[a]Max Planck Institute for Informatics
[b]University of Amsterdam
[b]LMU Munich

Abstract

Pre-trained language models (LMs) have recently gained attention for their potential as an alternative to (or proxy for) explicit knowledge bases (KBs). In this position paper, we examine this hypothesis, identify strengths and limitations of both LMs and KBs, and discuss the complementary nature of the two paradigms. In particular, we offer qualitative arguments that latent LMs are not suitable *as* a substitute for explicit KBs, but could play a major role *for* augmenting and curating KBs.

1. Introduction

The ability of pre-trained contextual language models (LMs) to capture and retrieve factual knowledge has recently stirred discussion as to what extent LMs could be an alternative to, or at least a proxy for, explicit knowledge bases (KBs). LMs, such as BERT [1], GPT [2] or T5 [3] are huge transformer-based neural networks trained in a self-supervised manner on huge text corpora, in order to predict sentence completions or masked-out text parts. In a setting called (masked) prompting or probing [4], these LMs complete a text sequence intended to elicit a relational assertion for a given subject. For example, GPT-3 correctly completes the phrase *"Alan Turing was born in"* with *"London"*, which can be seen as yielding a subject-predicate-object triple ⟨ `Alan Turing, born in, London` ⟩.

Starting from the LAMA probe [5], many works have explored whether this LM-as-KB paradigm could provide an alternative to structured knowledge bases such as Wikidata. Exemplary analyses investigated the inclusion of entity information [6], how to turn LMs into structured KBs [7], and how to incrementally add knowledge without side effects [8]. Other work studied how accuracy relates to the neural network's storage capacity [9] and whether QA performance scales with model size [10]. Another focus area is how LMs-as-KBs can be further augmented with a text retrieval component, to include informative passages (e.g., from Wikipedia) [11, 12, 13].

Although most works make their speculative nature clear (e.g., the title of [5] ends with a question mark), there is an implicit suggestion that LMs could replace structured KBs. On the other hand, NLP-centric works have identified various kinds of inconsistencies in LM outputs [14] or questioned their quantitative performance [15].

This paper discusses the potential of LMs *as* KBs and its "softer" variation of LMs *for* KBs.

Deep Learning for Knowledge Graphs (DL4KG)
srazniew@mpi-inf.mpg.de (S. Razniewski); ayates@mpi-inf.mpg.de (A. Yates); kassner@cis.lmu.de (N. Kassner); weikum@mpi-inf.mpg.de (G. Weikum)

	LM-as-KB	Structured KB
Construction	Self/Unsupervised 👍	Manual or semi-automatic 👎
Schema	Open-ended 👍	Typically fixed 👎
Maintenance		
- adding facts	Difficult, unpredictable side effects 👎	Easy 👍
- correcting/deleting	Difficult 👎	Easy 👍
Knows what it knows	No, assigns probability to everything 👎	Yes, content enumerable 👍
Entity disambiguation	No/limited 👎	Common 👍
Provenance	No 👎	Common 👍

Table 1
Differences of LMs-as-KBs and structured KBs

2. Background

LM-as-KB refers to efforts to use an LM as a source of world knowledge, as proposed by [5]. The knowledge representation is inherently latent, given by the entirety of the neural network's parameter values (in the billions). LMs in general have greatly advanced tasks like text classification, machine translation, information retrieval, and question answering (see, e.g., survey [16]).

KBs, on the other hand, have been steadily advanced since the mid 2000s (with early works like DBpedia, Freebase and Yago) [17]. They represent knowledge in the form of subject-predicate-object (SPO) triples along with qualifiers for non-binary statements. KBs have become key assets in major industry applications [18, 19], including search engines. A major issue for ongoing KB research is quality assurance as the KB is grown and maintained. This includes human-in-the-loop approaches throughout the KB life-cycle [20, 21, 22].

All LM-as-KB examples that follow are based on the GPT-3 daVinci model [2], one of the largest pre-trained LMs as of October 2021.

3. LM-as-KB

3.1. Intrinsic Considerations

The following are principal differences between LMs-as-KBs and structured KBs.

Predictions vs. lookups: While content of structured KBs can be explicitly looked up, LMs have a latent representation and output probabilities at probing time. This has the advantage of not requiring any schema design upfront. However, it implies that it is not possible to enumerate the knowledge stored in an LM, nor can we look up whether a certain fact is contained or not. For predictions with very high confidence scores, this is still viable. However, even top-ranked predictions often have low scores and near-ties. Properly calibrating scores and thresholding is a black art.

Example: GPT-3 does not have tangible knowledge that Alan Turing was born in London; it merely assigns this a high confidence of 83%. Yann LeCun, on the other hand, is given medium confidence in being a citizen of France and Canada (67% and 26%), but he actually has French and USA citizenship, not Canadian. The LM assigns USA a very low score. The Wikidata KB, on the

other hand, only states his French citizenship, not USA. Wikidata is incomplete, but it does not contain any errors.

Statistical correlations vs. explicit knowledge: Errors made by LMs-as-KBs are not random, but exhibit systematic biases [23, 15] due to frequent values and co-occurrences (including indirect co-occurrences captured latently).

Example: When prompting GPT-3 for awards won by Alan Turing, its top-confidence prediction is the Turing Award, and lower-ranked outputs include "Nobel Prize" and "the war" (none of them correct).

Awareness of limits: In KBs, absence of facts is explicit and easy to assert. Wikidata even supports a way of stating non-existence (no-value statements) to impose a local-closed-world view while following a general open-world assumption [24]. LM's latent representations inherently lack awareness of cases where no object exist, and so they easily produce non-zero or even high scores for incorrect assertions.

Example: Alan Turing was homosexual and never married. When prompting GPT-3 with the phrase "Alan Turing married", the top prediction is "Sara Lavington" with score 21%, and for the prompt "Alan Turing and his wife" it is "Sara Turing" (his mother's name). This is a case of LM hallucination [25, 26]. In contrast, Wikidata has an explicit statement ⟨ `Alan Turing, spouse, no value` ⟩ *denoting that he was unmarried.*

Coverage: The scope of KBs is usually limited by the fixed set of predicates specified in the KB schema. These can be hundreds (or even a few thousands) of interesting relations, but will hardly ever be complete. In particular, "non-standard relations", such as `worked with colleague`, `song is about person` (or `event`), `movie based on person's biography`, are missing in all of the major KBs. LMs, on the other hand, latently tap into the full text of Wikipedia, books, news, and more, and are thus able to capture some of these predicates.

Example: Creatively prompting GPT-3 can yield impressive nuggets of knowledge: the input phrases "Turing's colleague" and "Turing worked with" result in outputs like John Womersley, Hugh Alexander, Gordon Welchman (all correct). Likewise, the prompt "The Imitation Game film is about the life of" is completed with the high-confidence output Alan Turing. These anecdotes indicate the great power of LMs to go beyond the current scope and coverage of explicit KBs.

Curatability: In structured KBs, a knowledge curator can correct, add or remove assertions. For LMs, this is an open challenge, as these operations require major (non-monotonic) re-training, or the addition of explicit exceptions, which means reverting to a KB [27, 28].

Example: For the prompt "Alan Turing died in the town of", GPT-3 returns the top prediction "Warrington", which is wrong (he died in Wilmslow). The LM does not provide any hint on how to fix this (e.g., by changing the training corpus or parameters), and a knowledge curator has no way to tackle such errors.

Provenance: LMs have no ability to trace their outputs back to specific source documents (and passages) in the training data. KBs, on the other hand, consider reference sources as an indispensable pillar of scrutable veracity. Provenance is crucial for giving explanations to users, including knowledge engineers who maintain the KB and end-users in downstream applications. Also, without provenance, LMs have no way of pinpointing an incorrect prediction's root cause and correcting the underlying corpus (e.g., removing misleading documents).

Example: Reconsider the previous example of predicting "Warrington" as Turing's death place. The LM itself does not give any cue where this comes from. A diligent and smart Google user could detect a possible origin, namely, news and other reports about a memorial plaque at 2 Warrington Crescent in Maida Vale, London, which is near Turing's birth place. However, the knowledge engineer cannot be certain that this is indeed the culprit.
Correctly predicted facts need explanations, too. For example, the assertion that Turing was engaged with Joan Clarke may appear puzzling given his homosexuality. Pointing to explicit provenance is crucial evidence.

3.2. Pragmatic Considerations

Entity disambiguation: Although LMs are lauded for their ability to disambiguate words based on context, this happens latently, and there is no easy way to explicitly build this into probing procedures [9, 6]. Consequently, LMs mix up facts from distinct entities that share surface forms. Although structured KBs cannot perform disambiguation on their own either, they can correctly separate assertions.

Example: GPT-3 completes "Turing was a famous" with "mathematician", "computer", "code" etc., stemming from very different entities (including the Turing Machine).

Numbers and singletons: LMs are good at latently capturing knowledge about predicates with few possible object values, such as nationality or language-spoken. However, when the object values are rarely occurring values or even singletons (i.e., occurring only with a single subject), the latent representation is bound to produce errors, and explicit KB storage is superior. The same applies to many cases of numeric values, where the value distribution exhibits high entropy.

Example: For the input "The Turing Institute's address in London is", GPT-3 returns "Dilly's Den" or "the street called Dilly's Den" (possibly derived from the famous Piccadilly Circus; the correct value is British Library, 96 Euston Road, London NW1 2DB). Rephrasing the prompt does not lead to success either.

Subjects with zero or many objects: An important case where the brittleness of LM predictions becomes a significant problem is when a subject entity has no object value for a given predicate or has many distinct true values. The zero-value case often leads to the pitfall that the LM must predict some value. In the many-values case, we could go deep in the ranking of the LM output, but this would usually result in a wild mix of valid and spurious objects, and there is no guideline for how deep we should go into the ranking.

Example: To obtain a list of Turing Award winners, we could prompt GPT-3 with the phrase "the Turing Award was won by" and receive various predictions like "Stuart Shieber", "John Hopcroft" and "Andrew Yao" (1 false, 2 correct). There are currently 73 winners, all captured in Wikidata. By probing LMs, we would have to go very deep in the prediction ranking to see all of them, but only in a confusing mix of true and false positives.
As for zero-objects, the prompt for "the first woman on the moon was" returns Sally Ride, Eileen Collins and others. These are astronauts, but unfortunately, none of them ever landed on the moon. The ground-truth for this example is empty.

We summarize the main differences in Table 2.

4. LM-for-KB

Our view of how to harness the great potential of LMs is to leverage them *for KB curation*: maintaining high quality as the KB grows throughout its life-cycle. This is a major pain point in KB practice [20, 21, 22]. For example, when adding new entities, one needs to ensure that they are not duplicates (with slightly different alias names) of existing entities. Likewise, keeping the type system (aka ontology) clean while gradually extending it and ensuring the correctness of new facts are never-ending challenges.

The envisioned role of LMs is to *scrutinize SPO assertions* considered for augmenting the KB. For example, a new fact such as ⟨ `Leonardo da Vinci, has won, Turing Award` ⟩ could be "double-checked" by prompting the LM as to whether it yields high-confidence predictions for this candidate assertion. This is akin to the way knowledge graph embeddings [29] have been considered for KB completion. However, the key difference is that KG embeddings draw from the KG itself, and thus do not provide complementary evidence. LMs, on the other hand, bring in a new and largely independent perspective, by tapping into text corpora (including Wikipedia, but also news, books etc.). If the LM does not yield sufficiently confident support for the candidate fact, it should be refuted.

The converse direction, using LMs to predict assertions and thus generate candidates for new facts, is conceivable too. However, this needs major research to advance prediction accuracy.

5. Conclusion

In this paper we discussed the strengths and limitations of LMs *as* KBs in comparison to structured KBs. We believe that LMs cannot broadly replace KBs as explicit repositories of structured knowledge. While the probabilistic nature of LM-based predictions is suitable for task-specific end-to-end learning, the inherent uncertainty of outputs does not meet the quality standards of KBs. LMs cannot separate facts from correlations, and this entails major impediments for KB maintenance. We advocate, on the other hand, that LMs can be valuable assets *for* KB curation, by providing a "second opinion" on new fact candidates or, in the absence of corroborated evidence, signal that the candidate should be refuted. Other ways of combining the strengths of latent knowledge (LMs) and structured knowledge (KBs) could be promising as well, such as "KB-for-LM" approaches that allow a LM to look up facts from an external memory (e.g., [12, 30, 31, 32]) and thus have the potential to combine the strengths of both approaches.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL, 2019.

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019. OpenAI technical report.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv (2019).

[4] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, arXiv (2021).

[5] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: EMNLP, 2019.

[6] N. Poerner, U. Waltinger, H. Schütze, E-BERT: Efficient-yet-effective entity embeddings for BERT, in: Findings of EMNLP, 2020.

[7] C. Wang, X. Liu, D. Song, Language models are open knowledge graphs, arXiv (2021).

[8] R. Wang, et al., K-adapter: Infusing knowledge into pre-trained models with adapters, arXiv (2021).

[9] B. Heinzerling, K. Inui, Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries, in: EACL, 2021.

[10] A. Roberts, C. Raffel, N. Shazeer, How much knowledge can you pack into the parameters of a language model?, in: EMNLP, 2020.

[11] F. Petroni, P. Lewis, A. Piktus, T. Rocktäschel, Y. Wu, A. H. Miller, S. Riedel, How context affects language models' factual predictions, in: AKBC, 2020.

[12] K. Guu, K. Lee, Z. Tung, P. Pasupat, M.-W. Chang, Realm: Retrieval-augmented language model pre-training, in: ICML, 2020.

[13] P. Lewis, et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in: NeurIPS, 2021.

[14] Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, Y. Goldberg, Measuring and improving consistency in pretrained language models, arXiv (2021).

[15] B. Cao, H. Lin, X. Han, L. Sun, L. Yan, M. Liao, T. Xue, J. Xu, Knowledgeable or educated guess? revisiting language models as knowledge bases, in: ACL, 2021.

[16] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing, Computational intelligence magazine (2018).

[17] S. Razniewski, P. Das, Structured knowledge: Have we made progress? an extrinsic study of KB coverage over 19 years, in: CIKM, 2020.

[18] N. Noy, et al., Industry-scale knowledge graphs: lessons and challenges, CACM (2019).

[19] G. Weikum, L. Dong, S. Razniewski, F. Suchanek, Machine knowledge: Creation and curation of comprehensive knowledge bases, in: Foundations and Trends in Databases, 2021.

[20] J. Taylor, ~~Automated~~ knowledge base construction, AKBC invited talk, 2020. https://youtu.be/JsB4T35We0w?t=12032.

[21] A. Piscopo, E. Simperl, What we talk about when we talk about Wikidata quality: a literature survey, in: Symposium on Open Collaboration, 2019.

[22] K. Shenoy, F. Ilievski, D. Garijo, D. Schwabe, P. Szekely, A study of the quality of Wikidata, arXiv (2021).
[23] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots: Can language models be too big?, in: FAccT, 2021.
[24] H. Arnaout, S. Razniewski, G. Weikum, J. Z. Pan, Negative knowledge for open-world Wikidata, in: Companion Proceedings of the Web Conference, 2021.
[25] A. Rohrbach, et al., Object hallucination in image captioning, in: EMNLP, 2018.
[26] C. Wang, R. Sennrich, On exposure bias, hallucination and domain shift in neural machine translation, in: ACL, 2020.
[27] C. Zhu, A. S. Rawat, M. Zaheer, S. Bhojanapalli, D. Li, F. Yu, S. Kumar, Modifying memories in transformer models, in: arXiv, 2020.
[28] N. D. Cao, W. Aziz, I. Titov, Editing factual knowledge in language models, in: EMNLP, 2021.
[29] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, TKDE (2017).
[30] T. Févry, L. B. Soares, N. FitzGerald, E. Choi, T. Kwiatkowski, Entities as experts: Sparse memory access with entity supervision, in: EMNLP, 2020.
[31] H. Sun, L. B. Soares, P. Verga, W. W. Cohen, Adaptable and interpretable neural memory over symbolic knowledge, in: NAACL, 2021.
[32] N. Kassner, O. Tafjord, H. Schutze, P. Clark, Enriching a model's notion of belief using a persistent memory, in: arXiv, 2021.