A Survey of Knowledge Enhanced Pre-Trained Language Models

Linmei Hu, Zeyi Liu, Ziwang Zhao, Lei Hou, Liqiang Nie, Senior Member, IEEE, and Juanzi Li (Survey Paper)

Abstract—Pre-trained Language Models (PLMs) which are trained on large text corpus via self-supervised learning method, have yielded promising performance on various tasks in Natural Language Processing (NLP). However, though PLMs with huge parameters can effectively possess rich knowledge learned from massive training text and benefit downstream tasks at the fine-tuning stage, they still have some limitations such as poor reasoning ability due to the lack of external knowledge. Research has been dedicated to incorporating knowledge into PLMs to tackle these issues. In this paper, we present a comprehensive review of Knowledge Enhanced Pre-trained Language Models (KE-PLMs) to provide a clear insight into this thriving field. We introduce appropriate taxonomies respectively for Natural Language Understanding (NLU) and Natural Language Generation (NLG) to highlight these two main tasks of NLP. For NLU, we divide the types of knowledge into four categories: linguistic knowledge, text knowledge, knowledge graph (KG), and rule knowledge. The KE-PLMs for NLG are categorized into KG-based and retrieval-based methods. Finally, we point out some promising future directions of KE-PLMs.

Index Terms—Knowledge enhanced pre-trained language models, natural language generation, natural language processing, natural language understanding, pre-trained language models.

I. INTRODUCTION

With the continuous development of deep learning technologies in recent years, Pre-trained Language Models (PLMs) which are trained with unsupervised objectives over massive text corpora, have been widely used in the field of Natural Language Processing (NLP), and yielded state-of-the-art performance on various downstream tasks. Different from traditional supervised learning, PLMs based on self-supervised learning are usually pre-trained on general-purpose large-scale unlabeled data first, and then fine-tuned on small-scale labeled data for the specific tasks. Representative work, such as BERT [1], GPT [2], T5 [3], has refreshed benchmark records constantly in many Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks, successfully promoting the development of NLP.

As the size of PLMs grows larger, PLMs with billions of parameters have been extensively demonstrated to possess the ability of encoding rich linguistic [4], [5], [6] and factual knowledge [7], [8] implicitly into their parameters. However, due to the lack of the ability to leverage explicit knowledge from external knowledge sources, PLMs suffer from limited performance on downstream tasks. In particular, the prior study has found that traditional pre-training objectives often have weak symbolic reasoning capabilities, [9] since PLMs tend to concentrate on the word co-occurrence information.

Incorporating knowledge into PLMs can empower their memorization and reasoning abilities [10]. For instance, in the language understanding problem of “The monument to the people’s Heroes sits solemnly on [MASK] square”, traditional PLMs predict the output of the masked position as “the”, while knowledge enhanced PLMs predict “Tiananmen”, which is more accurate.

For language generation, although existing PLMs are able to obtain rich language information from text corpus and generate correct sentences, almost all of them fail to generate output towards capturing human commonsense, since they overlook the external world knowledge [11]. In other words, sentences generated by PLMs often conform to the grammatical norm, but not to logic. For example, given a concept set {hand, sink, wash, soap} to generate a sentence, conventional PLMs may generate "hands washing soap on the sink", while the PLM with extra knowledge generates "man is washing his hands with soap in a sink", which is more natural and logical.

To address the above issues, explicitly incorporating knowledge into PLMs has been an emerging trend in recent NLP studies. Wei et al. [12] reviewed the knowledge-enhanced PLMs along three taxonomies: types of knowledge sources, knowledge granularity, and application. Yin et al. [13] summarized the recent progress of pre-trained language model-based knowledge-enhanced models (PLMKEs) according to three crucial elements of them: knowledge sources, knowledge-intensive...
NLP tasks, and knowledge fusion methods. In this work, considering the fact that injecting knowledge into language models can promote both NLU and NLG tasks, and these two areas have different focuses, we aim to present a comprehensive review of Knowledge Enhanced Pre-trained Language Models (KE-PLMs) in these two areas to provide respective insights of KE-PLMs in NLU and NLG. KE-PLMs in NLU typically focus on fine-grained knowledge (e.g., lexical knowledge and syntax tree) that could assist natural language understanding. While KE-PLMs in NLG pay more attention on retrieving text knowledge or leveraging knowledge graph for better language generation. Each of these two fields is further divided into sub-categories based on the different types of used knowledge. The main contributions of this survey can be summarized as follows:

1) In this survey, we divide KE-PLMs into two main categories according to the downstream tasks: NLU and NLG. Appropriate taxonomies are respectively presented to highlight the focuses of these two different kinds of tasks in NLP.

2) For NLU, KE-PLMs are further divided into four sub-categories according to the types of knowledge: linguistic knowledge, text knowledge, knowledge graph (KG), and rule knowledge. For NLG, focused on the knowledge sources, KE-PLMs are further categorized into retrieval-based methods and KG-based methods. Fig. 1 shows our proposed taxonomies for NLU and NLG.

3) We discuss some possible directions that may tackle the existing problems and challenges of KE-PLMs in the future.

The rest of this paper is arranged as follows. In Section II, we provide the background of PLMs under the development of training paradigms in NLP. In Section III, we introduce the taxonomy of KE-PLMs in the field of NLU. In Section IV, we introduce the taxonomy of KE-PLMs in the field of NLG. For both NLU and NLG fields, we discuss the representative work of each leaf category in the taxonomy. In Section V, we propose the possible research directions of KE-PLMs in the future based on the existing limitations and challenges. Finally, we conclude in Section VI.

II. BACKGROUND

Pre-trained Language Models: Although the idea of pre-training on a language modeling task is not novel, the training paradigm has shifted to pre-train and fine-tune, with the emergence of ELMo [14] and ULMFiT [15]. Both are based on Long Short-Term Memory (LSTM) architecture. They propose to fine-tune the language model layer by layer for downstream tasks, and their performance demonstrates the competitiveness of pre-trained language models.

Unlike the early fully supervised method that learns salient features from limited data [16], language models can be trained on a large amount of raw textual data to obtain general-purpose representations. Then, the pre-trained models will be applied to various downstream tasks by fine-tuning them through task-specific objective functions [17]. For example, UNILM [18] unifies three language modeling objectives, which can be adapted to NLU and NLG tasks simultaneously.

As the Transformer architecture with multi-head self-attention mechanism is put forward [19], all popular language models, including GPT [2], BERT [1], BART [20] and T5 [3] are proposed based on the Transformer. The multi-head self-attention mechanism allows every word to attend to each other, making the models capture long-range dependencies and learn more expressive representations. These models differ in the model structure and training objectives. In particular, GPT is an autoregressive language model (unidirectional) which predicts the next word given all the previous words. BERT is a masked language model (bidirectional) that aims to predict the “masked” word conditioned on all the other words. BART and T5 are encoder-decoder language models that learn to generate a sequence when given an input sequence.

Prompt Learning: Instead of adapting PLMs to different downstream tasks by designing specific objective functions, “pre-train, prompt, and predict” which reformulates
downstream tasks through textual prompts has taken the place of “pre-train, fine-tune” to become the fourth paradigm in NLP [17]. In this paradigm, there is even no need for fine-tuning models using task-specific training objectives, PLMs themselves can directly be employed to predict the output that is desired, breaking through the problem of data constraints and bridging the gap of objective forms between pre-training and fine-tuning [7], [21], [22], [23], [24], [25], [26]. Previous work [17] conducts a comprehensive survey on the emerging field of prompt-based learning.

Though this prompt learning method has achieved promising results via constructing prompt information without changing the structure and parameters of PLMs significantly, it also calls for the necessity of choosing the most appropriate prompt template and verbalizer [22] which may have a great impact on model’s performance [22]. To this end, some work [27], [28], [29], [30], [31] has proposed to use knowledge as prompt to enhance the prompt-tuning process and reduce the cost of template construction and label mapping. Through injecting domain/task-relevant knowledge at the fine-tuning stage, PLMs can better serve downstream tasks and obtain better performance. In this survey, we also investigated the knowledge enhanced pre-trained language models that incorporate external knowledge via prompt learning.

**Large Language Models:** Since PLMs have raised the performance bar of NLP tasks, researchers further explore the effect of training even larger PLMs [32], [33]. The research shows that scaling PLMs typically results in an increased model capacity on downstream tasks, and when the parameter scale rises above a certain level, these larger-sized PLMs show surprising abilities that are absent from smaller PLMs. Thus the term large language models (LLMs) has been coined for the PLMs with significant size. Typically, LLMs often contain more than billions of parameters, usually larger than 10B [34]. The remarkable applications of LLMs include ChatGPT\(^1\) and GPT-4 [35], which exhibit extraordinary conversational capabilities with people.

Similar to PLMs, most of the existing LLMs also employ Transformer as their model architectures, and language modeling as their pre-training objectives. However, LLMs are much larger in model size, training data, and total computation than PLMs [34], making them more capable of general-purpose tasks and aligned with human values. The emergent abilities defined as “Abilities that are present in large models but not in smaller models in the literature” [36], are among the most noticeable features that characterize LLMs from PLMs. These abilities are crucial for LLMs to solve a series of complex tasks, thus improving the performance of language models greatly.

Despite the extraordinary performance achieved by LLMs, there still exists some major issues with knowledge utilization. LLMs may have difficulties in accomplishing tasks that need knowledge more recent than the training data, which requires LLMs to continuously update themselves with the latest data. While fine-tuning LLMs is costly and training LLMs incrementally may cause them to forget the old knowledge [37], some studies have explored to use external knowledge sources as the supplement. Atlas [38] leverages a retriever to retrieve relevant documents from large external knowledge source based on the input text, and jointly trains the retriever and the language model that fuses these documents with input text to enhance text generation. LLM-Augmenter [39] develops a plug-and-play (PhP) module to improve LLMs with external knowledge. Besides, LLMs tend to learn general language patterns from training data, which makes them struggle in tackling various specialized tasks [34]. Therefore, injecting more specific knowledge into LLMs is also an open problem that deserves further study.

### III. KE-PLMS for NLU

NLU is a subpart of NLP concerning all the methods that enable machines to understand and interpret the content of textual data. It extracts core semantic information from the unstructured text and applies this information to downstream tasks, thus playing a vital role in applications such as text classification, relation extraction, named entity recognition (NER), and dialogue system. In line with the taxonomy shown in Fig. 1, we divide the knowledge incorporated by KE-PLMs which are designed for NLU tasks into the following four categories according to their different types, that is, linguistic knowledge, text knowledge, knowledge graph, and rule knowledge. For each category, we discuss its representative methods.

#### A. Incorporating Linguistic Knowledge Into PLMs

Linguistic knowledge, mainly divided into lexical knowledge and syntax tree, is the most common auxiliary feature incorporated into PLMs [40]. Among them, lexical knowledge includes but is not limited to Part-of-Speech (POS) tagging and sentiment tags of words. LIBERT [41] introduces lexical relation classification (LRC) as a new pre-training task on the basis of standard BERT objectives, using synonyms and hypernym-hyponym pairs to predict whether two words are in specific semantic relations, which enhances the modeling ability of PLMs for semantic information. SenseBERT [42] integrates word-supersense (e.g., noun.food, noun.state) and predicts their corresponding supersenses by restoring the masked words, which can explicitly learn the semantic information of words in a given context. SKEP [43] improves the effect of PLM on the sentiment analysis task by integrating sentiment knowledge (sentiment words, polarity, and aspect-sentiment pairs). SentiPrompt [28] incorporates sentiment knowledge about aspects, opinions, and polarities into prompt through the construction of consistency and polarity judgment templates, and explicitly models term relations between aspect and opinion terms, better introducing task-related knowledge for the language models through prompt-tuning methods. LET [44] integrates semantic information of HowNet to improve the Chinese sentence matching task. KEAR [45] combines the knowledge from ConceptNet, dictionary entry definition, and labeled training data to enhance its performance on commonsense knowledge question answering. DictBERT [46] takes dictionary knowledge as the external source, and achieves knowledge enhancement by means of contrastive learning.

---

\(^{1}\)https://openai.com/blog/chatgpt/
Considering the different ways of incorporating syntax tree knowledge into PLMs, we divided the KE-PLMs into three categories as illustrated in Fig. 2, including introducing new relevant pre-training tasks [47], adopting new attention mechanism [48], [49], and designing new model structure [50]. For instance, LIMIT-BERT [47] realizes multi-task learning across five linguistic tasks, and simply sums up these task-specific losses together on the basis of a variety of linguistic knowledge such as POS tagging, semantic role labeling (SRL), dependency relations, syntax trees, and etc. in model training. Syntax-BERT [48] provides additional syntax information by constructing a syntax tree parser, and generates a new attention mechanism according to the constructed syntax tree. Specifically, different from standard BERT, they decompose the self-attention network into multiple sub-networks based on the tree structure, where each sub-network contains one relation in the syntax trees. Then they adopt a syntax-aware attention network to aggregate syntactic representations learned from different sub-networks. Syntax-augmented BERT [50] introduces a syntax-based graph neural network to fuse the syntax information from dependency trees to improve PLM.

Despite the above methods consistently incorporating linguistic knowledge into PLMs, they differ in the stages of knowledge fusion. LIBERT [41], SenseBERT [42], SKEP [43], SensePrompt [29], LET [44], KEAR [45], and DictBERT [46] fuse linguistic knowledge in the pre-training stage of PLMs to enhance the representation of the input, while SentiPrompt [28], LET [44], SLA [49], Syntax-augmented BERT [50] and KEAR [45] fuse knowledge in the fine-tuning stage of PLMs for improving task performance.

B. Incorporating Text Knowledge Into PLMs

Text knowledge is usually retrieved from general-domain text collection (such as WikiText [51], Wiktionary [45]) or large corpus (such as Wikipedia [52]). KNN-LM [51] selects the nearest K neighbors from training samples as knowledge incorporated into PLM, and its earlier idea comes from cache-LM [53] which remains the first K words in the cache. REALM [52] utilizes text corpus to train a text retriever explicitly, exploiting information retrieved from external knowledge bases such as Wikipedia documents to help the prediction of masked tokens. ExpBERT [54] and KEAR [45] also incorporate textual descriptions into their models to improve performance. OK-Transformer [55] incorporates large-scale out-of-domain commonsense descriptions to enhance the representation of input text. Specifically, for the input text, they first retrieve the corresponding commonsense descriptions from a large-scale knowledge base (i.e. ATOMIC2020) and use them as extra input. They adopt Transformer to encode the input text and each commonsense description, and then integrate all the commonsense representations into one commonsense embedding, fusing it with input text embedding to achieve knowledge enhancement. Kformer [56] retrieves text knowledge from external knowledge bases, and injects the embedded knowledge into the FFN layer of Transformer. REINA [57]
retrieves some training samples similar to the input from external datasets as knowledge to enhance PLM. UniK-QA [58] and UDT-QA [59] utilize text, knowledge graph and table knowledge simultaneously, and transform all the knowledge into text for knowledge enhancement. KiC [60] employs an external memory involving different types of knowledge such as dictionary and commonsense, and leverages a retriever to select useful knowledge pieces for the input instance. Then the knowledge-augmented input is fed into a text-to-text generation model.

In addition to the type of general text knowledge that has been mentioned above, some work also utilizes domain-specific corpora or scientific and technical literature for pre-training tasks. In particular, BioBERT [61] and SciBERT [62] conduct the pre-training process on large-scale scientific domain corpora and achieve promising results on downstream academic NLP tasks. S2ORC-BERT [63] leverages the same method as SciBERT on a larger corpus which contains numerous academic papers covering dozens of academic disciplines, and slightly promotes its performance on several downstream tasks.

As for the stage of fusing knowledge, REALM [52], SciBERT [62], BioBERT [61], and S2ORC-BERT [63] integrate text knowledge in the pre-training stage, while KNN-LM [51], ExpertBERT [54], KEAR [45], OK-Transformer [55], Kformer [56], REINA [57], UniK-QA [58], and UDT-QA [59] integrate knowledge in the fine-tuning stage.

C. Incorporating Knowledge Graph Into PLMs

Knowledge graph can be considered as a kind of powerful expression representing real world knowledge in the structural form of graph, where its nodes represent entities, and edges represent relations between entities [64], [65]. Compared with other types of knowledge, such as text knowledge which is unstructured, knowledge graph often contains more abundant structured information which makes it more applicable to enhance the potential learning capability of models [66], [67], [68], [69]. Here, we further divide KG into entity knowledge and triplet knowledge as shown in Fig. 3.

1) Entity Knowledge: As the most basic semantic unit, entity plays a vital role in many natural language scenarios, such as machine reading comprehension, NER, sentiment analysis, and etc [70]. Incorporating entity knowledge into PLMs helps to improve the semantic understanding ability of models and their performance in downstream tasks as well. According to the different ways of incorporating entity knowledge, we divide them into three sub-categories as follows.

One is to design entity related pre-training tasks [71], [72], [73], [74], [75]. ERNIE [71] proposes a multi-stage knowledge masking strategy of both entity level and phase level. WKLM [72] integrates knowledge of external entities through the entity replacement strategy, and determines whether each entity is replaced by performing binary classification prediction. KECP [73] adopts contrastive learning and prompt learning to integrate entity knowledge.

The second is to change the attention mechanism of the model. For example, LUKE [76] introduces an entity-aware self-attention mechanism to capture the entity information in calculating attention scores. Specifically, they treat both words and entities in the target text as input tokens, and employ different query matrices to compute attention scores between token pairs with different types (i.e., word-word pair, word-entity pair, entity-word pair, and entity-entity pair).
The third is to change the model structure. K-BERT [81] employs a knowledge fusion module to provide structural information for the language module. K-BERT [81] leverages an N-layer T-encoder structure which is similar to the BERT-base model to extract text information, and then uses a proposed K-encoder to integrate entity knowledge. KnowBert [78] retrieves entity embeddings related to the input text via an entity linker and then updates contextual word representations through word-to-entity attention, allowing the long-range interactions between words and all entities. EaE [79] expands each entity mention embedding in the input text by fusing it with the top K relevant entity embeddings retrieved by a proposed component called an entity memory layer. CokeBERT [80] develops a semantic-driven graph neural network to dynamically select knowledge context from KGs that has high relevance with entities in the input textual context, incorporating more semantic knowledge into text modeling. JAKET [81] proposes a joint training framework for knowledge graph and text, in which the language module encodes the contextual embeddings of text, while the knowledge module learns knowledge-based entity embeddings from KGs to provide structural information for the language module. KIC [60] employs an external memory involving different types of knowledge including entity knowledge for text generation.

Note that several methods may belong to more than one sub-category, and we assign them to the sub-category according to their most significant contributions.

2) Triplet Knowledge: In addition to the entity knowledge mentioned above, there are also a great number of relational triples in the knowledge graph, which can provide sufficient structured information for PLMs and also improve the semantic understanding ability of them. Similar to the KE-PLMs incorporating entity knowledge, we also divide KE-PLMs in this category into three sub-categories according to their specific ways of knowledge incorporation.

One is to design pre-training tasks related to triplets [74], [82], [75], [83]. ERICA [74] introduces both entity and relation discrimination tasks to deepen PLM's understanding of entities and relations through contrastive learning. KEPLER [82] trains knowledge embedding and masked language modeling objectives jointly to improve the knowledge representation. DKPLM [75] focuses on long-tail entities, enriching semantic information of low-frequency entities with knowledge graphs. Specifically, they first detect long-tail entities in the input sample, and replace them with corresponding knowledge triplets. And then, they design a special pre-training task to predict the replaced entities and relations. The knowledge injection process only happens in the pre-training stage. KP-PLM [83] designs two knowledge-aware pre-training tasks to incorporate knowledge triplets into multiple natural language prompts for NLU tasks. Specifically, they first construct a relevant knowledge sub-graph for the input sentence, and generate knowledgeable prompts from this sub-graph based on the designed prompt mapping rules. Then, these derived prompts are concatenated with the original input and used as the unified input of PLMs. In Table I, we summarize entity and triplet relevant pre-training tasks designed by the existing work.

The second is to change the attention mechanism of the model [84], [85], [45]. K-BERT [84] leverages a knowledge layer to inject relevant triplets from KG into the input sentence and transform it into a knowledge-rich sentence tree. Then this sentence tree is converted into a visible matrix to control the visible area of each word in the sentence, preventing the sentence from deviating from the correct semantics due to injecting too much knowledge. If K-BERT intends to expand input text into a sentence tree, the core concept of CoLAKE [85] is to expand the input context into word-knowledge graphs (WK graphs), and then feed these constructed WK graphs into masked self-attention to gather information of nodes. KEAR [45] proposes an external attention mechanism to enhance the Transformer architecture, and integrate external knowledge into its prediction process. We illustrate their attention mechanism in Table II, where \( Q \), \( K \), \( V \) denote the query, key, and value matrices, respectively. \( d_k \) is the dimension of the key, which is used as the scaling factor.

The third is to change the model structure, which usually introduces a knowledge fusion module [30], [31], [58], [59], [81], [86], [87], [88], [89], [90], [91], [92], as shown in Fig. 4(a). FAE [86] introduces an additional memory module of facts on the basis of EaE [79], so that it can effectively combine the information in the symbolic knowledge graph. Besides the layer structure of the original pre-trained model, K-ADAPTER [87] and KB-adapters [88] incorporate knowledge into PLM through external adapter modules. KLMO [89] uses a component named knowledge aggregator to fuse the embeddings of the input text and KG, which applies an entity-level cross-KG attention to interactively model entity segments in text along with entities and relations in KG, as shown in Fig. 4(b). KERM [90] designs a knowledge injector module that combines the knowledge between text corpus and KG for passage re-ranking task as shown in Fig. 4(c). JointLK [91] and GreaseLM [92] exploit GNNs for modeling extracted knowledge graphs and couple the LM with GNN modules to perform joint reasoning for commonsense reasoning. JAKET [81] proposes a joint training framework that models both text and knowledge graph to jointly learn the contextual and structural knowledge information. UniK-QA [58] and UDT-QA [59] develop unified knowledge representation architectures, to convert knowledge from heterogeneous sources including triplets, text and tables into text format for open-domain question answering. Besides, KnowPrompt [30] incorporates entity and relation knowledge contained in the relation labels into prompt templates, and inserts these templates into the input text for relation extraction. OntoPrompt [31] linearizes

| Method          | Pre-training Tasks/Objectives                                      |
|-----------------|-------------------------------------------------------------------|
| ERNIE [71]      | Token-level, Phrase-level and Entity-level MLM                     |
| WCLM [72]       | Entity replacement, MLM                                           |
| KECP [73]       | Token-level MLM, Span-level contrastive learning                   |
|ERICA [74]       | Entity and relation discrimination tasks, MLM                      |
| DKPLM [75]      | Relational knowledge decoding, Token-level MLM                     |
| KP-PLM [83]     | Prompt relevance inspection, Masked prompt modeling               |
| KEPLER [82]     | Knowledge embedding, MLM                                          |

Here MLM represents masked language modeling.
TABLE II
EXAMPLES OF CHANGING ATTENTION MECHANISM FOR INCORPORATING TRIPLET KNOWLEDGE

| Method        | Attention Mechanism     | Formalized expression                                      | Variation in Calculation                                                                 |
|---------------|-------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------|
| K-BERT [84]   | Masked Self-Attention   | $\text{Attn}(Q, K, V) = \text{Softmax} \left( \frac{QK^T + M}{\sqrt{d_k}} \right) V$ | visible matrix $M$ for the input sentence tree, $M_{ij} = 0$ if token $i$ and token $j$ are in the same branch, while $M_{ij} = -\infty$ if not |
| CoLAKE [85]   | Masked Self-Attention   | $\text{Attn}(Q, K, V) = \text{Softmax} \left( \frac{QK^T}{\sqrt{d_k}} + M \right) V$ | mask matrix $M$ for word-knowledge graph, $M_{ij} = 0$ if node $i$ and node $j$ are connected, while $M_{ij} = -\infty$ if not |
| KEAR [45]     | External Attention      | $\text{Attn}(Q, K, V) = \text{Softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$ | concatenate extra knowledge $K$ into the input text $H_0 = [X; K]$ |

Fig. 4. (a) Incorporating triplet knowledge through a knowledge fusion module. (b) KLMO designs a knowledge aggregator to fuse knowledge into the input token sequence. (c) KERM develops a knowledge injector to integrate knowledge explicitly.

In addition to using knowledge graph as auxiliary information to improve PLMs’ ability of language understanding, some work also learns a knowledge embedding representation while incorporating triplet knowledge, so that PLMs can complete some tasks related to knowledge reasoning, such as entity classification, relation prediction, knowledge graph completion and etc. Representative work includes ERICA [74], KEPLER [82], KLMO [89], FaE [86] mentioned above.

3) Fusion Stage: The above mentioned methods ERNIE [71], WKLM [72], LUKE [76], EaE [79], CoLAKE [85], FaE [86], ERICA [74], KEPLER [82], KLMO [89], DKPLM [75], JAKET [81], KP-PLM [83], and KERM [90] are pre-fusion methods that fuse knowledge in the pre-training stage, K-BERT [84], K-ADAPTER [87], KECP [73], KEAR [45], KB-adapters [88], JointLK [91], GreaseLM [92] are post-fusion methods which fuse knowledge in the fine-tuning stage. While some methods like ERNIE-THU [77] and KnowBert [78] fuse knowledge in both pre-training and fine-tuning stages. The two fusion stages are also called the training stage and reasoning stage, respectively. For example, CoLAKE [85] jointly learns the embeddings of entities and relations during the training phase, while K-BERT [84] injects triples from KG during the reasoning phase.

D. Incorporating Rule Knowledge Into PLMs

Logic rules always contain clear logical reasoning processes, and can formalize knowledge from external sources [93]. Incorporating this type of knowledge into PLMs can facilitate the demonstration of reasoning path via its good interpretability. For example, RuleBERT [94] utilizes the Horn rules of existing corpus to establish a training dataset and then fine-tunes the model on it. It adopts a probabilistic answer set programming to predict the probability of events, and tries to learn soft rules from PLM. The results show PLMs that reason with soft rules over natural language could improve their performance for deductive reasoning tasks. Besides, PTR [27] incorporates logic rules to construct task-specific prompts composed with sub-prompts designed manually, so that the model can encode task-related prior knowledge in the prompt-tuning and generate prompts that are more interpretable. Both RuleBERT and PTR incorporate rule knowledge in the fine-tuning stage.

E. Benchmarks in NLU

A large variety of knowledge-aware NLU tasks have been employed to evaluate the capabilities of KE-PLMs. Here, we summarize the existing dataset benchmarks including general language understanding evaluation (i.e. GLUE and SuperGLUE) datasets as well as task-specific datasets. Data statistics of task-specific datasets are shown in Table V. Descriptions of these benchmarks are listed as follows.
GLUE [95] is a widely used NLU benchmark to measure the performance of language models. It contains nine natural language understanding tasks.

SuperGLUE [96] is a GLUE-style benchmark aimed at conducting a more rigorous evaluation of language understanding. It consists of eight language understanding tasks which are more difficult.

LAMA (LAnceModel Analysis) probe [7] aims to evaluate the factual knowledge grasped in a language model via clone-style statement. It provides a series of knowledge sources including Google-RE, T-Rex, ConceptNet, and SQuAD. LAMA-UHN [97] tasks are then constructed by filtering out easy-to-answer samples. Both LAMA and LAMA-UHN are proposed to fairly compare the knowledge understanding abilities of KE-PLMs. The performance of some KE-PLMs is shown in Table III.

TACRED [98] is a large-scale supervised relation extraction dataset, which contains 119,474 examples from 42 relations and 106,264 sentences.

FewRel [99] is a few-shot relation classification dataset, which consists of 70,000 sentences on 100 relations constructed with Wikipedia. Evaluation of KE-PLMs on TACRED and FewRel datasets is shown in Table IV.

OpenEntity [100] is an entity typing dataset which contains 6 entity types and 2,000 instances. Given a sentence with an entity mention, the task is to predict a set of phrases that describe the types of the target entity.

FIGER [101] is a sentence-level entity typing dataset constructed with Wikidata. The task is to provide entity mentions with fine-grained types. Evaluation of KE-PLMs on OpenEntity and FIGER datasets is shown in Table VII.

CoNLL-2003 [102] is a named entity recognition dataset involving English and German two languages. It concentrates on four types of named entities: persons, locations, organizations, and names of miscellaneous entities.

CommonsenseQA [103] is a 5-way multiple-choice question-answering dataset utilized for commonsense reasoning. It contains 12,102 questions constructed with ConceptNet.

OpenBookQA [104] is a 4-way multiple-choice question-answering dataset for commonsense reasoning. It consists of 5,957 questions along with an open book of scientific facts.

Natural Questions [105] is an open-domain question-answering dataset. Questions are queries mined from Google search engine, and the answers are annotated in Wikipedia articles.

WebQuestions [106] is a dataset for open-domain question answering. It consists of questions from Google Suggest API, which target Freebase entities as the answers.

TriviaQA [107] is an open-domain question-answering dataset, which contains a set of question-answer pairs from trivia and quiz-league websites.

SST-2 [108] is a dataset for sentence-level sentiment classification. The task is binary sentence classification, which is to predict sentiment (positive or negative) for sentences collected from movie reviews.

Table IV summarizes the evaluation of KE-PLMs on TACRED and OpenEntity datasets. They are the most commonly used benchmarks for relation extraction and entity typing tasks respectively. Both tasks take macro precision (P), recall rate (R), and F1 score as evaluation metrics.

### Table III

| Method | LAMA | LAMA-UHN | LAMA-UHN |
|--------|------|----------|----------|
|        | Google-RE | T-Rex | UHN-Google-RE | UHN-T-Rex |
| ELMo [14] | 2.2 | 0.2 | 2.3 | 0.2 |
| BERT [1] | 11.4 | 32.5 | 5.7 | 23.3 |
| E2E [79] | 9.4 | 37.4 | - | - |
| CoLAKE [85] | 9.5 | 28.8 | 4.9 | 20.4 |
| DKPLM [75] | 10.8 | 32.0 | 5.4 | 22.9 |
| KP-PLM [83] | 11.0 | 32.3 | 5.6 | 22.5 |

### Table IV

| Method         | TACRED | OpenEntity |
|----------------|--------|------------|
|                | P      | R | F1 | P | R | F1 |
| ERNIE-THU [77] | 69.97  | 66.08 | 67.97 | 78.42 | 72.90 | 75.56 |
| KnowBert [78]  | 71.60  | 71.40 | 71.50 | 78.60 | 73.70 | 76.10 |
| KEPLER [82]    | 71.50  | 72.50 | 72.00 | 77.80 | 74.60 | 76.20 |
| CoLAKE [85]    | -      | -    | -    | 77.00 | 75.70 | 76.40 |
| K-ADAPTER [87] | 70.14  | 74.04 | 72.04 | 78.99 | 76.27 | 77.61 |
| DKPLM [75]     | 72.61  | 73.53 | 73.07 | 79.20 | 75.90 | 77.50 |
| KP-PLM [83]    | 73.30  | 73.90 | 73.50 | 80.50 | 76.10 | 78.20 |

IV. KE-PLMS FOR NLG

The goal of NLG is to enable machines to generate language texts that can be understood by humans and follow the way where humans express themselves. Incorporating various forms of knowledge into generation models other than input sequences helps to improve the performance of text generation tasks. Referring to the survey on knowledge-enhanced text generation [40], we further divide KE-PLMs in the field of NLG into two categories based on their different knowledge sources: one is retrieval-based method and the other is KG-based method.

A. Incorporating Retrieval-Based Knowledge Into PLMs

The retrieval-based methods mainly integrate and utilize additional knowledge related to the input sequences through retrieval. Other than the input sequence itself, the additional knowledge is retrieved from external sources such as online search engines, large data sets, and training sets to guide the generation process. Considering whether the methods focus on re-ranking the retrieved items for generation or not, we divide these retrieval-based methods into two sub-methods as shown in the left of Fig. 5: one is retrieval augmented generation method that aims to improve generation by retrieving related knowledge [57], [109], [110], [111], [112], [113], [114], [115], [116], and the other is retrieve, rerank and rewrite method which focuses on re-ranking retrieved items for generation [117], [118].

The flow charts of these two kinds of methods are shown in
**TABLE V**

**KNOWLEDGE-GROUNDED NLU DATASET BENCHMARKS**

| Task                        | Dataset   | # Train | # Dev  | # Test  | Model                  |
|-----------------------------|-----------|---------|--------|---------|------------------------|
| Relation classification    | TACRED    | 75,050  | 25,764 | 18,660  | DietBERT [46], ExpBERT [54], ERICA [74], DKPLM [75], KnowBERT [76], KP-PLM [83], LUKE [76], KEPLER [82], ERNIE-THU [77], K-ADAPTER [87], PIR [27] |
| /extraction                 | FewRel    | 448,000 | 112,000 | 140,000 | KP-PLM [83], CoLAKE [85], ERNIE-THU [77], JAKET [81], KEPLER [82] |
| Entity typing               | OpenEntity| 200     | 200    | 200     | DKPLM [73], LUKE [76], ERNIE-THU [77], KnowBERT [78], KEPLER [82], KP-PLM [83], CoLAKE [85], K-ADAPTER [87] |
|                             | FIGER     | 2,000,000 | 10,000 | 563     | WKLM [72], ERICA [74], K-ADAPTER [87] |
| Named entity recognition    | CoNLL-2003| 203,621 | 51,362 | 46,435  | DietBERT [46], SLA [49], LUKE [76] |
| Commonsense reasoning       | CommonsenseQA [103] | 9,741 | 1,221 | 1,140 | KEAR [43], DietBERT [46], OK-Transformer [55], JointLk [91], GreaseLM [92] |
|                             | OpenBookQA [104] | 4,957 | 500   | 500    | DietBERT [46], JointLk [91], GreaseLM [92] |
| Question answering          | Natural Questions [105] | 307,373 | 7,830 | 7,842 | REALM [52], UniK-QA [58], UDT-QA [59], ERICA [74] |
|                             | WebQuestions [106] | 3,778 | -     | 2,032 | REALM [52], UniK-QA [58], UDT-QA [59], WKLM [72], EA [79], FaE [86] |
|                             | TriviaQA   | 67,291 | 11,274 | 10,790 | UniK-QA [58], WKLM [72], ERICA [74], EaE [79] |
| Sentiment classification    | SST-2      | 8644   | 1101   | 2210   | SKEP [43], Syntax-BERT [48], SLA [49] |
| Knowledge probing           | LAMA       | -      | -      | -      | -                      |

*Notice that the splits of datasets might be different in studies.*

**TABLE VI**

**KNOWLEDGE-GROUNDED NLG DATASET BENCHMARKS**

| Task                        | Dataset           | # Train | # Dev  | # Test  | Model                  |
|-----------------------------|-------------------|---------|--------|---------|------------------------|
| Dialogue generation         | Wizard of Wikipedia [109] | 15,430  | 1,948  | 1,933  | MemNet [109], CNTF [124], PLUG [118], SKT [117], CNTF [124] |
|                             | CMU_Doc [129]     | 3,373   | 229    | 619    | KG-FID [113], RAG [110], RETRO [115] |
| Question answering          | Natural Questions [105] | 307,373 | 7,830  | 7,842  | KG-FID [113], RAG [110] |
|                             | TriviaQA [107]    | 87,291  | 11,274 | 10,790 | KG-BART [111], KFCNet [111], KGR4 [112] |
| Commonsense reasoning       | CommonGen [130]   | 67,389  | 4,018  | 6,042  | GRF [125], MoKGE [127] |
|                             | oNLG-ART [131]    | 50,481  | 7,252  | 2,976  | GRF [125], MoKGE [127], CE-PR [120] |
| Explanation reasoning       | ComVE [132]       | 25,596  | 1,428  | 2,976  | KEPM [122], GRF [125] |
| Story generation            | ROCStories [133]  | 90,000  | 4,081  | 4,081  | REINA [57] |
| Summarization               | CNN/Dailymail [134] | 257,226 | 13,368 | 11,490 | REINA [57] |
|                             | XSum [139]        | 204,045 | 11,332 | 11,334 | REINA [57] |

*Notice that the splits of datasets might be different in studies.*

Fig. 5. Further categorization of retrieval-based method and KG-based method. The left figure demonstrates the categorization of retrieval-based method, and the right demonstrates the categorization of KG-based method.
TABLE VII
COMPARISONS BETWEEN PLMS AND KE-PLMS ON CommonsGen BENCHMARKS

| Method         | BLEU-3/4 | ROUGE-2/L | METEOR | CIDEr | SPICE |
|----------------|----------|-----------|--------|-------|-------|
| T5 [3]         | 39.00    | 28.60     | 22.01  | 42.57 | 30.10 |
| BART [20]      | 36.30    | 26.30     | 22.23  | 41.58 | 26.90 |
| KG-BART [11]   | 42.10    | 30.90     | 23.38  | 44.54 | 32.40 |
| KFCNet [111]   | 57.33    | 51.46     | 26.81  | 47.92 | 38.92 |
| KGR4 [112]     | -        | 42.82     | -      | -     | 18.42 |

Fig. 6. Flow charts of retrieval-based methods. (a) demonstrates the retrieval augmented generation method. (b) demonstrates the retrieve, rerank and rewrite method. Notably, the retrieved candidate templates will be reranked before rewriting.

Fig. 6. Notably, these KE-PLMs incorporate external knowledge in the fine-tuning stage to improve their performance on downstream tasks.

In the line of retrieval augmented generation method, MemNet [109] proposes a Transformer memory network that can retrieve topic-relevant knowledge based on the dialogue history, and then generate the next dialogue utterance with the help of this retrieved knowledge. RAG [110] leverages a retriever to find the top-K relevant documents given the input sequence, and uses them as additional context when generating the target text. Specifically, they employ a pre-trained neural retriever to search documents from Wikipedia based on the input sample. Then, they simply concatenate these documents with the input, and send the combined input to a seq-to-seq transformer to generate text. KFCNet [111] first retrieves prototypes that comprise concepts in the given concept set, while keeping these retrieved results semantically similar to the target sentences. Then it applies two contrastive learning modules in both encoder and decoder to capture global target information and learn general features from the multiple retrieved prototypes. REINA [57] retrieves some training samples which are similar to the input text and takes them as knowledge to improve the effect of machine translation. KGR4 [112] divides the generation of commonsense into four stages, that is, retrieval of commonsense, generation of commonsense by means of generation models, refinement and correction of the generated commonsense statements, and scoring of the generated statements. Best results can be obtained through this process. KG-Fid [113] is based on the Fusion-in-Decoder (FID) [119] model, but it proposes to solve the validity and efficiency of FID with the help of KG, which improves the ability of open domain question answering significantly. SeeKeR [114] leverages a set of documents retrieved from the search engine to generate knowledge response. This method can incorporate up-to-date information through its three-module framework (search, knowledge generation, and final response). RETRO [115] retrieves from large text databases and builds trillions of tokens as retrieve sources to expand the language model. UnifiedSKG [116] unifies six categories of tasks (i.e., semantic parsing, question answering, data-to-text generation, fact verification, conversation, and formal language-to-text translation tasks) into a text-to-text format and introduces linearized knowledge to strengthen their performance.

Representative work of the retrieve, rerank and rewrite method, such as SKT [117], regards knowledge selection as a sequential decision-making process, and uses the sequential latent variable model to improve the accuracy of knowledge selection in multiple rounds of dialogue. PLUG [118] retrieves related knowledge from Wikipedia, dictionary, and knowledge graph, and then ranks them based on statistical and semantic information for knowledge-grounded dialogue generation.

B. Incorporating KG-Based Knowledge Into PLMs

In order to distinguish the granularity of knowledge utilized by different KE-PLMs more specifically, we divide existing work into three categories: knowledge extracted by path finding, triplet knowledge, and subgraph knowledge extracted from KG as shown in the right of Fig. 5.

The first is to extract knowledge by path finding [120], [121]. In this way, the relation path is clearly reasoned to make reliable decisions. CE-PR [120] and MRG [121] mainly perform explicit reasoning on relation paths, significantly improving the effectiveness of text generation. Specifically, CE-PR first retrieves a subgraph given the source concepts, and then scores each triple on this graph. It propagates scores along the paths to each node from the source concepts and preserves the nodes with higher scores. MRG leverages the reasoning module to infer paths step by step from the source concepts, and then uses the sentence realization (i.e., sentence generation) module to generate a complete sentence based on the inferred paths. Fig. 7 demonstrates the inference process of these two methods.

The second sub-category is KE-PLMs based on triplet knowledge [122], [123], [124]. KEPM [122] converts commonsense triplets in the knowledge bases into natural language statements based on templates to provide additional information for story generation. MixGEN [123] adopts an encoder-decoder framework to incorporate expert knowledge (from dataset annotation), explicit knowledge (from knowledge graph), and implicit knowledge (from generative PLM) to produce implications of toxic text. CNTF [124] models the commonsense, named entities and
topic-specific knowledge via a multi-hop attention module to facilitate the dialogue generation task.

The third one is subgraph knowledge [11, 125, 126], [127, 128]. Different from the first two categories, subgraph contains context information on related concepts, which plays an important role in understanding related concepts and language generation. This kind of method generally uses GNNs to model extracted subgraphs, and then fuse these extracted subgraphs to enhance the natural language generation ability.

For the integration of subgraph knowledge, we further divide them based on the specific position they integrate as shown in Fig. 8. Some work incorporates subgraph knowledge into only encoder to improve the language understanding ability, such as JointGT [126], MoKGE [127], and PLAN [128]. For example, JointGT [126] is constructed for KG-to-text generation task, which first linearizes the given knowledge graph into a text sequence as input, and then utilizes a structure-aware semantic aggregation module at Transformer encoder to combine the graph structural information. The structural knowledge of KGs is effectively integrated for the decoding process to generate a target text sequence aligned with the input graph. Some works inject subgraph knowledge into the decoder, such as GRF [125]. In this way, every step of decoding can be traced, so that the model can be better interpreted. Others introduce subgraph knowledge into both encoder and decoder, such as KG-BART [11]. Specifically, KG-BART is designed to leverage KGs for generative commonsense reasoning. Given a set of commonsense concepts, it first constructs a concept-reasoning graph containing relations between concepts, and a concept-expanding graph that enriches the structure between concepts with neighborhood similarity.

Then, these two graphs are utilized as additional input for the encoder and decoder based on the graph attention mechanism respectively, incorporating KG-augmented knowledge into the concept representation process.

Among the above mentioned KE-PLMs for NLG, JointGT [126], UnifiedSKG [116], PLUG [118] are pre-fusion methods which fuse knowledge during pre-training, and MemNet [109], RAG [110], SKT [117], CE-PR [120], MRG [121], KEPM [122], GRF [125], KFCNet [111], KG-BART [11], KG4 [112], KG-FiD [113], SeeKeR [114], RETRO [115], MixGEN [123], CNTF [124], MoKGE [127] are post-fusion methods which fuse knowledge during fine-tuning.

In Tables IX and X, we summarize the existing KE-PLMs for NLU and NLG.

C. Benchmarks in NLG

General evaluation benchmarks for generation are critical for facilitating research in NLG. Existing generation benchmarks

Fig. 8. Integration position of subgraph knowledge. (a) Incorporating KG into the encoder. (b) Incorporating KG into the decoder. (c) Incorporating KG into both the encoder and decoder.

| Method     | # params | Natural Q. | Web Q. | TriviaQA |
|------------|----------|------------|--------|----------|
| T5 [3]     | 11,318M  | 34.5       | 37.4   | 50.1     |
| EeE [79]   | 367M     | 39.0       | -      | 53.4     |
| REALM [52] | 330M     | 40.4       | 40.7   | -        |
| UniK-QA[58] | 990M    | 54.0       | 57.8   | 64.1     |
| UDT-QA [59] | 1,320M  | 55.2       | 52.0   | -        |
| RAG [110]  | 626M     | 44.5       | 45.2   | 56.8     |
| KG-FiD [113] | 994M   | 53.4       | -      | 69.8     |

Exact match scores are reported.
| Method          | Knowledge Type      | Fusion in Pre-training | Fusion in Fine-tuning                              | NLU or NLG tasks                                                                 |
|-----------------|---------------------|------------------------|---------------------------------------------------|----------------------------------------------------------------------------------|
| LIBERT [41]     | lexical             | Yes                    |                                                   | lexical simplification, sentence/sentence-pair classification, NLI                |
| SenseBERT [42]  | lexical             | Yes                    |                                                   | word supersense disambiguation, WiC                                              |
| SKEP [43]       | lexical             | Yes                    |                                                   | sentence/aspect-level sentiment classification, opinion role labeling            |
| SentiPrompt [28]| lexical             | Yes                    |                                                   | triplet extraction, pair extraction, aspect term extraction                     |
| LET [44]        | lexical             | Yes                    |                                                   | Chinese short text matching                                                     |
| KEAR [45]       | lexical, general text, triplet | Yes |                                                   | commonsense reasoning                                                           |
| DictBERT [46]   | lexical             | Yes                    | Yes                                               | NER, RE, commonsense reasoning, GLUE                                             |
| LIMIT-BERT [47] | syntax tree         | Yes                    |                                                   | syntactic parsing, semantic parsing                                             |
| Syntax-BERT [48]| syntax tree         | Yes                    |                                                   | sentiment classification, NLL, GLUE                                              |
| SLA [49]        | syntax tree         | Yes                    |                                                   | sentiment classification, NER, grammatical error detection                      |
| Syntax-augmented BERT [50] | syntax tree | Yes                          |                                                   | semantic role labeling, NER, RE                                                |
| KNN-LM [51]     | general text        | Yes                    |                                                   | language modeling                                                              |
| REALM [52]      | general text        | Yes                    |                                                   | open-QA                                                                         |
| ExpBERT [54]    | general text        | Yes                    |                                                   | RE                                                                              |
| OK-Transformer [55] | general text     | Yes                          |                                                   | commonsense reasoning, text classification                                      |
| Kformer [56]    | general text        | Yes                    |                                                   | commonsense reasoning, medical QA                                               |
| UniK-QA [58]    | general text, triplet | Yes                     |                                                   | multi-source QA                                                                |
| UDT-QA [59]     | general text, triplet | Yes                     |                                                   | open-domain QA                                                                 |
| SciBERT [62]    | domain-specific text | Yes                        |                                                   | sequence tagging, sentence classification, dependency parsing                    |
| BioBERT [61]    | domain-specific text | Yes                        |                                                   | biomedical NER, RE, QA                                                          |
| S2ORC-BERT [63] | domain-specific text | Yes                        |                                                   | inline citation detection, bibliography parsing, bibliography linking            |
| ERNIE [71]      | entity              | Yes                    |                                                   | NLI, semantic similarity, NER, sentiment analysis, QA                           |
| WKLM [72]       | entity              | Yes                    |                                                   | QA, ET                                                                          |
| ERICA [74]      | entity, triplet     | Yes                    |                                                   | RE, ET, QA                                                                      |
| KECP [73]       | entity              | Yes                    |                                                   | extractive QA                                                                  |
| DKPLM [75]      | entity, triplet     | Yes                    |                                                   | knowledge probing, RE, ET                                                       |
| LUKE [76]       | entity              | Yes                    |                                                   | ET, RC, NER, close-style/extractive QA                                          |
| ERNIE-THU [77]  | entity              | Yes                    |                                                   | ET, RC, GLUE                                                                    |
| KnowBert [78]   | entity              | Yes                    |                                                   | RE, WiC, ET                                                                     |
| EAE [79]        | entity              | Yes                    |                                                   | knowledge probing, open-domain QA, RE                                            |
| JAKET [81]      | entity, triplet     | Yes                    |                                                   | RC, QA over KG, entity classification                                            |
| KEPLER [82]     | entity, triplet     | Yes                    |                                                   | RC, ET, GLUE, link prediction                                                   |
| KP-PLM [83]     | triplet             | Yes                    |                                                   | knowledge probing, RE, ET                                                       |
| K-BERT [84]     | triplet             | Yes                    |                                                   | sentence classification, QA, NER                                                |
| CoLAKE [85]     | triplet             | Yes                    |                                                   | ET, RE, knowledge probing, GLUE, knowledge graph completion                     |
| FaE [86]        | entity, triplet     | Yes                    |                                                   | open-domain QA                                                                  |
| KLMO [89]       | entity, triplet     | Yes                    |                                                   | ET, RC                                                                          |
| K-ADAPTER [87]  | general text, triplet | Yes                      |                                                   | ET, QA, RC                                                                       |
| KB-adapters [88]| entity, triplet     | Yes                    |                                                   | knowledge-probing using response selection, fact memorization, response generation |

Here: RC: relation classification; RE: relation extraction; ET: entity typing; NER: named entity recognition; QA: question answering; WiC: words in context; NLI: natural language inference; GLUE: general language understanding evaluation.
TABLE X
SUMMARIZATION OF KE-PLMS

| Method   | Knowledge Source                        | Fusion in Pre-training | Fusion in Fine-tuning | NLU or NLG tasks                      |
|----------|----------------------------------------|------------------------|-----------------------|---------------------------------------|
| KERM [90] | triplet                                | Yes                    |                       | passage re-ranking                    |
| JointLK [91] | triplet                             | Yes                    |                       | commonsense reasoning                 |
| GreaselM [92] | triplet                            | Yes                    |                       | commonsense reasoning, medical QA    |
| RuleBERT [94] | rule                                         | Yes                    |                       | rule reasoning                        |
| PTR [27] | rule                                   | Yes                    |                       | RC                                    |
| MemNet [109] | retrieved text                        | Yes                    |                       | open-domain dialogue generation       |
| RAG [110] | retrieved text                         | Yes                    |                       | open-domain/abstractive QA, Jeopardy question generation, fact verification |
| KFCNet [111] | retrieved text                       | Yes                    |                       | commonsense/keyword generation        |
| REINA [57] | general text, retrieval augmented generation | Yes               |                       | summarization, language modeling, machine translation, QA |
| KGR4 [112] | retrieved text                         | Yes                    |                       | commonsense reasoning                 |
| KG-FiD [113] | retrieved text                        | Yes                    |                       | open-domain QA                        |
| SeeKeR [114] | retrieved text                        | Yes                    |                       | open-domain dialogue, prompt completion |
| RETRO [115] | retrieved text                        | Yes                    |                       | language modelling, QA                |
| UnifiedSKG [116] | structured knowledge               | Yes                    |                       | structured knowledge grounding        |
| SKT [117] | retrieved text                         | Yes                    |                       | knowledge-grounded dialogue           |
| PLUG [118] | retrieved text, knowledge graph        | Yes                    |                       | knowledge-grounded dialogue, conversational recommendation |
| CE-PR [120] | knowledge graph                      | Yes                    |                       | commonsense explanation generation    |
| MRG [121] | knowledge graph                       | Yes                    |                       | story/review/description generation   |
| KEPM [122] | knowledge graph                       | Yes                    |                       | story generation                      |
| MixGEN [123] | knowledge graph, expert knowledge     | Yes                    |                       | toxicity explanation                 |
| CNTF [124] | knowledge graph                       | Yes                    |                       | dialogue generation                   |
| GRF [125] | knowledge graph                       | Yes                    |                       | story ending generation, abductive NLG, explanation generation |
| KG-BART [11] | knowledge graph                     | Yes                    |                       | commonsense reasoning/QA             |
| JointGT [126] | knowledge graph                  | Yes                    |                       | KG-to-text generation                |
| MoKGE [127] | knowledge graph                       | Yes                    |                       | commonsense explanation generation, abductive commonsense reasoning |

*Here RC: relation classification, QA: question answering.*

such as GLGE [136] and KiIT [137], do not especially concentrate on the tasks utilized for knowledge-enhanced generation [40]. And most of the KE-PLMs are developed just for specific NLG tasks. Therefore, we summarize the existing dataset benchmarks used for different knowledge-grounded NLG tasks as shown in Table VI. Descriptions of these datasets are listed as follows.

**Wizard of Wikipedia** [109] is a knowledge-grounded open-domain dialogue dataset. It covers 1,365 conversation topics and 22,311 dialogues between two speakers, one of whom can respond based on the knowledge retrieved from Wikipedia.

**CMU_DoG** [129] is a document-grounded conversation dataset. It contains 4,112 conversations concerned about the movie domain, and conversations are happened based on the contents of a certain document.

**Natural Questions** [105] is an open-domain question-answering dataset. Questions are real queries that come from the Google search engine. Annotators generate the answers from Wikipedia page that contains the information required for answering the question.

**TriviaQA** [107] is an open-domain question-answering dataset, which contains 95,956 question-answer pairs written by trivia enthusiasts.

**CommonGen** [130] is a generative commonsense reasoning dataset. It consists of 77,000 commonsense descriptions over 35,000 unique concept sets. Given a set of common concepts, the generation task is to construct a cohesive sentence describing a scenario based on them.

**αNLG-ART** [131] is a dataset for abductive commonsense reasoning. The task is also referred as abductive commonsense generation (NLG), which is to generate a valid hypothesis given the incomplete observation scenarios.

**ComVE** [132] is a dataset for commonsense explanation generation. It is one of the subtasks from SemEval-2020 Task 4 [138], which aims at generating an explanation given a counterfactual statement for sense-making.
ROCStories [133] is a dataset for story generation and story understanding evaluation, which contains 50,000 five-sentence commonsense stories.

CNN/DailyMail [134] a text summarization dataset characterized by long documents and multi-sentence summaries. It contains 311, 672 document-summary pairs which come from CNN and Daily Mail websites.

XSum [135] is a summarization dataset composed of 226,711 articles from BBC. The single-document summarization task aims to create a one-sentence summary that leverages information from various parts of the article.

Table VII summarizes the evaluation of KE-PLMs on CommonGen datasets. And Table VIII summarizes the evaluation of KE-PLMs both in NLU and NLG on three question-answering datasets.

V. FUTURE DIRECTIONS

In this section, we propose some possible research directions of KE-PLMs in the future, which may meet the existing problems and challenges.

A. Integrating Knowledge From Homogeneous and Heterogeneous Sources

Since most of the existing work only utilizes knowledge from a single source, such as knowledge graph or web resource, exploring how to integrate knowledge from heterogeneous sources is still a valuable direction for future research.

As we present in the section above, some prior work has tried to incorporate different types of knowledge to improve the performance of question-answering. For example, UniKQA [58] integrates external knowledge including text, tables, and relational triplets in the knowledge base. Through the heuristic method of linearizing heterogeneous knowledge sources including knowledge base (KB) triples and semi-structured tables into text, it unifies structured knowledge involved in KBQA and unstructured knowledge involved in TextQA, expanding the sources of external knowledge. UDT-QA [59] introduces structured knowledge such as knowledge graphs and tables into open-domain question answering, and converts them into open-domain question answering.

In the field of open-domain question answering, improving the ability of PLMs to integrate multiple knowledge sources can effectively increase the knowledge coverage, so that models can generate more reliable answers.

B. Exploring Multi-Modal Knowledge

Most of the current research focuses merely on text knowledge with fewer multi-modal sources. In fact, images, videos, and audio in addition to textual and tabulated information can also become the knowledge sources of PLMs, which can further improve the performance of KE-PLMs.

Several studies have explored integrating multi-modal knowledge. Representative work includes KB-VLP [139], and ERNIE-VIL [140]. KB-VLP [139] extracts knowledge information from the external knowledge base based on both the input text and image, and uses the knowledge as additional inputs to enhance the model’s ability of semantic alignment and knowledge perception. ERNIE-VIL [140] parses the input description texts of images into structured scene graphs, and designs cross-modal pre-training tasks to pay attention to detailed semantic alignments across vision and language modalities.

Since images and associated text contain rich semantics, the injection of these different modalities of knowledge and concentration on detail semantics can make them complement and enhance each other, which will boost the performance of PLMs on both NLU and NLG tasks.

C. Providing Interpretability Evidence

Although many existing KE-PLMs have achieved great success on a series of text generation tasks, it should not be ignored that, if the generation process requires commonsense knowledge reasoning, the performance of models will be affected.

Some work has attempted to tackle this problem [52], [87], [110]. For example, GRF [125] utilizes external knowledge graphs for explicit commonsense reasoning, and incorporates rich structural information in order to perform dynamic multi-hop reasoning on multiple relational paths. Reasoning paths obtained in this process provide a theoretical basis for the generation of results. This work suggests that, giving an explicit reasoning path will help improve the interpretability of models and make predictions more rational.

D. Learning Knowledge in a Continuous Way

Existing work is usually trained on a large number of static or non-updated data in the pre-training stage. But models may forget the original knowledge learned before when facing new tasks, which leaves them vulnerable to a phenomenon called catastrophic forgetting problem [141]. With the continuous growth of knowledge from heterogeneous sources, exploring methods to make models master new knowledge while not forgetting the previous one learned in the past requires continual learning (also called life-long learning) to integrate various knowledge constantly.

ELLE [142] proposes an extension module that maintains the network function to expand the width and depth of the model, so that the model can effectively acquire new knowledge and retain the old to a greater extent at the same time. K-ADAPTER [87] and KB-adapters [88] adds the adapters with PLMs to store factual and linguistic knowledge, so as to continuously incorporate more knowledge into PLM.

Incorporating knowledge continuously is a promising direction in future research [40]. The application of continuous and increasing pre-training will effectively improve the universality of PLMs, and solve the catastrophic forgetting problem while incorporating more knowledge.

E. Optimizing the Efficiency of Incorporating Knowledge Into Large Models

The scale of pre-trained models and injection of knowledge has become increasingly large in recent years [143], thus
brings severe challenges to the computational efficiency and computational resources that cannot be ignored. Though most of the existing work has achieved good results in various pre-training tasks, few studies mention the cost of knowledge fusion in the process.

In view of this challenge, we propose the following two possible directions that may be worth further exploring: one is to improve the efficiency of knowledge acquisition and filtering, and the other is to optimize the computational burden.

Existing work, such as ZeRO [144], has been explored in the second area. Based on traditional data parallel training mode, ZeRO deeply optimizes the redundant space and eliminates the memory occupied by redundancy through dividing the parameters, gradients, and optimizer states of the model into different processes.

F. Increasing the Variety of Results Generated

It is a vital research direction in NLG to generate alternative outputs or predict all possible results for the real situation, which is also the purpose of output diversity in the generative commonsense reasoning task. Existing work, such as MoKGE [127], uses the diversified knowledge reasoning of commonsense knowledge graph to complete the diversity generation of NLG. Based on the observation of human annotations, the concepts related to original input are associated into the generation process, and the mixture of expert method is used to generate diversified reasonable outputs, thus increasing the diversity of generated results.

VI. CONCLUSION

In this survey, we present a comprehensive review of KE-PLMs from the perspective of NLU and NLG, and respectively propose proper taxonomies for both NLU and NLG to highlight their different focuses. We also discuss the representative works in the taxonomies, as well as the evaluation benchmarks. Finally, in view of the existing problems and challenges, we discuss potential future research directions of KE-PLMs, hoping to facilitate relevant research in this promising area.

REFERENCES

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol., 2019, pp. 4171–4186.

[2] A. Radford et al., “Improving language understanding by generative pre-training,” OpenAI Blog, 2018. [Online]. Available: https://openai.com/research/language-unsupervised/

[3] C. Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” J. Mach. Learn. Res., vol. 21, pp. 140:1–140:67, 2020.

[4] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith, “Linguistic knowledge and transferability of contextual representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol., 2019, pp. 1073–1094.

[5] Y. Belinkov and J. R. Glass, “Analysis methods in neural language processing: A survey,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol., 2019, pp. 3348–3354.

[6] I. Vulic, E. M. Ponti, R. Litschke, G. Glavas, and A. Korhonen, “Prob- ing pretrained language models for lexical semantics,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2020, pp. 7222–7240.

[7] F. Petroni et al., “Language models as knowledge bases?,” in Proc. Conf. Empirc Methods Natural Lang. Joint Conf. Natural Lang. Process., 2019, pp. 2463–2473.

[8] S. Li et al., “How pre-trained language models capture factual knowledge? A causal-inspired analysis,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2022, pp. 1720–1732.

[9] A. Talmor, Y. Elazar, Y. Goldberg, and J. Berant, “olMLps - on what language model pre-training captures,” Trans. Assoc. Comput. Linguistics, vol. 8, pp. 743–758, 2020.

[10] Y. Sun et al., “ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation,” 2021, arXiv:2107.02157.

[11] Y. Liu, Y. Wen, L. He, H. Peng, and S. Y. Philip, “KG-BART: Knowledge graph-augmented bert for generative commonsense reasoning,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 6418–6425.

[12] X. Wei, S. Wang, D. Zhang, P. Bhatia, and A. Arnold, “Knowledge- enhanced pretrained language models: A comprehensive survey,” 2021, arXiv:2110.08455.

[13] D. Yin et al., “A survey of knowledge-intensive NLP with pre-trained language models,” 2022, arXiv:2202.08772.

[14] M. E. Peters et al., “Deep contextualized word representations,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol., 2018, pp. 2227–2237.

[15] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2018, pp. 328–339.

[16] X. Liu et al., “Self-supervised learning: Generative or contrastive?,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 1, pp. 857–876, Jan. 2023.

[17] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” 2021, arXiv:2107.13586.

[18] L. Dong et al., “Unified language model pre-training for natural language understanding and generation,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 13 042–13 054.

[19] A. Vaswani et al., “Attention is all you need,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[20] M. Lewis et al., “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 7871–7880.

[21] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, “Auto-Prompt: Eliciting knowledge from language models with automatically generated prompts,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2020, pp. 4222–4235.

[22] T. Schick and H. Schütze, “Exploiting cloze-questions for few-shot text classification and natural language inference,” in Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguistics, 2021, pp. 255–269.

[23] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” in Proc. Int. Joint Conf. Natural Lang. Process., 2021, pp. 4882–4897.

[24] Y. Gu, X. Han, Z. Liu, and M. Huang, “PPT: Pre-trained prompt tuning for few-shot learning,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2022, pp. 8410–8423.

[25] N. Zhang et al., “Differentiable prompt makes pre-trained language models better few-shot learners,” in Proc. Int. Conf. Learn. Representations, 2022, pp. 1–19.

[26] N. Ding et al., “OpenPrompt: An open-source framework for prompt-learning,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2022, pp. 105–113.

[27] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, “PTR: Prompt tuning with rules for text classification,” 2021, arXiv:2105.11259.

[28] C. Li et al., “SentiPrompt: Sentiment knowledge enhanced prompt-tuning for aspect-based analysis,” 2021, arXiv:2109.08306.

[29] S. Hu et al., “Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2022, pp. 2225–2240.

[30] X. Chen et al., “KnowPrompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction,” in Proc. ACM Conf. Web Conf., 2022, pp. 2778–2787.

[31] H. Ye et al., “Ontology-enhanced prompt-tuning for few-shot learning,” in Proc. ACM Web Conf., 2022, pp. 778–787.

[32] T. B. Brown et al., “Language models are few-shot learners,” in Proc. Int. Conf. Neural Inf. Process. Syst., vol. 33, pp. 1877–1901, 2020.

[33] A. Chowdhery et al., “PaLM: Scaling language modeling with pathways,” 2022, arXiv:2204.05605.

[34] W. X. Zhao et al., “A survey of large language models,” 2023, arXiv:2303.18223.
Y. Yao, S. Huang, L. Dong, F. Wei, H. Chen, and N. Zhang, “Kformer: Enhancing natural language representation with natural language explanations,” in Proc. Conf. Empir. Methods Natural Lang. Joint Conf. Natural Lang. Process., 2019, pp. 3613–3618.

K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. S. Weld, “S2ORC: The semantic scholar open research corpus,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 4969–4983.

A. Hogan et al., “Knowledge graphs,” ACM Comput. Surveys, vol. 54, no. 4, pp. 1–37, 2022.

B. Xie and L. Zou, “Knowledge graph quality management: A comprehensive survey,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 5, pp. 4969–4988, May 2023.

Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: A survey of approaches and applications,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 12, pp. 2724–2743, Dec. 2017.

X. Wang, F. Luo, Q. Wu, and Z. Bao, “How context or knowledge can benefit healthcare question answering,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 1, pp. 575–588, Jan. 2023.

L. Fang, Y. Luo, K. Feng, K. Zhao, and A. Hu, “A knowledge-enriched ensemble method for word embedding and multi-sense embedding,” IEEE Trans. Knowl. Data Eng., vol. 36, no. 6, pp. 5534–5549, Nov. 2020.

Q. Li, D. Wang, S. F. K. Song, Y. Zhang, and G. Yu, “OERL: Enhanced representation learning via open knowledge graphs,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 9, pp. 8880–8892, Sep. 2023.

Y. Zhao, H. Zhou, A. Zhang, R. Xie, Q. Li, and F. Zhuang, “Connecting embeddings based on multiplex relational graph attention networks for knowledge graph entity typing,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 5, pp. 4608–4620, May 2023.

Y. Sun et al., “ERNIE: Enhanced representation through knowledge integration,” 2019, arXiv:1904.09223.

W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, “Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–13.

J. Wang et al., “KECP: Knowledge enhanced contrastive prompting for few-shot extractive question answering,” 2022, arXiv:2205.03071.

Y. Qin et al., “ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process., 2021, pp. 3380–3386.

T. Zhang et al., “DKPLM: Decomposable knowledge-enhanced pre-trained language model for natural language understanding,” in Proc. AAAI Conf. Artif. Intell., 2022, pp. 11 703–11 711.

I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, “LUKE: Deep contextualized entity representations with entity-aware self-attention,” in Proc. Conf. Empir. Methods Natural Lang. Process., 2020, pp. 6442–6454.

Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, “ERNIE: Enhanced language representation with informative entities,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 1441–1451.

M. E. Peters et al., “Knowledge enhanced contextual word representations,” in Proc. Conf. Empir. Methods Natural Lang. Joint Conf. Natural Lang. Process., 2020, pp. 43–54.

T. Févy, L. B. Soares, N. FitzGerald, E. Choi, and T. Kwiatkowski, “Entities as experts: Sparsely memory access with entity supervision,” in Proc. Conf. Empir. Methods Natural Lang. Process., 2020, pp. 4937–4951.

Y. Su et al., “CokeBERT: Contextual knowledge selection and embedding towards enhanced pre-trained language models,” AI Open, vol. 2, pp. 127–134, 2021.

D. Yu, C. Zhu, Y. Yang, and M. Zeng, “JAKET: Joint pre-training of knowledge graph and language understanding,” in Proc. AAAI Conf. Artif. Intell., 2022, pp. 11 630–11 638.

X. Wang et al., “KEPLER: A unified model for knowledge embedding and pre-trained language representation,” Trans. Assoc. Comput. Linguistics, vol. 9, pp. 176–194, 2021.

J. Wang et al., “Knowledge prompting in pre-trained language model for natural language understanding,” 2022, arXiv:2210.08536.

W. Liu et al., “K-BERT: Enabling language representation with knowledge graph,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 2901–2908.

T. Sun et al., “ColALKE: Contextualized language and knowledge embedding,” in Proc. Int. Conf. Comput. Linguistics, 2020, pp. 3660–3670.
[86] P. Verga, H. Sun, L. B. Soares, and W. W. Cohen, “Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge,” 2020, arXiv: 2007.00849.

[87] R. Wang et al., “K-adapter: Infusing knowledge into pre-trained models with adapters,” in Proc. Int. Joint Conf. Natural Lang. Process., 2021, pp. 1405–1418.

[88] D. Emlen, D. Bonadiman, S. Alqahtani, Y. Zhang, and S. Mansour, “Injecting domain knowledge in language models for task-oriented dialogue systems,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2022, pp. 11962–11974.

[89] L. He, S. Zheng, T. Liang, and F. Zhang, “KLMo: Knowledge graph enhanced pretrained language model with fine-grained relationships,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2021, pp. 4536–4542.

[90] Q. Dong et al., “Incorporating explicit knowledge in pre-trained language models for passage re-ranking,” in Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2022, pp. 1490–1501.

[91] Y. Sun, Q. Shi, L. Qi, and Y. Zhang, “JointK: Joint reasoning with language models and knowledge graphs for commonsense question answering,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2022, pp. 5049–5060.

[92] X. Zhang et al., “GreaseLM: Graph reasoning enhanced language models,” in Proc. Int. Conf. Learn. Representations, 2022, pp. 1–16.

[93] L. Von Rueden et al., “Informed machine learning—a taxonomy and survey of integrating knowledge into learning systems,” 2019, arXiv: 1903.12394.

[94] M. Saeed, N. Ahmadi, P. Nakov, and P. Papotti, “RuleBERT: Teaching NLP tasks with rule-based knowledge,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2021, pp. 1460–1476.

[95] A. Wang, A. Singh, I. Michael, F. Hill, O. Levy, and S. R. Bowman, “GLUE: A multi-task benchmark and analysis platform for natural language understanding,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 353–355.

[96] A. Wang et al., “SuperGLUE: A stickier benchmark for general-purpose language understanding systems,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 3261–3275.

[97] N. Pörner, U. Waltinger, and H. Schütze, “BERT is not a knowledge base (yet): Factual knowledge versus name-based reasoning in unsupervised QA,” 2019, arXiv: 1911.03681.

[98] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, “Position-aware attention and supervised data improve slot filling,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 205–215.

[99] X. Han et al., “FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 4803–4809.

[100] E. Choi, O. Levy, Y. Choi, and L. Zettlemoyer, “Ultra-fine entity typing,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2018, pp. 87–96.

[101] X. Ling, S. Singh, and D. S. Weld, “Design challenges for entity linking,” in Trans. Assoc. Comput. Linguistics, vol. 3, pp. 315–328, 2015.

[102] E. F. T. K. Sang and F. D. Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol., 2003, pp. 142–147.

[103] A. Talmor, J. Herzig, N. Lourie, and J. Berant, “CoresenseQA: A question answering challenge targeting commonsense knowledge,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics - Hum. Lang. Technol., 2019, pp. 4149–4158.

[104] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, “Can a suit of armor conduct electricity? A new dataset for open book question answering,” in Proc. Adv. Neural Info. Process. Syst., 2020, pp. 9459–9474.

[105] H. Li, Y. Gong, J. Jiao, R. Zhang, T. Baldwin, and N. Duan, “KFCNNet: Knowledge filtering and contrastive learning network for generative commonsense reasoning,” 2021, arXiv: 2109.06704.

[106] X. Liao et al., “KGR4: Retrieval, retrospective, refine and rethink for commonsense generation,” in Proc. AAAI Conf. Artif. Intell., 2022, pp. 11 029–11 037.

[107] D. Yu et al., “KG-FID: Infusing knowledge graph in fusion-in-decoder for open-domain question answering,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2022, pp. 4961–4974.

[108] K. Shuster, M. Komellli, L. Adolphs, S. Roller, A. Szlam, and J. Weston, “Language models that seek for knowledge: Modular search & generation for dialogue and prompt completion,” 2022, arXiv: 2203.13224.

[109] S. Borgeaud et al., “Improving language models by retrieving from trillions of tokens,” in Proc. Int. Conf. Mach. Learn., 2022, pp. 2206–2240.

[110] T. Xie et al., “UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models,” 2022, arXiv: 2201.05966.

[111] B. Kim, J. Ahn, and G. Kim, “Sequential latent knowledge selection for knowledge-grounded dialogue,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–14.

[112] Y. Li et al., “Knowledge-grounded dialogue generation with a unified knowledge representation,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2022, pp. 206–218.

[113] G. Izacard and E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” in Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguistics, 2021, pp. 874–880.

[114] H. Ji, P. Ke, S. Huang, F. Wei, and M. Huang, “Generating commonsense explanation by extracting bridge concepts from reasoning paths,” in Proc. Conf. Asia-Pacific Chapter Assoc. Comput. Linguistics, Int. Joint Conf. Natural Lang. Process., 2020, pp. 248–257.

[115] L. Zhao, J. Xu, J. Lin, Y. Zhang, H. Yang, and X. Sun, “Graph-based multi-hop reasoning for long text generation,” 2020, arXiv: 2009.13282.

[116] J. Guan, F. Huang, M. Huang, Z. Zhao, and X. Zhu, “A knowledge-enhanced pretraining model for commonsense story generation,” Trans. Assoc. Comput. Linguistics, vol. 8, pp. 93–108, 2020.

[117] R. Sridhar and D. Yang, “Explaining toxic text via knowledge enhanced text generation,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2022, pp. 811–826.

[118] D. Varshney, A. Prabhakar, and A. Etkal, “Commonsense and named entity aware knowledge grounded dialogue generation,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2022, pp. 1322–1335.

[119] H. Ji, P. Ke, S. Huang, F. Wei, and M. Huang, “Language generation with multi-hop reasoning on commonsense knowledge graph,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2020, pp. 725–736.

[120] P. Ke et al., “JointGT: Graph-text joint representation learning for text generation from knowledge graphs,” in Proc. Int. Joint Conf. Natural Lang. Process., 2021, pp. 2526–2538.

[121] W. Yu, C. Zhu, L. Qin, Z. Zhang, T. Zhao, and M. Jiang, “Diversifying content generation for commonsense reasoning with mixture of knowledge graph experts,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2022, pp. 1896–1906.

[122] Y. Lu et al., “Neuro-symbolic procedural planning with commonsense prompting,” in Proc. Int. Conf. Learn. Representations, 2023.

[123] K. Zhou, S. Prabhumoye, and A. W. Black, “A dataset for document grounded conversations,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 708–713.

[124] B. Y. Lin et al., “CommonGen: A constrained text generation challenge for generative commonsense reasoning,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2020, pp. 1823–1840.

[125] C. Bhatgavatula et al., “Construct commonsense reasoning,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–18.

[126] C. Wang, S. Liang, Y. Zhang, X. Li, and T. Gao, “Does it make sense? and why? A pilot study for sense making and explanation,” in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 4020–4026.

[127] N. Mostafazadeh et al., “A corpus and cloze evaluation for deeper understanding of commonsense stories,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2018, pp. 839–849.
Linmei Hu received the PhD degree from Tsinghua University, in 2018. She works as an associate professor with the School of Computer Science and Technology, Beijing Institute of Technology. Her research interests include knowledge graph, natural language processing, and multimodal content analysis. She has published some papers in top conferences and journals, such as ACL, AAAI, EMNLP, ACM SIGKDD, IEEE MultiMedia, ACM Transactions on Information Systems, and IEEE Transactions on Knowledge and Data Engineering.

Zeyi Liu receive the BS degree from the University of Science and Technology Beijing, in 2023. She is currently working toward the master’s degree in computer science and technology with the Beijing University of Posts and Telecommunications.

Ziwang Zhao received the BS degree from Xidian University, Xi’an, China, in 2021. He is currently working toward the master’s degree in computer science and technology with the Beijing University of Posts and Telecommunications. His research interests include pre-trained models, knowledge graph, and multimodal.

Lei Hou received the PhD degree from Tsinghua University, in 2016. He is an assistant professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include knowledge graph construction and application, news and user-generated content mining. He has published many research papers on top-tier international conferences, such as ACL, EMNLP, AAAI, etc.

Liqiang Nie (Senior Member, IEEE) received the BEng degree from Xi’an Jiaotong University, and the PhD degree from the National University of Singapore (NUS). He is IAPR fellow, is currently the dean with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen). His research interests include primarily in multimedia content analysis and information retrieval. He has co-/authored more than 100 CCF-A papers and 5 books, with 15 k plus Google Scholar citations. He is an AE of IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, ACM Transactions on Multimedia Computing, Communications, and Applications, and Information Science. Meanwhile, he is the regular area chair or SPC of ACM MM, NeurIPS, IJCAI and AAAI. He is a member of ICME steering committee. He has received many awards over the past three years, like ACM MM and SIGIR best paper honorable mention in 2019, the AI 2000 most influential scholars 2020, SIGMM rising star in 2020, MIT TR35 China 2020, DAMO Academy Young Fellow in 2020, SIGIR best student paper in 2021, first price of the provincial science and technology progress award in 2021 (rank 1), and provincial youth science and technology award in 2022. Some of his research outputs have been integrated into the products of Alibaba, Kwai, and other listed companies.

Juanzi Li received the PhD degree from Tsinghua University, in 2000. She is a professor with the Department of Computer Science and Technology, Tsinghua University. Her research interests include knowledge engineering and semantic web, text and social network mining. She has published more than 100 research papers in top international journals and conferences such as IEEE Transactions on Knowledge and Data Engineering, KDD, AAAI, ACL, EMNLP, etc.