Increment #1 SU-15
Review literature and news on integration methods for KGs into LLMs

In 2022, [3] conducted a literary review on Language Models as Knowledge Bases and suggested five essential criteria that Pre-trained Language Models (PLMs) should meet to be considered proficient knowledge bases: access, edit, consistency, reasoning and interpretability.

Access. In PLMs, factual knowledge can not be directly accessed or queried, as it is *implicitly* encoded; on the contrary, knowledge in KBs is easily accessed via manual query instructions. Nevertheless, previous research has shown the efficiency of PLMs in few-shot and zero-shot learning settings, demonstrating that knowledge learned during pre-training is accessible via fine-tuning or prompting [6]. Current research concentrates on how to query PLMs likewise KBs, focusing on specific access patterns for different types of knowledge, e.g., probing via fill-in-the-blank prompts [33] and downstream fine-tuning.

Edit. Editing a specific fact in PLMs is not simple, since facts can not be directly accessed but they are encoded in the weights of the model. Given the pre-trained nature of PLMs, encoded knowledge may become outdated, incorrect [23], biased or toxic [13, 4]; furthermore, PLMs may have learnt sensitive information that should not be memorised [8]. To learn up-to-date, correct and unbiased knowledge, the naive solution would be to re-train the model on an updated corpus. This process is even more expensive, considering the increasing dimensions of (L)LMs. Additionally, several studies have evidenced that scaling PLMs to lager sizes is not the solution for generating factually correct information; contrarily, altering even a single weight may affect several others, leading to catastrophic forgetting [40]. In [7] a set of rules editing methods should conform to is proposed

- *Generality.* Editing methods should be abled of changing facts of any PLM not specifically trained on adaptability, e.g., meta-learning [12];
- *Reliability.* Editing a PLM should only affect the targeted fact, not the unrelated information;
- *Consistency.* Changes should be consistent across semantically equivalent inputs [15, 27].

On the other hand, in triplets KBs can be directly added, modified and deleted.

Consistency. [11] showed that PLMs lack consistency in their answers. As a matter of facts, LLMs may provide different answers to the same underlying factual question, depending on the information seen during the training phase, which might be not up-to-date or conflicting with other information; on the other hand, KGs are built with consistency in mind, and several algorithms to manage conflicts are available. It is important to keep in mind that consistency does not imply correctness: incorrect information can still be consistent in the KB or PLM.

Reasoning. Recent studies have demonstrated that PLMs can handle various types of reasoning when fine-tuned with data designed to highlight their reasoning capabilities. PLMs exhibit strong performance on reasoning tasks framed in natural language, leveraging the knowledge acquired during their pre-training. However, [31] observed that even the most advanced models struggle to execute more than 2 or 3 non-trivial steps of complex reasoning. Furthermore, [45] questions whether it is appropriate to label these processes as "reasoning," suggesting that PLMs may simply be mimicking human thought processes. For more intricate tasks represented in formal structures, such as structured reasoning [19], PLMs perform poorly. In contrast, reasoning within KGs benefits from a clear and explicitly defined reasoning path.

Explainability and Interpretability. PLMs are not explainable nor interpretable, since factual knowledge is hard to identify by simply observing the output. Explainability is determined by the extent to which the output of a deep learning model can be motivated. By explaining the behaviour of a model, it is also possible to correct its errors and improve its trustworthiness. Interpretability focuses on understanding the inner workings and mechanisms of the model. With the advent of the Transformer architecture [39, 9] interpretability was overshadowed by the impressive results this architecture was able to achieve. Still, some model capabilities have not been interpreted yet, and long-term usage of such models is highlighting some problematic behaviours. In KGs, the outputs are easy to read and interpret.

With respect to the aforementioned five aspects, it is clear that PLMs still have a long way to go to in order to serve as knowledge bases. Knowledge-enhanced PLMs outperform traditional PLMs in capturing factual and relational information, as well as language understanding and generation. This gap encouraged the creation of various benchmark datasets and tasks to assess to what extent PLMs can to leverage their encoded knowledge. Experimental analysis have reported that PLMs experience difficulties in accurately recall factual information. [41] found that PLMs struggle to distinguish sense-making natural language sentences from those that do not. [42] demonstrated that BART [21] performance in answering closed-book questions are very limited, since it does not remember training facts with high precision. Of course, the five above-mentioned qualities can be extended to Large Language Models.

1 Enhancing Large Language Models with Knowledge Graphs

The objective of integrating Large Language Model with Knowledge Graphs is to attenuate hallucination occurrences [2] and improve output interpretability and correctness. Across several surveys on enhancing LLMs with KGs each one defines its own taxonomy: [28] uses *KG-enhanced LLM pre-training*, *KG-enhanced LLM inference* and *KG-enhanced LLM interpretability*; [2] *Knowledge-aware inference*, *Knowledge-aware training* and *Knowledge-aware validation*; [49] *Before-training enhancement*, *During-training enhancement*, *Post-training enhancement*. In spite of the names of choice, all the taxonomies are grounded on the very same distinguishing element, the stage at which the external knowledge is integrated.

1.1 Knowledge-aware learning.

Methods integrating external knowledge in the pre-training stage resolve two challenges arising when integrating knowledge into LLMs: heterogeneous embedding space and knowledge noise, that is unrelated knowledge diverting the sentence from its correct meaning.

A possible strategy to solve these issues is to unify text and KG triples into same input format [49]. This enhancement strategy demonstrated improved reasoning abilities without having to increase the number of parameters and the training time. Additionally, KG-enhanced data better describe and model the common sense knowledge [25, 34, 47, 30, 32, 51]. However, this solution requires additional computational resources and time, with the risk of introducing noise, and makes the pre-training more complex. This strategy better suites domains without a sufficient training corpus.

A second strategy is to leverage KGs during the pre-training phase in order to improve the knowledge expression of LLMs [28, 49]. These methods are suitable for handling time-insensitive, complex knowledge-grounded tasks such as understanding common-sense, considering that their design does not include frequent knowledge updates but requires re-training. These methods enable the LLM to learn knowledge directly during training by improving their encoder and training task. Studies propose to incorporate knowledge encoders or external knowledge modules to enable learning form both text and KG concurrently. This strategy proved to improve the performance on downstream tasks, considering that it adaptively incorporate knowledge while learning the parameters. Furthermore, it allows to customise a model to a specific domain or task by introducing special information or modules. On the other hand, this strategy requires to increase the training time and the parameter size, making the model more prone to overfitting, and the resulting model will be limited to the scope of the knowledge in the training data. Some methods *integrate KGs into the training objective* via designing novel knowledge-aware training objectives [51, 32, 44, 46], for instance via exposing more knowledge entities in the pre-training objective. Other approaches *integrate KGs into the LLM inputs* by introducing relevant knowledge sub-graph directly into the inputs of LLMs [35, 25, 34, 50]; however, this method introduces knowledge noise and often focuses on popular entities, ignoring low-frequency entities. Finally, there are methods that perform *KGs instruction-tuning* in order to fine-tune LLMs to better comprehend the structure of KGs and effectively follow user instructions to conduct complex tasks. These methods employ both facts and structure of the KGs to create instruction-tuning datasets, allowing the fine-tuned LLM to extract factual and structural knowledge from KGs, enhancing reasoning abilities. Nevertheless, in order to update the knowledge re-training is required [43].

1.2 Knowledge-aware inference.

At inference stage, the pre-trained model generates a text or a prediction according to a given prompt. These methods tackle LLMs weak-points such as incorrect outputs due to unclear input context, knowledge gaps and training data biases, inability to generalise to unseen scenarios; furthermore, LLMs find multi-step reasoning challenging and are not devised for searching extra information. Therefore, given their convenient structured

knowledge [16], research has recently focused on how to integrate knowledge graphs into the inference stage [2, 28, 52]. Knowledge graph enhanced LLM inference leverages KGs at inference time guaranteeing up-to-date knowledge, differently from the category of methods illustrated above, which might not be able to generalise on unseen information. For this reason, these methods aim at maintaining the knowledge space and the text space separate. As a result these methods are appropriate for handling domain-specific knowledge with frequent updates. However, the fine-tuning requires labelled data and the design of prompts rely on prior knowledge, therefore the possible lack of the latter may represent a limitation.

Retrieval-Augmented Generation In this line of research, the introduction of non-parametric memorisation techniques – e.g., Retrieval-Augmented Generation (RAG) [22], Adaptive Retrieval [26], Graph Retrieval-Augmented Generation [10, 17] – has been proposed. In RAG retrievable knowledge acts as non-parametric memory, which is easily updatable, accommodates extensive long-tail knowledge, and can encode confidential data.

RAG techniques find applications in very diverse domains, from Natural Language Processing to Computer Vision. Tasks of interest to this project are those involving text generation, especially knowledge-aware generation, which translate in the downstream task of Question Answering (QA). QA systems integrated with RAG techniques aim at generating a correct response based on information collected from several sources. A possible strategy is to retrieve the top-k most relevant article snippets with respect to the input query; these bits of information are then forwarded to the LLM, that generates k intermediate responses and finally summarises them into one definitive answer [14, 18]. Some models take this strategy a step further, implementing attention mechanisms to integrate the input question with the retrieved information within the model, that will produce the final answer [5].

However, traditional RAG techniques still face several limitations in real-world scenarios [29]. Their semantic similarity approach is not suitable for capturing the textual interconnections and relational knowledge. Excessively lengthy context in prompts can degrade the performance: [24] noticed that a better performance is achieved when relevant information occurs at the beginning or end of the input context, but worsens when relevant information is in the middle of long contexts. Lastly, vector RAG techniques cannot grasp global information, so they cannot adequately perform Query-Focused Summarisation (QFS) tasks, that are *sense-making* queries requiring a global comprehension of the data [10] – e.g., "*What are the key trends in how scientific discoveries are influenced by interdisciplinary research over the past decade?*" – rather than the retrieval of a specific piece of information. Sense-making tasks require reasoning over "*connections [...] to anticipate their trajectories and act effectively*" [20]. Numerous LLMs – e.g., GPT [1], Qwen2 [48], Llama [37], Gemini [36] – have shown great capabilities in sense-making tasks; nevertheless, when RAG is required, traditional vector RAG approaches cannot manage an entire corpus. Graph Retrieval-Augmented Generation tackles the issue integrating RAG with graph data like Knowledge Graphs (KGs) [16]. Information organised in graphs enables RAG to leverage the interconnections between

multiple texts, and to take advantage of the abstraction and summarisation of textual data.

GraphRAG[1] [10] is a graph-based RAG strategy for enabling *sense-making* over an entire text corpus. The GraphRAG pipeline consists of two main phases: 1) indexing and 2) querying. During *indexing*, the input corpus is split into customisable text units – e.g., paragraphs or sentences – so that an LLM can extract entities, relations and claims. GraphRAG has several default prompts the LLM must be prompted with for the extraction, but they may not fit domain-specific corpora. Therefore, GraphRAG let users to automatically or manually tune the prompts. *Auto tuning* uses input data and LLM interactions to create domain adapted prompts for the generation of the knowledge graph. Thereafter, hierarchical clustering with Leiden technique [38] is performed on the knowledge graph, to detect community structures within the graph; entities in each cluster are distributed across different communities for a more detailed analysis. A *community* is a group made of densely intra-connected nodes, but sparsely inter-connected to other groups in the graph. For each community and its members, a summary is generated in a bottom-up, hierarchical manner. These summaries provide a general outlook on the data – i.e., principal entities, relations and claims in the community – and act as contextual information during the querying stage. At *querying* time, GraphRAG offers two strategies, fit to the information need: global search and local search. Considering the hierarchical nature of the community structure, queries can be answered leveraging the community summaries from different levels. Whether a particular hierarchy level in the community offers the best balance of summary detail and scope for general sense-making questions or not is still an open question.

Global search is suitable for holistic, comprehensive queries that require reasoning over the entire data corpus and community summaries, e.g., "*What are the top five themes in the data?*". Global search implements a *map-reduce* strategy. For a given community level, the summaries are randomly shuffled and divided into chunks of fixed size. At *map* step, intermediate answers are generated in parallel, and the LLM scores in a $[0, 100]$ range how relevant to the query each of them is; answers scoring 0 are excluded. In the *reduce* step, intermediate answers are ranked according to their relevance and iteratively aggregated into a new context, until the token limit is reached. The final context is employed to generate the global answer. The quality of the global search answer is affected by the level of the community hierarchy chosen for getting community reports. Lower hierarchy levels, with their detailed reports, tend to yield more thorough responses, but also increase the time and resources for generating the response due to the quantity of reports.

Local search instead is appropriate for queries reasoning on precise entities occurring in the documents, e.g. "*What are the healing properties of chamomile?*". The local search technique locates a group of entities within the knowledge graph that are semantically linked to the user's input. These entities act as gateways into the knowledge graph, facilitating the retrieval of additional pertinent information, including associated entities, relationships, entity covariates, and community reports.

[1] `https://microsoft.github.io/graphrag/`

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Agrawal, G., Kumarage, T., Alghamdi, Z., Liu, H.: Can knowledge graphs reduce hallucinations in llms?: A survey. arXiv preprint arXiv:2311.07914 (2023)
3. AlKhamissi, B., Li, M., Celikyilmaz, A., Diab, M., Ghazvininejad, M.: A review on language models as knowledge bases (2022), https://arxiv.org/abs/2204.06031
4. Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: Can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. p. 610–623. FAccT '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3442188.3445922, https://doi.org/10.1145/3442188.3445922
5. Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Van Den Driessche, G.B., Lespiau, J.B., Damoc, B., Clark, A., et al.: Improving language models by retrieving from trillions of tokens. In: International conference on machine learning. pp. 2206–2240. PMLR (2022)
6. Brown, T.B.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
7. Cao, N.D., Aziz, W., Titov, I.: Editing factual knowledge in language models (2021), https://arxiv.org/abs/2104.08164
8. Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T.B., Song, D., Erlingsson, Ú., Oprea, A., Raffel, C.: Extracting training data from large language models. CoRR **abs/2012.07805** (2020), https://arxiv.org/abs/2012.07805
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
10. Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Larson, J.: From local to global: A graph rag approach to query-focused summarization. arXiv preprint arXiv:2404.16130 (2024)
11. Elazar, Y., Kassner, N., Ravfogel, S., Ravichander, A., Hovy, E.H., Schütze, H., Goldberg, Y.: Measuring and improving consistency in pretrained language models. CoRR **abs/2102.01017** (2021), https://arxiv.org/abs/2102.01017
12. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
13. Gehman, S., Gururangan, S., Sap, M., Choi, Y., Smith, N.A.: Realtoxicityprompts: Evaluating neural toxic degeneration in language models. arXiv preprint arXiv:2009.11462 (2020)
14. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.: Retrieval augmented language model pre-training. In: International conference on machine learning. pp. 3929–3938. PMLR (2020)
15. Hase, P., Diab, M., Celikyilmaz, A., Li, X., Kozareva, Z., Stoyanov, V., Bansal, M., Iyer, S.: Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs (2021), https://arxiv.org/abs/2111.13654
16. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Kirrane, S., Labra Gayo, J.E., Navigli, R., Neumaier, S., Ngonga Ngomo, A.C., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J.F., Staab, S., Zimmermann, A.: Knowledge Graphs. No. 22 in Synthesis Lectures on Data, Semantics, and Knowledge, Springer (2021). https://doi.org/10.2200/S01125ED1V01Y202109DSK022, https://kgbook.org/

17. Hu, Y., Lei, Z., Zhang, Z., Pan, B., Ling, C., Zhao, L.: Grag: Graph retrieval-augmented generation (2024), https://arxiv.org/abs/2405.16506
18. Izacard, G., Grave, E.: Leveraging passage retrieval with generative models for open domain question answering. arXiv preprint arXiv:2007.01282 (2020)
19. Kassner, N., Schütze, H.: Negated LAMA: birds cannot fly. CoRR **abs/1911.03343** (2019), http://arxiv.org/abs/1911.03343
20. Klein, G., Moon, B., Hoffman, R.R.: Making sense of sensemaking 1: Alternative perspectives. IEEE intelligent systems **21**(4), 70–73 (2006)
21. Lewis, M.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)
22. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **33**, 9459–9474 (2020)
23. Lin, X.V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., et al.: Few-shot learning with multilingual language models. arXiv preprint arXiv:2112.10668 (2021)
24. Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics **12**, 157–173 (2024)
25. Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P.: K-bert: Enabling language representation with knowledge graph. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 2901–2908 (2020)
26. Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., Hajishirzi, H.: When not to trust language models: Investigating effectiveness of parametric and non-parametric memories (2023), https://arxiv.org/abs/2212.10511
27. Mitchell, E., Lin, C., Bosselut, A., Finn, C., Manning, C.D.: Fast model editing at scale (2022), https://arxiv.org/abs/2110.11309
28. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering (2024)
29. Peng, B., Zhu, Y., Liu, Y., Bo, X., Shi, H., Hong, C., Zhang, Y., Tang, S.: Graph retrieval-augmented generation: A survey (2024), https://arxiv.org/abs/2408.08921
30. Poerner, N., Waltinger, U., Schütze, H.: E-bert: Efficient-yet-effective entity embeddings for bert. arXiv preprint arXiv:1911.03681 (2019)
31. Polu, S., Han, J.M., Zheng, K., Baksys, M., Babuschkin, I., Sutskever, I.: Formal mathematics statement curriculum learning. arXiv preprint arXiv:2202.01344 (2022)
32. Rosset, C., Xiong, C., Phan, M., Song, X., Bennett, P., Tiwary, S.: Knowledge-aware language model pretraining. arXiv preprint arXiv:2007.00655 (2020)
33. Shin, T., Razeghi, Y., au2, R.L.L.I., Wallace, E., Singh, S.: Autoprompt: Eliciting knowledge from language models with automatically generated prompts (2020), https://arxiv.org/abs/2010.15980
34. Sun, T., Shao, Y., Qiu, X., Guo, Q., Hu, Y., Huang, X., Zhang, Z.: Colake: Contextualized language and knowledge embedding. arXiv preprint arXiv:2010.00309 (2020)
35. Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., Liu, J., Chen, X., Zhao, Y., Lu, Y., et al.: Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. arXiv preprint arXiv:2107.02137 (2021)
36. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)

37. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
38. Traag, V.A., Waltman, L., Van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific reports **9**(1), 1–12 (2019)
39. Vaswani, A.: Attention is all you need. Advances in Neural Information Processing Systems (2017)
40. Wallat, J., Singh, J., Anand, A.: Bertnesia: Investigating the capture and forgetting of knowledge in bert (2021), https://arxiv.org/abs/2106.02902
41. Wang, C., Liang, S., Zhang, Y., Li, X., Gao, T.: Does it make sense? and why? a pilot study for sense making and explanation. arXiv preprint arXiv:1906.00363 (2019)
42. Wang, C., Liu, P., Zhang, Y.: Can generative pre-trained language models serve as knowledge bases for closed-book qa? arXiv preprint arXiv:2106.01561 (2021)
43. Wang, J., Huang, W., Shi, Q., Wang, H., Qiu, M., Li, X., Gao, M.: Knowledge prompting in pre-trained language model for natural language understanding. arXiv preprint arXiv:2210.08536 (2022)
44. Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., Tang, J.: Kepler: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics **9**, 176–194 (2021)
45. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)
46. Xiong, W., Du, J., Wang, W.Y., Stoyanov, V.: Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. arXiv preprint arXiv:1912.09637 (2019)
47. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: Luke: Deep contextualized entity representations with entity-aware self-attention. arXiv preprint arXiv:2010.01057 (2020)
48. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al.: Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 (2024)
49. Yang, L., Chen, H., Li, Z., Ding, X., Wu, X.: Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. IEEE Transactions on Knowledge and Data Engineering (2024)
50. Zhang, T., Wang, C., Hu, N., Qiu, M., Tang, C., He, X., Huang, J.: Dkplm: decomposable knowledge-enhanced pre-trained language model for natural language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 11703–11711 (2022)
51. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: Ernie: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129 (2019)
52. Zhao, P., Zhang, H., Yu, Q., Wang, Z., Geng, Y., Fu, F., Yang, L., Zhang, W., Cui, B.: Retrieval-augmented generation for ai-generated content: A survey. arXiv preprint arXiv:2402.19473 (2024)