Automated Knowledge Graph Completion for Natural Language Understanding: Known Paths and Future Directions.*

Giovanni Buzzega[1,†], Veronica Guidetti[1,†], Federica Mandreoli[1,†], Luca Mariotti[1,†], Andrea Belli[2,†] and Paolo Lombardi[2,†]

[1] *Dipartimento di Scienze Fisiche, Informatiche e Matematiche, Univ. Modena e Reggio Emilia, Modena, Italy*
[2] *Expert.ai, Modena, Italy*

Abstract

Knowledge Graphs (KGs) are large collections of structured data that can model real world knowledge and are important assets for the companies that employ them. KGs are usually constructed iteratively and often show a sparse structure. Also, as knowledge evolves, KGs must be updated and completed. Many automatic methods for KG Completion (KGC) have been proposed in the literature to reduce the costs associated with manual maintenance.

Motivated by an industrial case study aiming to enrich a KG specifically designed for Natural Language Understanding tasks, this paper presents an overview of classical and modern deep learning completion methods. In particular, we delve into Large Language Models (LLMs), which are the most promising deep learning architectures. We show that their applications to KGC are affected by several shortcomings, namely they neglect the structure of KG and treat KGC as a classification problem. Such limitations, together with the brittleness of the LLMs themselves, stress the need to create KGC solutions at the interface between symbolic and neural approaches and lead to the way ahead for future research in intelligible corpus-based KGC.

Keywords

Knowledge Graphs, Natural Language Understanding, Large Language Models, Knowledge Graph Completion

1. Introduction

A Knowledge Graph (KG) conveys real-world knowledge in a graph-structured way where nodes represent entities of interest and edges represent potentially different relations between entities [1]. Since Google announced its Knowledge Graph (KG) in 2012, KGs have quickly become

SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy

[*] This paper is partially funded by Region Emilia-Romagna through the LR 14 year 2021 Project "IbridAI - Hybrid approaches to Natural Language Understanding"

[†] These authors contributed equally.

giovanni.buzzega@unimore.it (G. Buzzega); veronica.guidetti@unimore.it (V. Guidetti); federica.mandreoli@unimore.it (F. Mandreoli); luca.mariotti@unimore.it (L. Mariotti); abelli@expert.ai (A. Belli); plombardi@expert.ai (P. Lombardi)

0000-0002-4117-5312 (G. Buzzega); 0000-0003-2233-0097 (V. Guidetti); 0000-0002-8043-8787 (F. Mandreoli); 0009-0004-7709-5404 (L. Mariotti)

standard solutions in knowledge representation and have played an increasingly important role in many knowledge-aware tasks.

Many companies have introduced the use of KGs into their business processes, improving the performance of their products, e.g., recommendation systems have improved data representation and explainability of recommendations, question-answering systems have increased efficiency and started to answer multi-hop questions, and Information Retrieval has improved the accuracy of results and achieved greater search efficiency [2].

A notable example of enterprise KG is Sensigrafo (from the Italian words sensi = "senses, meanings" and grafo = "graph") developed by the Italian software house Expert.AI[1]. The primary use of Sensigrafo is Natural Language Undestanding (NLU), a branch of AI that uses computer software to understand language as text or speech input. Specifically, Sensigrafo is not meant to be an encyclopedic reference tool for end-users like most KGs [3, 4] but a core module engineered to disambiguate text that is to assign it semantic meaning. To this end, Sensigrafo gives a machine-oriented representation of a language's lexicon where every item is provided with a set of attributes making explicit, in a machine-readable format, all grammatical, syntactical, and semantic characteristics belonging to words and concepts.

As real-world knowledge changes and grows, KGs must be kept updated. One possible solution to KG enrichment is adding novel entities and relations manually. This approach requires expensive human efforts, but, at the same time, it allows adding information to the Knowledge Base with fine-grained control. This is a desirable feature when considering KGs built for NLU tasks since the disambiguation process relies on the principles used to build the KG. Thus, when the KG expansion process is performed, preserving a certain degree of sparsity is imperative. Manual expansion is indeed the main current approach adopted for Sensigrafo, which is constantly updated by several computational linguists.

On the other hand, the literature presents many automatic approaches for KG enrichment, usually referenced as KG Completion (KGC) [5]. Most of these methods adopt machine learning solutions that can be effective when the relations are characterized by sharp logical properties such as symmetry or transitivity and can greatly speed up and reduce the costs of the enrichment process. Still, most of these approaches rely only on the internal structure of the KG, making them unsuitable for sparse KGs and unable to inject new knowledge from external sources.

In recent years, Large Language Models (LLMs), namely gigantic deep learning architectures specifically tailored to manipulate natural language, are showing astounding results in NLU tasks. For this reason, more studies are proposing KG maintenance based on the extraction of knowledge implicitly contained in LLMs after pre-training. LLMs are trained in a self-supervised way on vast amounts of text from the most disparate sources. Therefore, unlike standard KG schemas, their syntactic and semantic content is far from being systematic, organized, and robust, making LLM-based solutions for KGC still unreliable.

KGC methods based on LLMs are ineffective partly due to how such techniques use them, i.e., as oracles to be queried. We believe that to overcome this limitation, it is necessary to use a hybrid approach. In fact, our work stems from "IbridAI - Hybrid approaches to Natural Language Understanding", a project aiming to improve NLU by bridging the gap between symbolic and deep-learning approaches to language representation. One of the challenges of this project is to

[1] https://expert.ai

develop methods to perform corpus-based KGC not only automatically but also *consistently* to retain the principles used to build the reference KG.

In this paper, we move the first steps toward this ambitious objective. To introduce the field of interest, in Section 2, we compare some enterprise and open-access KGs and discuss their applications. We next summarize the main findings concerning state-of-the-art approaches in language manipulation, namely, LLMs. Afterward, Section 4 presents some classical and modern deep-learning methods used to maintain and enrich KGs. Finally, Section 5 discusses the open challenges in KG completion and enrichment, current state-of-the-art limitations of deep learning methods, and possible future research directions.

2. Overview on KGs and their use in NLU

A KG is a set of real-world facts and semantic relations encoded in the form of triples *(subject, predicate, object)*, where the predicate is the relation that links two concepts. Formally we define a KG as $G = (E, R, \mathcal{T})$ where E is the set of all entities, which are nodes of the graph, and R is the set of all relations, and $\mathcal{T} \subseteq E \times R \times E$ is the set of all triples. In each triple (h, r, t) with $h, t \in E$, $r \in R$, h is the head entity, r is the relation link and t is the tail entity [5]. As discussed in the introduction, companies create and use KGs for their business and thus usually keep them private [6].

Table 1 compares the main properties of Sensigrafo with the most important freely accessible KGs [1], that is DBpedia [7], Freebase [8], OpenCyc [9], Wikidata [4], and YAGO [3].

Each node in Sensigrafo is called *syncon* and groups a set of words that can denote identical or similar meanings. The English standard Sensigrafo contains about 440,000 syncons, grouping more than 580,000 words, rules for inflections, and 80+ relation types that yield about 7.1 million links between concepts.

Compared to Sensisgrafo, most of these collaborative KGs have been automatically created and therefore store a very high number of triples. The number of classes highly varies among these KGs, ranging from 736 (DBpedia) up to 300K (Wikidata) and 570K (YAGO). These KGs are cross-domain, and their content depends on the text corpora that have been integrated into the KGs. The number of relation types also varies with KGs, ranging from 106 (YAGO) up to 70k (Freebase). Freebase also contains the largest number of entities, reaching almost fifty million [10]. Generally speaking, the resulting collaborative KGs are rich in relation types and link density because their construction aims to include as much real-world knowledge as possible. In contrast, KGs specifically designed for NLU, such as Sensigrafo, are modeled with an eye for sparsity and only contain most general link types.

In order to use KGs for NLU tasks we must identify named entities in the textual sources and then associate them with entities in the knowledge base. This is carried out by Entity linking which is composed of two subtasks: Named entity recognition (NER), and named entity disambiguation (NED).

As a first step NER classifies the entities mentioned in the text. The identified named entities will later be linked to the knowledge base entities by the NED subtask. The NER task has been historically carried out using rule-based approaches and hand-crafted features [12]. Recently, performances have been improved with the use of statistical approaches such as Conditional

	DBpedia	Freebase	OpenCyc	Wikidata	YAGO	Sensigrafo
No. of triples	411 885 960	3 124 791 156	2 412 520	748 530 833	1 001 461 792	7 185 541
No. of classes	736	53 092	116 822	302 280	569 751	280 428
No. of relations	2819	70 902	18 028	1874	106	80+
No. of entities	4 298 433	49 947 799	41 029	18 697 897	5 130 031	443 932
entities / classes	5840.3	940.8	0.35	61.9	9.0	1.6

Table 1
KGs comparative table: open-source KGs (DBpedia, Freebase, OpenCyc, Wikidata and YAGO) [11] and enterprise KG (Sensigrafo).

Random Fields [13], a biaffine model that scores pairs of start- and end-tokens inside a sentence [14], classification models via various machine learning algorithms [15], and deep learning models [16]. Subsequently NED selects the correct entity from a set of candidates that are identified by NER. Possible approaches include ensemble learning through Support Vector Machines [17], reinforcement learning [18], and a ranking algorithm based on semantic and syntactic metrics from on WordNet [19].

3. Large Language Models in a nut

Since the invention of Transformers [20], LLMs have monopolized state-of-the-art NLP applications [21]. Transformers [20] are architectures merging memory networks with the concept of self-attention. Self-attention is the mechanism that allows assigning weights to tokens based on their relevance so that they can pass more information to the last layers of the network. One notable example of LLM is *BERT* [22] and its numerous successors, which surpassed human performances in some benchmarks [23]. BERT consists of bidirectional transformer encoders that can read the entire text simultaneously and compute attention over previous and following tokens. Thus, BERT outputs can be considered as *contextual embeddings* of each subword token in the input text [24]. Since the release of BERT in 2018, larger LMs have appeared in the literature. Some of them are based on BERT, such as RoBERTa [25] and DeBERTa [23], while others are not, as GPT-3 [26] and PaLM [27]. LLMs are usually trained in a self-supervised fashion, drastically reducing the amount of human effort. Training usually involves *cloze tasks*, i.e., given a partial text, the model should try to fill in the blank.

One of the strengths that led to the success of LLMs is related to their *transfer-learning* ability: once the pre-trained models are published, a little effort is needed to *fine-tune* them for a specific task. Usually, fine-tuning consists of supervised training of at least a part of the LLM parameters and a simple specialized neural network (typically a classifier), which is appended to the LLM and trained for the downstream task. Training LLM parameters creates specialized contextual embeddings that are used as input for the post-pended network. Nevertheless, it was quickly realized that training, and thus finetuning, comes at a cost. A 2019 study [28] estimated training costs of comparatively small models like BERT and GPT-2 to be tens to thousands of USD. In 2020, [29] estimated that developing models with more than one billion parameters could require a budget of millions of USDs. Currently, only large companies can afford the enormous costs associated with training ever-growing deep learning models. Surprisingly, it recently became clear that pre-trained LLMs (PLLMs) show outstanding emergent peculiarities, such as in-context learning (ICL) skills [30], that could make finetuning not only expensive but also superfluous.

ICL belongs to the class of *probing methods*, i.e., methods using the frozen representation of PLLMs to address a specific task. ICL requires a few examples and some instructions (not necessarily) to form a demonstration context. The final predictions are obtained through zero- or few-shot learning techniques [26, 31, 32]. ICL provides an interpretable interface to communicate with LLMs in a *training-free* learning framework and opens up the possibility of testing the consistency of PLLMs, their reaction to different input prompts, and their performance in classical generalization (training/test distribution are the same) and out-of-distribution generalization [30, 33, 34].

The effectiveness of ICL indicates that PLLMs contain structured knowledge. In fact, the pioneeristic work of Petroni et al., [35] showed that PLLMs recall relational and factual knowledge without any fine-tuning. Such knowledge can take the form of a *Knowledge Base* that is constructed using PLLMs as oracle-based entity linkers. Several studies deepen this awareness; see [36] for a recent review. As PLLMs showed ICL capabilities, researchers became more interested in evaluating their intrinsic capabilities and internal structure, neglecting downstream task finetuning [37, 30].

Over the last few years, different benchmarks have been developed to measure the abilities, consistency, and quality of PLLMs. Some of the tasks include Open-Domain Closed-Book Question Answering, Cloze and Completion tasks, Common Sense Reasoning, and In-context Reading Comprehension [27]. Several studies evaluated general-purpose language understanding [38], the consistency of LLMs against paraphrasing [33], the reaction of PLLMs to Chains of Though (CoT) reasoning [34], how much PLLMs rely on semantic priors and their ability to learn input-label mapping [39]. Other works focused on testing whether PLLMs can grasp implicit common sense [40] and their knowledge organization[35, 41, 40].

4. KG maintenance and evolution

The main task in KG maintenance is called KG Completion (KGC). KGC aims to attain new knowledge that was not previously stored in a KG, i.e., adding factual knowledge to a pre-existing KG. In the literature, two main tasks are associated with KGC: Entity Prediction and Link Prediction. Entity prediction identifies the missing entity in a given incomplete triple, while link prediction forecasts the most appropriate relation between two entities. This section sheds light on the difference between classical and modern LLM-based approaches.

4.1. Classical KGC approaches

Classical approaches to KGC can be split into static and dynamic approaches (i.e. based on external sources). Different static KGC techniques use Tensor/matrix factorization, Translation, and Neural Network models [5, 42]. For example, TransE [43] is a translational model that interprets both relations and entities as vectors in the same low-dimensional space; the translation operation defined by a relation vector is used to link entities that were previously not linked. RESCAL [44] models the KG as a 3−dimensional tensor and learns the parameters of a bilinear scoring function for each relation and the entity embeddings. An extensive comparison of classical KGC techniques is presented in [45].

The main limitation of these approaches is that KGs are modeled statically, mostly relying on the structural information already encoded in the KG. This implies that new knowledge is hard to translate into new triples. In fact, it is known that the effectiveness of representation learning based on KG structure decreases dramatically if KGs are sparse [46]. Since the gold standard for KG enrichment comes from human efforts and is based on external (usually textual) natural language resources, the automatization of KG curation should be based on external corpora. Developing dynamic ways to perform KGC based on corpora would allow for a greater degree of interpretability as the source of information can be clearly tracked and it would be possible to perform continuous fact-checking and maintenance of the KG. In particular, we are interested in methods that can jointly model the text in a coherent and reasonable format and complete the pre-existing KG with only relevant facts. Some studies [47, 48] specifically build KGs from text and extract relations using pattern templates. Unfortunately, this approach trades finding the implicit textual relation between identified entities with a rich relation set which is not suited for NLU-oriented KGs. Another approach to KGC and relation extraction considers both text and the existing KG structure by learning latent space representation for both. Similar tasks are known in the literature as *text-enhanced knowledge embedding* [46, 49] and *relation extraction* [50]. These particular approaches require ad hoc architectures, so they cannot easily exploit state-of-the-art models for NLU.

4.2. LLMs applications to KGC

While most KGC methods rely on structural information already present in the KG, some recent approaches have started to leverage LLMs to use third-party data sources [5]. Some approaches use PLLMs fine-tuning to perform KGC. For example, KG-BERT [51] reframes the KGC task as a sequence classification problem. In particular, relation triples are transformed into sentences based on their definitions. A simple classification network is then appended to BERT, and the whole architecture is finetuned to perform a relation prediction task. A similar approach is used in a model named BLP [52], combining BERT and four different relational models.

A recent line of research uses the hidden KG contained in PLLMs to test their consistency and compare their performance at a semantic level. The aim of the works on this subject is usually twofold: providing pipelines for speeding up and automatizing KG completion and understanding LLMs knowledge base organization [53]. In particular, [54] aim to reconstruct and complete an existing KG, namely Word-Net [55], starting from a PLLM. They show that contextualized word embeddings encode high-level concepts and hierarchical relationships, creating a taxonomy. A more general approach presented in [56] aims to create from scratch or complete existing KGs using PLLMs in an unsupervised fashion. Starting from a corpus, they match candidate facts therein with the knowledge in PLLMs via a beam search in their attention weights matrices. Afterward, they map facts (triples) onto a preexisting or open schema KG. This allows performing entity linking inside pre-existing KGs or growing them in case new facts/links are found in the corpus. PALT is another framework to reduce the amount of computational effort in KG completion, [57]. It uses a transfer learning technique to translate KGC into a fill-in-the-blank task (formulated as next-sentence prediction), achieving competitive results in link prediction and triples classification. KGs rarely contain information about the strength of the relationship between entities. Therefore, starting from an existing KG, [58] use PLLMs to weight KG edges, focusing on ConceptNet. They do so by transforming KG relations into natural language sentences and then using the LLM to rate such sentences using a perplexity-based measure. Their results show that the augmented KG improves performances on refining existing word embedding for semantic relatedness. Finally, in [59], the authors found a way to crawl the internal KG of LLMs. Starting from a seed entity, they build a way to grow and expand a KG without knowing a priori what relations are associated with such an entity. To grow the KG and maintain high precision, they decompose the crawling into multiple subtasks that are solved via few-shot ICL tasks, also allowing the model not to know the answer.

Deriving KG from PLLMs can also be done in a corpora-independent fashion. In particular, starting from PLLMs, [60] harvest a knowledge graph. Their method can be extended to any relationship (not only triples). This framework automatically generates diverse prompts by paraphrasing an initial one and searches within an LLM for entity pairs that consistently satisfy diverse sentences.

5. Current limitations, open challenges and future research perspective in corpus-driven KGC

The state-of-the-art overview presented in the previous sections points out that one of the most promising approaches for corpus-driven KGC is represented by LLMs. However, the common motif behind such LLMs applications is the sole interpretation of knowledge contained in PLLMs while neglecting the reference KG. In the following, we underline how current approaches present several limitations when doing corpus-based KGC on KGs with well-defined semantics and a sparse structure and discuss future research directions to cope with these problems. In fact, the enrichment of NLU-oriented KGs is desired to the extent of performance gains. For this reason, we can state that KGC methods need to be parsimonious and consistent. *Parsimonious* in that KGC must retain the sparse structure of the KG and *consistent* in that KGC must comply with the KG's organizing principles, semantics, and structural properties. As we shall see, the limitations relate mainly to the brittleness of the knowledge stored by PLLMs and to current methods that translate KGC into a classification task and rely on PLLMs only.

State-of-the-art PLLMs challenges. Current methods for KGC based on LLMs rely exclusively on the knowledge acquired by LLMs in their pretraining phase. Thus, the predicted facts might be outdated or even wrong. Indeed, studies regarding the consistency, organization, and depth of knowledge learned by LLMs are quite recent and have revealed several weaknesses. Specifically, some recent studies show that PLLMs mainly rely on associations and lack logical reasoning, better distinguish coarse-grained concepts, perform poorly on CoT evaluations, and may suffer from conceptual hallucinations triggered by word co-occurrence [61, 34, 40]. Indeed, PLLMs contain some prior information about entities [35] that cannot always be overwritten and may lead to out-of-context bias. Moreover, they struggle in encoding specific concepts, taxonomical classes, and concepts that are too frequent [54]. Finally, it was found that the ability to learn about a concept in a corpus depends on its frequency and its specificity, justifying the birth of finetuned BERT versions for real-world applications (such as biology, science, and law [62, 63, 64]).

For all the aforementioned reasons, creating an interface that allows LLMs to communicate with symbolic KGs poses some problems. Indeed, while the symbolic approach to language follows rules of abstraction and systematization of knowledge, deep learning models rely primarily on the volume (sometimes more than the quality) of the texts on which they are trained. The lack of guarantees on the effectiveness of the performance of LLMs does not, to date, allow the creation of an automation pipeline in the updating and maintenance of KGs. These problems could be partially solved by investigating how to improve the training of LLMs to consolidate their ICL capabilities. Indeed, open challenges in ICL concern retrieving correct knowledge from PLLMs by making it more robust and insensitive to prompting templates format, the selection of in-context examples, and their ordering [65, 66, 67, 68]. In addition, ICL frameworks should be designed not only to test the consistency of PLLMs against paraphrasing [33] and transitive relations. They should also test whether LMMs satisfy the properties which different relations between KG entities should satisfy, e.g., anti-symmetry of "is a" relations or transitivity in synonyms.

KCG: from classification to representation learning. As mentioned in the previous section, the most used approach for interpreting the embeddings of triplets coming from PLLMs is based on classification. Classification outcomes are always forced to predict a relation between two entities and do not leave much room for uncertainty. This may be problematic in KGC tasks as it was shown that word co-occurrence triggers conceptual hallucinations in LLMs which are biased towards false positive predictions (e.g., prediction of false conceptual properties) [40]. This is undesirable with non-transparent models and may severely compromise disambiguation tasks.

Regardless of the inherent biases of LLMs, modeling KGC as a classification problem prevents the correct handling of KGs where multiple relations connect two entities. This problem would affect both the disambiguation process, which must find the correct triple in a sentence, and the link prediction task, aiming to detect the right relation. An ideal solution for this purpose would be to define a probability measure for relationships between entities. This method should be able to handle multiple outcomes and quantify the uncertainty in the answer so that relationships predicted with low confidence do not enter the KG.

The last problem we would like to discuss arises when KGC based on LLMs acts on KGs equipped with a disambiguator tailored to visit it. In this case, integrating the various components in a unique framework is not only about respecting the structural features of the KG but also about the interaction between the LLM and the disambiguator. Finding a way to teach PLLMs to respect the structure of the KG and the rules used by the disambiguator is an open problem that has not been addressed to date. We argue that a more sensible approach to KGC should rely on representation learning instead of classification, like classical Translational or Tensor/matrix factorization of KGC approaches. This would increase the "maneuvering room" at the interface between KGs and LLMs, also giving greater importance to the pre-existing KG structure and its navigator, if any.

In conclusion, we believe that LLMs should not be used as a Knowledge base to be queried, especially when doing corpus-based KGC. In fact, recently developed LLM-based techniques for KGC mainly rely on artificial prompts paraphrasing standard KG triplets. Misusing these methods on real corpora may give rise to syntactic and/or semantic hallucinations, as one would lose control of LLMs' prior knowledge. Hence, we believe that the task of KGC should use PLLMs only to extract facts from documents, and not from direct interpretations of the latent knowledge of these models.

Pending more robust LLM pretraining techniques, the interface between LLMs and symbolic NLUs should rely on representation learning techniques, such as contrastive learning [69]. These methods would depart from the mere standard classification of triplets via cloze tasks and would allow mimicking and, hopefully, learning the principles used in creating symbolic KGs and disambiguation systems, leading to a consistent, dynamical deep-learning approach to KGC.

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3447772. doi:10.1145/3447772.

[2] C. Peng, F. Xia, M. Naseriparsa, F. Osborne, Knowledge graphs: Opportunities and challenges, 2023.

[3] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, G. Weikum, Yago2: exploring and querying world knowledge in time, space, context, and many languages, Proceedings of the 20th international conference companion on World wide web (2011).

[4] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. URL: https://doi.org/10.1145/2629489. doi:10.1145/2629489.

[5] T. Shen, F. Zhang, J. Cheng, A comprehensive overview of knowledge graph completion, Knowledge-Based Systems 255 (2022) 109597. URL: https://www.sciencedirect.com/science/article/pii/S095070512200805X. doi:https://doi.org/10.1016/j.knosys.2022.109597.

[6] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: Lessons and challenges, Commun. ACM 62 (2019) 36–43. URL: https://doi.org/10.1145/3331166. doi:10.1145/3331166.

[7] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia, Semantic Web 6 (2015) 167–195.

[8] K. D. Bollacker, P. Tufts, T. Pierce, R. Cook, A platform for scalable, collaborative, structured information integration, 2007.

[9] C. Matuszek, J. Cabral, M. Witbrock, J. DeOliveira, An introduction to the syntax and content of cyc., 2006, pp. 44–49.

[10] M. Färber, A. Rettinger, Which knowledge graph is best for me?, ArXiv abs/1809.11099 (2018).

[11] M. Färber, A. Rettinger, Which knowledge graph is best for me?, arXiv preprint arXiv:1809.11099 (2018).

[12] D. E. Appelt, J. R. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, M. Tyson, SRI International FASTUS SystemMUC-6 test results and analysis, in: Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995, 1995. URL: https://aclanthology.org/M95-1019.

[13] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 260–270. URL: https://aclanthology.org/N16-1030. doi:10.18653/v1/N16-1030.

[14] J. Yu, B. Bohnet, M. Poesio, Named entity recognition as dependency parsing, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6470–6476. URL: https://aclanthology.org/2020.acl-main.577. doi:10.18653/v1/2020.acl-main.577.

[15] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, IEEE Transactions on Knowledge and Data Engineering 34 (2022) 50–70. doi:10.1109/TKDE.2020.2981314.

[16] S. Ali, K. Masood, A. Riaz, A. Saud, Named entity recognition using deep learning: A review, in: 2022 International Conference on Business Analytics for Technology and Security (ICBATS), 2022, pp. 1–7. doi:10.1109/ICBATS54253.2022.9759051.

[17] A. Alokaili, M. Menai, Svm ensembles for named entity disambiguation, Computing 102 (2020). doi:10.1007/s00607-019-00748-x.

[18] Z. Fang, Y. Cao, Q. Li, D. Zhang, Z. Zhang, Y. Liu, Joint entity linking with deep reinforcement learning, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 438–447. URL: https://doi.org/10.1145/3308558.3313517. doi:10.1145/3308558.3313517.

[19] W. Bouarroudj, Z. Boufaida, L. Bellatreche, Named entity disambiguation in short texts over knowledge graphs, Knowl. Inf. Syst. 64 (2022) 325–351. URL: https://doi.org/10.1007/s10115-021-01642-9. doi:10.1007/s10115-021-01642-9.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[21] L. Fan, L. Li, Z. Ma, S. Lee, H. Yu, L. Hemphill, A bibliometric review of large language models research from 2017 to 2023, arXiv preprint arXiv:2304.02020 (2023).

[22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[23] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled

attention, arXiv preprint arXiv:2006.03654 (2020).

[24] D. Jurafsky, J. H. Martin, Speech and language processing, 2021.

[25] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[26] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[27] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, arXiv preprint arXiv:2204.02311 (2022).

[28] E. Strubell, A. Ganesh, A. McCallum, Energy and policy considerations for deep learning in nlp, arXiv preprint arXiv:1906.02243 (2019).

[29] O. Sharir, B. Peleg, Y. Shoham, The cost of training nlp models: A concise overview, arXiv preprint arXiv:2004.08900 (2020).

[30] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, Z. Sui, A survey for in-context learning, arXiv preprint arXiv:2301.00234 (2022).

[31] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, arXiv preprint arXiv:1904.09751 (2019).

[32] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, M. Lewis, Measuring and narrowing the compositionality gap in language models, arXiv preprint arXiv:2210.03350 (2022).

[33] Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, Y. Goldberg, Measuring and improving consistency in pretrained language models, Transactions of the Association for Computational Linguistics 9 (2021) 1012–1031.

[34] Y. Zhang, A. Backurs, S. Bubeck, R. Eldan, S. Gunasekar, T. Wagner, Unveiling transformers with lego: a synthetic reasoning task, arXiv preprint arXiv:2206.04301 (2022).

[35] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473. URL: https://aclanthology.org/D19-1250. doi:10.18653/v1/D19-1250.

[36] B. AlKhamissi, M. Li, A. Celikyilmaz, M. Diab, M. Ghazvininejad, A review on language models as knowledge bases, arXiv preprint arXiv:2204.06031 (2022).

[37] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258 (2021).

[38] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Superglue: A stickier benchmark for general-purpose language understanding systems, Advances in neural information processing systems 32 (2019).

[39] J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, et al., Larger language models do in-context learning differently, arXiv preprint arXiv:2303.03846 (2023).

[40] H. Peng, X. Wang, S. Hu, H. Jin, L. Hou, J. Li, Z. Liu, Q. Liu, Copen: Probing conceptual

knowledge in pre-trained language models, arXiv preprint arXiv:2211.04079 (2022).

[41] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, et al., Kilt: a benchmark for knowledge intensive language tasks, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 2523–2544.

[42] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[43] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[44] M. Nickel, V. Tresp, H.-P. Kriegel, et al., A three-way model for collective learning on multi-relational data., in: Icml, volume 11, 2011, pp. 3104482–3104584.

[45] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Transactions on Knowledge Discovery from Data (TKDD) 15 (2021) 1–49.

[46] Z. Wang, J. Li, Z. Liu, J. Tang, Text-enhanced representation learning for knowledge graph, in: Proceedings of International joint conference on artificial intelligent (IJCAI), 2016, pp. 4–17.

[47] N. Kertkeidkachorn, R. Ichise, T2kg: An end-to-end system for creating knowledge graph from unstructured text, in: Workshops at the Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[48] A. Rossanez, J. C. Dos Reis, R. d. S. Torres, H. de Ribaupierre, Kgen: a knowledge graph generator from biomedical scientific literature, BMC medical informatics and decision making 20 (2020) 1–24.

[49] X. Han, Z. Liu, M. Sun, Neural knowledge acquisition via mutual attention between knowledge graph and text, Proceedings of the AAAI Conference on Artificial Intelligence 32 (2018). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11927. doi:10.1609/aaai.v32i1.11927.

[50] A. Bastos, A. Nadgeri, K. Singh, I. O. Mulang, S. Shekarpour, J. Hoffart, M. Kaul, Recon: relation extraction using knowledge graph context in a graph neural network, in: Proceedings of the Web Conference 2021, 2021, pp. 1673–1685.

[51] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, arXiv preprint arXiv:1909.03193 (2019).

[52] D. Daza, M. Cochez, P. Groth, Inductive entity representations from text via link prediction, in: Proceedings of the Web Conference 2021, 2021, pp. 798–808.

[53] V. Swamy, A. Romanou, M. Jaggi, Interpreting language models through knowledge graph extraction, in: 35th Conference on Neural Information Processing Systems (NeurIPS 2021), CONF, 2021.

[54] C. Aspillaga, M. Mendoza, Á. Soto, Inspecting the concept knowledge graph encoded by modern language models, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 2984–3000.

[55] G. A. Miller, WordNet: An electronic lexical database, MIT press, 1998.

[56] C. Wang, X. Liu, D. Song, Language models are open knowledge graphs, arXiv preprint

arXiv:2010.11967 (2020).

[57] J. Shen, C. Wang, Y. Yuan, J. Han, H. Ji, K. Sen, M. Zhang, D. Song, PALT: Parameter-lite transfer of language models for knowledge graph completion, in: Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3833–3847. URL: https://aclanthology.org/2022.findings-emnlp.281.

[58] J. Omeliyanenko, A. Zehe, L. Hettinger, A. Hotho, Lm4kg: Improving common sense knowledge graphs with language models, in: The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part I 19, Springer, 2020, pp. 456–473.

[59] R. Cohen, M. Geva, J. Berant, A. Globerson, Crawling the internal knowledge-base of language models, arXiv preprint arXiv:2301.12810 (2023).

[60] S. Hao, B. Tan, K. Tang, H. Zhang, E. P. Xing, Z. Hu, Bertnet: Harvesting knowledge graphs from pretrained language models, arXiv preprint arXiv:2206.14268 (2022).

[61] H. Li, Language models: past, present, and future, Communications of the ACM 65 (2022) 56–63.

[62] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240.

[63] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, arXiv preprint arXiv:1903.10676 (2019).

[64] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Legal-bert: The muppets straight out of law school, arXiv preprint arXiv:2010.02559 (2020).

[65] Z. Zhao, E. Wallace, S. Feng, D. Klein, S. Singh, Calibrate before use: Improving few-shot performance of language models, in: International Conference on Machine Learning, PMLR, 2021, pp. 12697–12706.

[66] S. Shin, S.-W. Lee, H. Ahn, S. Kim, H. Kim, B. Kim, K. Cho, G. Lee, W. Park, J.-W. Ha, et al., On the effect of pretraining corpora on in-context learning by a large-scale language model, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022, pp. 5168–5186.

[67] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, L. Zettlemoyer, Rethinking the role of demonstrations: What makes in-context learning work?, arXiv preprint arXiv:2202.12837 (2022).

[68] J. Kim, H. J. Kim, H. Cho, H. Jo, S.-W. Lee, S.-g. Lee, K. M. Yoo, T. Kim, Ground-truth labels matter: A deeper look into input-label demonstrations, arXiv preprint arXiv:2205.12685 (2022).

[69] P. H. Le-Khac, G. Healy, A. F. Smeaton, Contrastive representation learning: A framework and review, Ieee Access 8 (2020) 193907–193934.