KILM: Knowledge Injection into Encoder-Decoder Language Models

Yan Xu[1,2*], **Mahdi Namazifar**[1], **Devamanyu Hazarika**[1], **Aishwarya Padmakumar**[1], **Yang Liu**[1], **Dilek Hakkani-Tür**[1]
[1]Amazon Alexa AI
[2]Hong Kong University of Science and Technology
yxucb@connect.ust.hk, mahdinam@amazon.com, dvhaz@amazon.com
padmakua@amazon.com, yangliud@amazon.com, hakkanit@amazon.com

Abstract

Large pre-trained language models (PLMs) have been shown to retain implicit knowledge within their parameters. To enhance this implicit knowledge, we propose Knowledge Injection into Language Models (KILM), a novel approach that injects entity-related knowledge into encoder-decoder PLMs, via a generative knowledge infilling objective through continued pre-training. This is done without architectural modifications to the PLMs or adding additional parameters. Experimental results over a suite of knowledge-intensive tasks spanning numerous datasets show that KILM enables models to retain more knowledge and hallucinate less while preserving their original performance on general NLU and NLG tasks. KILM also demonstrates improved zero-shot performances on tasks such as entity disambiguation, outperforming state-of-the-art models having 30x more parameters.[1]

1 Introduction

Large pre-trained language models (PLMs) (Radford et al., 2019; Lewis et al., 2020a; Raffel et al., 2020) have achieved great success across all NLP tasks. However, recent studies also reveal that PLMs are susceptible to memorizing the pre-training corpora rather than capturing the knowledge within them (Niven and Kao, 2019; Talmor et al., 2020; Yasunaga et al., 2022; Li et al., 2022). Particularly for generation tasks, PLMs are notorious for hallucinating text that is factually incorrect or hard to verify (Logan et al., 2019; Sun et al., 2020; Lin et al., 2020; Longpre et al., 2021). To address these issues, one approach is to retrieve relevant knowledge and integrate it explicitly with PLMs (He et al., 2020; Liu et al., 2021b). Another direction is incorporating the additional knowledge sources into the pre-training step (Zhang et al., 2019; Xiong et al., 2019; Liu et al., 2022; Wang et al., 2021b). While the former suffers from the issue of falling back on the models themselves without retrieved information (Krishna et al., 2021), knowledge-focused pre-training can be complementary to those methods (Longpre et al., 2021) and shows its advantage on generalization.

In this paper, we propose an approach for injecting knowledge into encoder-decoder PLMs, such as BART, as a continued pre-training process. We refer to it as *Knowledge Injection into Language Models* (KILM). Instead of introducing additional parameters to PLMs or modifying the model architectures to incorporate additional knowledge, KILM infills knowledge sentences by adopting a novel knowledge infilling objective that includes a knowledge reconstruction step in addition to the original pre-training objectives of BART.

The aim of KILM is to teach PLMs additional content about concepts and entities that they encounter in a given context, so that the models are able to ground an entity mention with additional information and "describe" what that entity is (see Figure 1). It should be emphasized that in this process, the context is especially important for cases when an entity mention can refer to multiple entities, e.g., *Titanic* which can refer to the *British ship* or to the *1997 movie*. We utilize the *short descriptions* of entities in Wikipedia which comprise of entity definitions as the knowledge source (§3.1). Although there are existing works leveraging similar knowledge for PLM enhancement, they ignore the relationship among entities, contexts, and entity-centric knowledge, and restrict their applications to NLU tasks. In contrast, we propose a distinct structure (§3.2) to augment Wikipedia articles with short descriptions of the entity mentions in the context, thus model this essential relationship, so as to force PLMs to learn the correlation among entities and contexts, and differentiate between the entities with similar surface forms during

[*]Work done in part while Yan was an intern at Amazon Alexa AI.
[1]The code is available at https://github.com/alexa/kilm.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5013–5035

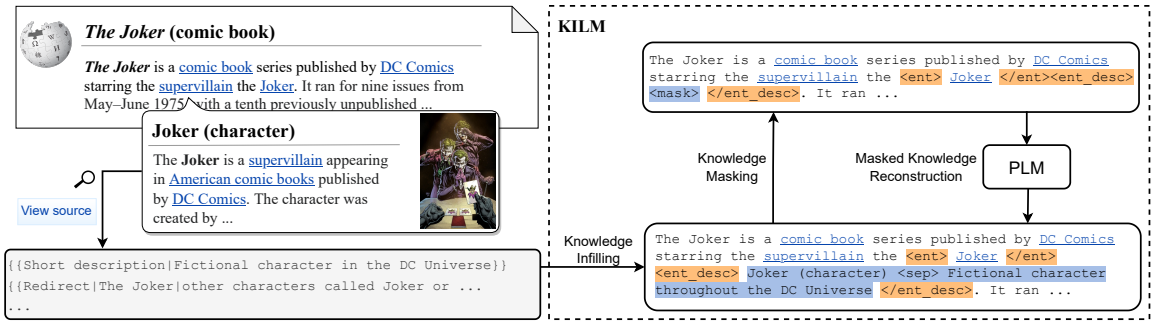

Figure 1: The illustration of the proposed KILM technique for injecting knowledge into PLMs. In the given example, the mention, *Joker*, is linked to the page of Wikipedia entity *Joker (character)*. While the figure only shows knowledge infilling, knowledge masking, and masked knowledge reconstruction steps, the proposed method is combined with the original pre-training objectives of PLMs for continued pre-training.

continued pre-training. With recent work that highlights the need for explicit grounding for PLMs to truly understand text (Merrill et al., 2021), we posit that KILM takes a step in that direction.

The proposed structure for knowledge infilling in KILM is further leveraged as a structured prompt in downstream tasks (see §4.2). We demonstrate better knowledge retention with KILM in zero-shot for entity disambiguation and appositive generation tasks, showing the effectiveness of the proposed method. We also find that BART with KILM outperforms BART on QA tasks and is less prone to hallucination on tasks such as knowledge-grounded response generation. As mentioned earlier, KILM relies on continued pre-training of PLMs, which presents the possibility of catastrophic forgetting of original skills of the PLM. We mitigate this by retaining the original training objectives of BART during the continued pre-training stage. We empirically verify that our proposed objective does not degrade the general language modeling ability of the PLM, nor affect the fluency of these models for natural language generation (NLG) tasks. Although we focus on short descriptions of entities as the knowledge source for KILM, other forms of knowledge can also be used, which we leave for future exploration.

We summarize our contributions as follows:

(1) We propose a novel approach, KILM, to leverage Wikipedia annotations in pre-training of PLMs. We inject knowledge into BART, solely through continued pre-training, with no change in the architecture of the PLMs. KILM enables entity-based knowledge injection with knowledge in natural-language form. KILM's distinct structure also offers a direct way to probe the entity knowledge retained in pre-trained models.

(2) We show that KILM enhances the performance of BART on knowledge-intensive tasks while maintaining its original performance on other downstream tasks. KILM demonstrates improved zero-shot performance on entity disambiguation task, outperforming state-of-the-art models having 30x more parameters.

2 Related Work

Knowledge-Enhanced LMs To enhance PLMs' use of knowledge, a number of work has attempted to augment them with external knowledge sources, such as knowledge graphs (KGs) (Yin et al., 2022). Some recent work introduced additional non-parametric memories into the models (Zhang et al., 2019; Rosset et al., 2020) to obtain entity embeddings and modified the model structures to accommodate extra information (Yamada et al., 2020; Wang et al., 2021a,b), while others changed the masking schema with the additional information (Sun et al., 2019; Wang et al., 2022), or converted the external KGs into natural language text as an additional pre-training corpus (Xiong et al., 2019; Zhou et al., 2020; Liu et al., 2022; Agarwal et al., 2021; Li et al., 2022).

Modeling with Text Linking and Enrichment Our motivation bears similarity to *text linking* (Yasunaga et al., 2022; Deng et al., 2021; Arora et al., 2022) during pre-training and *text enrichment* (Elazar et al., 2022). Modeling the links between documents or metadata is motivated by the fact that PLMs, pre-trained on plain text, are not directly trained to capture inter-dependencies between documents. The similarity between the above tasks and ours lies in the ways humans implicitly *link* information when reading or generat-

ing language. However, the former tasks are restricted to relationships within the text, while our goal is to ground the concepts and entities to their related descriptions in encyclopedic resources.

Pre-training with Hypertext Besides PLMs that are pre-trained with natural language corpora, HTLM (Aghajanyan et al., 2021) directly pre-trains simplified crawled HTML data based on BART models and CM3 (Aghajanyan et al., 2022) extends HTLM into a multimodal setting with causal masked language modeling. The target of HTLM and CM3 is to better leverage the enormous web-scraped data source for pre-training. In contrast, our work aims to leverage hypertext to explore how to inject extra knowledge into PLMs with a custom-designed structure to furnish advantages to PLMs in performing knowledge-intensive tasks.

3 Methodology

Although KILM is model-agnostic and could be used for any PLM (more on this in §5), in this work, due to high computation costs, we focus on applying KILM to BART (Lewis et al., 2020a).

3.1 Preliminaries

Wikipedia is a widely-used text corpus for LM pre-training. It is often processed as a collection of individual articles in the form of flat natural language text. However, due to the existence of hyperlinks in its text, Wikipedia is also a complex web of connected Wikipedia topics, also known as Wikipedia entities. These hyperlinks build connections between different Wikipedia entities and establish a rich source of information that is mostly ignored in current pre-training approaches. Moreover, most Wikipedia articles come with a *short description* of the entity (topic) discussed in the article. These short descriptions provide definitions for Wikipedia entities. In this work, we take an initial step towards using these additional information within Wikipedia articles and utilizing "short descriptions" of entities for continued pre-training of PLMs. Note that the proposed approach could be expanded to ***other annotated text corpora***.

3.2 KILM: Knowledge Injection into Language Models

We propose KILM, which extends the text-infilling objective to knowledge infilling objective through continued pre-training. KILM, as shown in Figure 1, consists of three steps: (1) knowledge infilling, (2) knowledge masking, and (3) masked knowledge reconstruction.

Knowledge Infilling As mentioned in §3.1, in this work, we mainly focus on injecting PLMs with hyperlinks and entity descriptions as the entity-related knowledge into PLMs. Specifically, we process Wikipedia data such that entity mentions in Wikipedia articles (which are annotated by hyperlinks) are marked with a start-of-entity token `<ent>` and an end-of-entity token `</ent>`. Also, each entity mention is followed by an entity-related knowledge sentence marked with `<ent_desc>` and `</ent_desc>` as start- and end-of-description tokens. The inserted knowledge component (highlighted in blue in Figure 1) consists of the corresponding hyperlinked entity (which might be different from the entity's surface form in the text) and the entity's short description connected with the `<sep>` token, where the short description is obtained from a lookup table extracted from the Wikipedia dump. We denote this knowledge infilling transformation as KNINFILL.

Knowledge Masking The processed data is used for the continued pre-training of a PLM. During this step, we conduct knowledge masking transformation (denoted as KNMASK) and the model is trained to reconstruct the whole inserted knowledge component from a single `<mask>` token with respect to the context. More specifically, assuming the ith token t_i is a mention of an entity, the masked input sequence $\mathbf{X}$ and the output sequence $\mathbf{Y}$ can be denoted as:

$$\mathbf{X} = \{t_1, ..., t_{i-1}, \texttt{<ent>}, t_i, \texttt{</ent>, <ent\_desc>}, \texttt{<mask>}, \texttt{</ent\_desc>}, t_{i+1} ..., t_N\},$$

$$\mathbf{Y} = \{t_1, ..., t_{i-1}, \texttt{<ent>}, t_i, \texttt{</ent>, <ent\_desc>}, k_1, ..., k_L, \texttt{</ent\_desc>}, t_{i+1} ..., t_N\},$$

where t_n represents the nth token of the original target sequence and k_l represents the lth token in the knowledge sequence of length L.

Masked Knowledge Reconstruction The parameters θ of the PLM are optimized by a masked knowledge reconstruction loss:

$$\mathcal{L}_{kn} = \mathbb{E}\left(\sum_{l=1}^{L} -\log\left(p\left(k_l | t_{1:(i+l+2)}, \mathbf{X}, \theta\right)\right)\right).$$

Since our goal is to inject entity-related knowledge without disrupting the function of the original BART as a general PLM, the masked knowledge reconstruction loss is combined with the original

Task	Knowledge type	Task adapation	Input/Prompt	Target
Entity Disambiguation	entity	✗	***Context*** D**:** The Big Blue River is ... Driftwood White, <ent> Wabash </ent><ent_desc> <mask> </ent_desc>, and ... ***Candidate*** S^1**:** Wabash River<sep>Tributary of the Ohio ... ***Candidate*** S^2**:** Wabash, Indiana<sep>Wabash is a city in ...	Wabash River
Appositive Generation	entity	✗	The game achieved the highest ... matchup between Larry Bird and Spartans' point guard <ent> Magic Johnson </ent><ent_desc> <mask> </ent_desc>.	a rivalry that lasted throughout their professional careers
In-Context Few-Shot QA	factoid	✗	Question: What jobs did Ben Franklin do? Answer: Diplomat Question: What did Ben Franklin invent? Answer: <mask>	Lightning rod
KGRG	encyclopedia	✓	<speaker2>Ross was an American painter and television host. <speaker1>That's cool. What else? <speaker2>	He created the show "The Joy of Painting"

Table 1: A summary of the knowledge-intensive tasks that are studied in this work. *KGRG* is short for *Knowledge Grounded Response Generation* task. Examples of input and target formats are provided above along with the task information. The definitions of the knowledge types are discussed in the corresponding sections in §4.2.

text infilling objective of BART during continued pre-training.[2] At training time, the model is optimized by minimizing the reconstruction loss over the whole target sequence instead of only the recovered masked spans. As a result, the training objectives force the model to learn to copy the tokens from the input sequences when the token is not a mask token during the pre-training process. This is to help the model recognize the inserted knowledge components in the training sequences and ensure the fluency of the PLM on NLG tasks. The weights of different objectives for loss are calculated based on the proportion of the corresponding spans across the entire sequence. We summarize the proposed KILM algorithm in Appendix B.

The advantages of leveraging this structure for training are two-fold. First, this structure builds an alignment between the entity-related knowledge and the corresponding mention in the paragraphs. Second, the injected knowledge can be easily induced by probing the PLM with the structured prompts proposed for KILM (§4.2).

4 Experiments

We start by exploring the performance of BART+KILM on knowledge-intensive tasks (§4.2). Later, we also demonstrate that KILM does not degrade the original language modeling skills of BART in both NLU and NLG benchmarks (§4.3).

4.1 Pre-training Details

Data To extract the short descriptions and the hyperlinks from Wikipedia articles, we process a Wikipedia dump from scratch.[3] We assign the first sentence of the Wikipedia page as the short description if the "short description" attribute is missing in the raw data. We use the processed data by only leveraging the paragraphs from the summary sections of Wikipedia as our primary training corpus (denoted as *primary setting*), while we also explore a *data upscaling setting* where we use the entire Wikipedia articles. We split the articles with document strides of 512 and consider one snippet as a data sample. We randomly select one entity from the paragraphs in each iteration for dynamic entity-centric knowledge injection.[4] After data preprocessing, we obtain a collection of 5.70 million data samples for the primary setting and 7.85 million data samples for the data upscaling setting from Wikipedia. We split the corpus into a training set and a validation set with around 10k samples, for evaluation. In the following sections, KILM without a subscript indicates that it is conducted under the default primary setting, while KILM under data upscaling setting will be denoted as KILM_{DU}. For pre-training in the primary setting, the model is continually trained for 7,000 steps, and for the data upscaling setting, the model is trained for 50,000 steps.[5] Refer to Appendix C.1 for details.

[2]The comparison between the text infilling and sentence permutation objectives shows the advantage of the former objective over the latter (Lewis et al., 2020a), so we only preserve the text infilling objective for KILM to simplify the continued pre-training task.

[3]The Wikipedia dump is downloaded from https://dumps.wikimedia.org/enwiki/.

[4]We select different entities in each iteration.

[5]Most of our results are based on KILM in the primary setting, and due to the computational resource cost, only for a

Models	AIDA	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	Avg	Parameters
CM3-medium (Aghajanyan et al., 2022)‡	78.0	80.1	75.4	81.4	68.5	76.2	76.6	2,700M
CM3-large (Aghajanyan et al., 2022)‡	80.1	80.8	77.7	82.8	**72.4**	80.2	79.0	13,000M
BART-base	33.8	57.6	44.6	37.8	36.4	46.1	42.7	139M
BART-base+Merge	28.2	43.3	27.1	19.5	27.3	39.9	30.9	139M
BART-base+KILM (ours)	80.0	83.7	74.7	78.2	63.7	71.3	75.3	139M
BART-large	34.4	58.8	42.3	38.9	36.9	46.5	43.0	406M
BART-large+KILM (ours)	84.6	86.4	79.8	80.9	66.1	75.4	78.9	406M
BART-large+KILM$_{\text{DU}}$ (ours)	**86.2**	**87.8**	**84.3**	**83.7**	68.4	**79.9**	**81.7**	406M

Table 2: InKB Micro F1 on zero-shot entity disambiguation tasks with candidates from Le and Titov (2018). ‡The results are from CM3 under the zero-shot setting.

Baselines Besides the original BART, we also report on another BART-base baseline that is continue pre-trained on a merge of Wikipedia corpus and short descriptions for 7,000 steps (same number of steps as KILM) with only text infilling objective. The short descriptions are converted to general text based on the format: **"<Entity> is <Short Desc>"**. This model is denoted as BART-base+Merge. We demonstrate input and output formats of pre-training in Table C6. This baseline is introduced to separately evaluate the role of the distinct structure that is introduced in this work, as well as the additional training steps and data.

4.2 Knowledge-Intensive Tasks

First, we study the effectiveness of KILM on knowledge-intensive tasks (Petroni et al., 2019; Roberts et al., 2020; Petroni et al., 2021). As shown in Table 1, we evaluate BART+KILM on ***entity disambiguation*** and ***appositive generation*** tasks, which have similar objectives to the continued pre-training of KILM. We also evaluate if KILM can contribute to downstream tasks where the pre-training objective of KILM is not fully aligned with those of the downstream tasks. Specifically, We include ***question answering (QA)*** and ***knowledge grounded response generation (KGRG)*** tasks.

Zero-shot Entity Disambiguation The entity disambiguation task requires the model to link a mention q to the correct entity, given a context $\mathbf{D}$ and several candidate entities. ***Without fine-tuning***, we evaluate BART+KILM by picking the candidate with the lowest perplexity of generating short descriptions $\{\mathbf{S}^i\}_{i=1}^N$ using structured prompts among the candidate entities $\{\mathbf{E}^i\}_{i=1}^N$ in entity disambiguation datasets.[6] It can be expressed as:

$$\mathbf{X}_i = \textsc{KnMask}(\textsc{KnInfill}(\mathbf{D}, q, \mathbf{S}^i)) \quad (1)$$

$$E^{i^*} = \arg\max_i \sum_t \log p(s_t^i | \mathbf{X}_i, \theta). \quad (2)$$

We use the same datasets and candidate sets as those in Le and Titov (2018). InKB micro-F1 results are shown in Table 2, where CM3, a series of huge PLMs trained with multimodal hypertext (see §2), are tested in a zero-shot setting. We also included the performances of BART and BART-base+Merge for reference.[7] BART+KILM outperforms CM3-large, which has over 30x more parameters, for half of the datasets. BART+KILM$_{\text{DU}}$ outperforms CM3-large in four out of six datasets. CM3 as a PLM has an impressive performance on entity disambiguation task with no additional training, and this comparison shows that BART+KILM can outperform CM3 with much less parameters. We also present results comparing BART+KILM with BLINK (Wu et al., 2020) in Table C1, where we see that it performs competitively compared to BLINK (which is fine-tuned for entity disambiguation). Moreover, the large gap between the performance of BART+KILM and BART+Merge shows that the proposed distinct structure (and not necessarily the data) plays a key role in the performance of BART+KILM in this task.

Appositive Generation Appositive generation is the task of adding background information for named entities in a sentence in the form of an appositive phrase. As shown in Table 1, we construct structured prompts to probe PLMs without fine-tuning on AppositiveCorpus (Kementchedjhieva et al., 2020). We consider the generated texts recovered from the mask tokens in the short description field as the generated appositives.[8]

subset of knowledge intensive tasks we also report the results for data upscaling setting too.

[6]Note that the reference entities in this task come from Wikipedia, hence we can use the associated entity description for each reference entity.

[7]More details are included in Appendix C.3.

[8]Since the pre-training corpus of BART includes Wikipedia articles, BART can also recover appositives from

Model	News ORG			News PER		
	Ap.	Pref.	NH.	Ap.	Pref.	NH.
BART-base	26.0	17.8	41.7	48.0	14.3	28.3
+KILM	**97.0**	**51.5**	**56.8**	**94.0**	**36.0**	**42.0**
Model	**Wiki ORG**			**Wiki PER**		
	Ap.	Pref.	NH.	Ap.	Pref.	NH.
BART-base	48.5	26.7	49.7	30.8	7.3	32.7
+KILM	**98.0**	**48.0**	**61.0**	**89.9**	**40.3**	**50.3**

Table 3: Human evaluation results on Appositive Generation in News and Wikipedia domains on org- and person-type entities (see Appendix C.7). *Ap.*, *Pref.*, and *NH.* mean *Is Appositive*, *Preference*, and *Not Hallucinated*. Numbers in bold are significantly better than those from BART at p-value of 0.05 in a pairwise t-test.

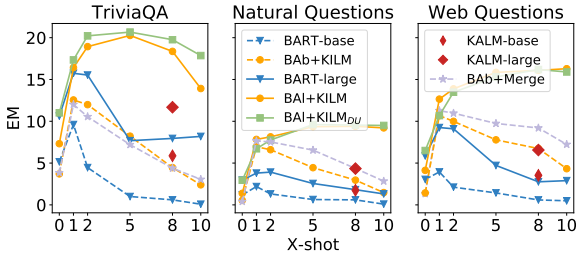

Figure 2: Results on QA datasets with different shots. BART results are in blue, while the results of BART+KILM are in orange and green. We use dashed and solid lines to denote the base- and large-size models, respectively. Also "BAb" and "BAl" correspond to BART-base and BART-large, respectively. KILM$_{DU}$ is KILM with data upscaling where entire Wikipedia articles are used instead of only their first paragraphs.

Since automatic metrics only assess the text overlap based performance (Table C3 in Appendix C.4 with comparisons with SOTA), we conduct human evaluation for a more comprehensive evaluation from three aspects: *Is Appositive* (Ap.), *Preference* (Pref.), and *Not Hallucinated* (NH.). *Ap.* evaluates whether the generation is an appositive or not, while *Pref.* evaluates the suitability of the generated appositives to the context. *NH.* evaluates whether the model generates a hallucinated appositive or not, verifying whether the generated appositive is factually correct. Pairwise A/B testing is utilized to compare the performances of BART before and after KILM (in the primary setting) on all four subsets of ApposCorpus. For each comparison, the same context and two options generated by models for comparison are first randomly shuffled and then are shown to the annotators. Each comparison requires three judgments. 50 data samples are randomly selected from each subset. More details of human evaluation are included in Appendix C.7. Table 3 lists the human evaluation results in terms of the winning rate (ties are counted as wins for both), where we observe that BART+KILM generates better appositives and hallucinates less in all four subsets. These results indicate that BART+KILM possesses more entity-related knowledge than BART.

In-Context Few-Shot QA The implicit knowledge embedded in the parameters can support large PLMs to obtain competitive results on open-domain QA tasks without accessing external knowledge (Roberts et al., 2020; Radford et al., 2019; Brown et al., 2020). We conduct in-context few-shot experiments, in the primary setting of KILM, on TriviaQA (Joshi et al., 2017), Natural Questions (NQ) (Kwiatkowski et al., 2019), and Web Questions (WQ) (Berant et al., 2013) datasets. Similar to the settings of GPT-3 (Brown et al., 2020), we put several example QA pairs into the input sequences of both the encoder and decoder. The format of prompting is shown in Table 1, while the example QA pairs are retrieved with a TF-IDF retriever[9] from the corresponding training set. The tokens recovered from the mask tokens from the decoder will be considered as the generated answers.

We illustrate learning trends with different "shots" in Figure 2 on all three datasets. Interestingly, BART+KILM mostly performs worse than the original BART under the zero-shot setting. However, appending demonstrations into the contexts enables BART+KILM to outperform the original BART by a large margin. With the data upscaling setting, KILM$_{DU}$ provides comparable (or even larger) improvements to BART under the few-shot setting while slightly improving the zero-shot performances of BART. Though far from perfect, these results suggest that KILM significantly improves the in-context learning ability of BART on all three QA datasets. KILM also enables BART to pack factoid knowledge more effectively within its parameters, which supports QA. BART-base+KILM outperforms BART-large under the in-context few-shot setting for the NQ and WQ datasets. The performance of the baseline model, BART+Merge, shows a similar trend to BART+KILM with little advantage on NQ and WQ datasets. This indicates that pre-training with data in "<Entity> is <Short

mask tokens without further task adaptation.

[9]The implementation is based on https://github.com/efficientqa/retrieval-based-baselines.

Model	Seen Test			Unseen Test		
	PPL	R1	R2	PPL	R1	R2
SKT	52.0	19.3	6.8	81.4	16.1	4.2
KAT-TSLF	14.4	21.7	7.6	15.8	20.7	7.2
BART-base	**17.1***	18.7	4.9	**20.9***	17.5	4.0
+Merge	21.4	**19.3**	**5.2**	26.8	**18.0**	**4.2**
+KILM	21.5	**19.3***	**5.2**	26.9	17.9*	**4.2***
BART-large	**14.2***	20.6	5.8	**18.7**	18.5	4.3
+KILM	18.9	**20.8***	**5.9**	24.9	**18.8***	**4.5***

Table 4: WoW test set results. *PPL* denotes perplexity, while *R1/2* denotes ROUGE-1/2 metrics. While both SKT (Kim et al., 2019) and KAT-TSLF (Liu et al., 2021a) use external knowledge as inputs, BART and BART+KILM are evaluated without knowledge to better demonstrate the impact of KILM. $^*p < 0.05$ in a pairwise t-test for comparison between ours and BART.

Model	Seen Test			Unseen Test		
	Flu.	Info.	NH.	Flu.	Info.	NH.
BART-base	59.7	**64.0**	48.4	65.8	**70.3**	46.6
+KILM	**66.7**	63.0	**60.3***	**69.2**	69.3	**58.8**†

Table 5: Human evaluation results on WoW test sets without external knowledge inputs. Flu., Info., and NH. are *Fluency*, *Informativeness* and *Not Hallucinated* respectively. *Model performs significantly better than the baseline ($p < 0.05$); †Pairwise t-test ($p < 0.07$).

Desc>" format is more suitable for QA tasks. Nevertheless, the proposed distinct structure does not bring much obstacle to BART+KILM on QA tasks.

Knowledge Grounded Response Generation (KGRG) The KGRG task requires topical and factual knowledge (Petroni et al., 2021) for a chatbot to make engaging conversations with users on various topics (Ghazvininejad et al., 2018). We fine-tune BART before and after KILM on the Wizard of Wikipedia (WoW) (Dinan et al., 2018) dataset without using knowledge as input, to better study the impact of the injected knowledge under a knowledge-unavailable setting. The generated responses are evaluated with PPL, ROUGE-1 and ROUGE-2 metrics. In Table 4, BART+KILM offers a consistent and significant advantage over BART on ROUGE scores, whereas it underperforms BART on PPL. The performance gap on PPL can be attributed to the fact that many of the responses in WoW contain hallucination (Dziri et al., 2022), which is somewhat mitigated by KILM. Compared to the strong baseline with external knowledge inputs, BART+KILM even performs comparably

Model	**GLUE** Avg.	**CNN** R1	**XSUM** R1
BART-base	83.3	42.79	**40.83***
+KILM	**83.8**	**42.86**	40.76
BART-large	87.1	**44.14***	**45.17**
+KILM	**87.7**	43.15	45.07

Table 6: Results on the GLUE and summarization test sets. We report average score of Matthews correlation for CoLA and accuracy scores for other tasks in GLUE benchmark; and ROUGE-1 for summarization. *pairwise t-test $p < 0.05$.

with SKT (Kim et al., 2019). Note that the performance of BART+Merge shows no difference from BART+KILM, which suggests that the introduced distinct structure does not affect BART's application of injected knowledge on WoW.

While automatic metrics are important in KGRG evaluation, they do not always tell the whole story (Hazarika et al., 2022), therefore we also conduct human evaluation on WoW test sets from three aspects, namely *Fluency (Flu.)*, *Informativeness (Info.)*, and *Not Hallucinated (NH.)*. *Flu.* focuses on whether the responses are fluent and consistent with respect to the conversation so far, while *Info.* evaluates whether the responses contain verifiable factual information. The evaluation on *NH.* is only valid when a response is informative. The settings of human evaluation are the same as those for appositive generation (see Appendix C.7). The results in Table 5 demonstrate that BART+KILM performs comparably with BART in terms of fluency and informativeness, while it tends to **hallucinate less** when generating factual information in the responses, especially in unseen domains.

4.3 General Tasks

We now evaluate the impact of KILM on models' performance on general NLU and NLG tasks using the GLUE benchmark (Wang et al., 2018) and summarization datasets, CNN/Dailymail (Hermann et al., 2015) and XSUM (Narayan et al., 2018), by fine-tuning both BART and BART+KILM for comparison. The summary of the results is shown in Table 6, and the detailed results shown in Table C4 and Table C5. BART+KILM outperforms BART marginally on GLUE and the differences for summarization datasets are small. These results suggest that KILM preserves the performance of the original BART on downstream NLU and NLG tasks, and even in some cases it improves it. They

also verify that KILM does not cause catastrophic forgetting of the original learnings in BART, thus making BART+KILM a reliable PLM.

5 Discussions

Roles of Introduced Special Tokens The introduced special tokens to mark beginning and end of entities (<ent>, </ent>) and entity descriptions (<ent_desc>, </ent_desc>) form a distinct structure in pre-training samples, which inserts entity-centric knowledge into pre-training corpora, thus injects knowledge in PLMs. We discuss the roles of these special tokens from the following aspects:

Entity Knowledge Probing: This distinct structure in KILM provides a tool for probing the entity-related knowledge retained in PLMs. To demonstrate this, we probe BART+KILM by prompting it to generate short descriptions for entities in validation set[10] of the pre-training corpus. The probing format and the corresponding results are shown in Appendix A.1 and Table A1. BART+KILM achieve around 60 unigram F1 scores with no performance gap with the data samples from a subset of the training set. These results indicate that we can easily recall the entity description knowledge in different contexts without sensitivity to prompt designs. It is shown that the proposed pre-training structure is the **main contributor** of the improvements on entity-related datasets, especially in zero-shot manner. By leveraging the introduced special tokens, the knowledge retained in PLMs can be more efficiently leveraged on downstream tasks.

Structured Prompt: The special tokens also provide convenient knowledge probing for zero-shot entity-centric tasks, such as entity disambiguation and appositive generation (§4.2).

Are New Special Tokens Needed? There are a few reasons for introducing new special tokens in KILM for marking entities and their descriptions instead of reusing existing tokens, such as commas or parentheses. First, many entities have commas and parentheses in their names, making the entity descriptions indistinguishable from the contexts. For instance, there are 378,093 entities in English Wikipedia with a comma in their names, such as the entity *"Mars, Aurgazinsky District, Republic of Bashkortostan"*. Second, using commas or parentheses could break the fluency of the text. In a context like "*The Baltic states [...] is used to group three countries: Estonia, Latvia, and Lithuania*", adding a short description for the entity "Estonia" using a comma would break the fluency of the sentence. Finally, using commas or parenthesis will overload their meanings, and during prompting of the model for knowledge probing it will result in a lack of clarity for the model as to how the comma or parenthesis should be interpreted.

Is KILM's impact equal on different domains and tasks? Despite the above-mentioned gains, BART+KILM appears to be less knowledgeable than BART on person-type entities, as manifested in the performance gap between organization- and person-type entities in appositive generation (Table 3). That may be due to the type of knowledge content injected by KILM. The entity knowledge required for generating appositives varies vastly from biographies to relationships with other people. However, short descriptions in Wikipedia for person-type entities focus mostly on their nationality and occupation. Also, many of them are similar [11]. This problem also affects the performance in Table A2 on G-RE datasets in LAMA benchmark. More analyses are in Appendix A.2. We leave the study of enriching the knowledge content for pre-training as future work.

The proposed pre-training structure shows its strength in entity-related tasks. Nevertheless, KILM may downgrade to conventional knowledge-augmented pre-training (BART+Merge) when the pre-training objective of KILM is not fully aligned with those of the downstream tasks.

Placement of Knowledge Component An ablation study on the knowledge component placement in KILM is presented in Appendix A.3, where we show that putting short descriptions right after entity mentions results in better performance compared to placing them at the end of sentences.

Extending KILM for Other PLM Architectures In this paper, we choose BART as the default PLM; however, KILM can also be applied to other PLMs by adjusting their training objectives for knowledge infilling. For decoder-only PLMs, such as GPT-2, the knowledge component, i.e., short descriptions, can be moved to the end of the target sequence (similar to CM3) instead of being adjoined the surface form of the entity. As for encoder-only PLMs, such as BERT, contrastive training strategy introduced in

[10] The articles in validation set are not included in the pre-training process, whereas the involved entities mostly are.

[11] For example short descriptions for both *Columbus Short* and *Drew Fuller* are *"American actor"*

LinkBERT (Yasunaga et al., 2022) is one option for the training objective of KILM. Due to the substantial computational cost of training these models, we leave these explorations for future works.

Justifications on the additional cost during pre-training Injecting additional knowledge text into pre-training corpora may introduce additional costs during the pre-training process. While entity descriptions used in the paper are usually a one-sentence definition of an entity, the average length of short descriptions is 13.81 words. Considering that we split the Wikipedia articles with document strides of 512, the inserted tokens for short descriptions only take 2.6% of the length of the whole sequence, which does not bring much more training cost.

6 Conclusion

In this paper, we propose a novel method, KILM, to inject entity-related knowledge into large PLMs through continued pre-training. Our approach enhances the performance of the original PLMs on knowledge-intensive tasks, especially in zero- and few-shot settings, while not causing catastrophic forgetting of the knowledge in the origianl PLMs. The proposed distinct structure for entity knowledge shows its effectiveness on flexibly probing the injected knowledge in different contexts.

Limitations

In this paper, we propose a continued pre-training method to inject knowledge into large pre-trained language models. There are eight V100 GPUs involved in each pre-training experiment and the whole pre-training process takes 5 days for the base-size model and 13 days for the large-size model, in primary settings. These numbers in data upscaling settings are significantly greater (30 days for the large-size model). Despite its advantage in reducing resource need in inference time, KILM is both time-consuming and computationally resource-consuming during training time.

Similar to any model-based generation system, KILM could be prone to generating factually incorrect statements with regard to entities. These statements might also be prone to be biased based on ethnicity, race, and sexual orientation.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3554–3565.

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. 2022. Cm3: A causal masked multimodal model of the internet. arXiv preprint arXiv:2201.07520.

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. Htlm: Hyper-text pre-training and prompting of language models. arXiv preprint arXiv:2107.06955.

Simran Arora, Sen Wu, Enci Liu, and Christopher Ré. 2022. Metadata shaping: A simple approach for knowledge-enhanced language models. In Findings of the Association for Computational Linguistics: ACL 2022, pages 1733–1745.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pages 65–72.

Brigitte LM Bauer. 2017. Nominal apposition in Indo-European: Its forms and functions, and its evolution in Latin-Romance, volume 303. Walter de Gruyter GmbH & Co KG.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1533–1544.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 1247–1250.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. In International Conference on Learning Representations.

Xiang Deng, Yu Su, Alyssa Lees, You Wu, Cong Yu, and Huan Sun. 2021. Reasonbert: Pre-trained to reason with distant supervision. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6112–6127.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In International Conference on Learning Representations.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar R Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5271–5285.

Yanai Elazar, Victoria Basmov*, Yoav Goldberg, and Reut Tsarfaty. 2022. Text-based np enrichment. Transactions of the Association for Computational Linguistics, 10:764–784.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1. Release date, pages 06–26.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. Semantic Web, 9(4):459–479.

Devamanyu Hazarika, Mahdi Namazifar, and Dilek Hakkani-Tür. 2022. Attention biasing and context augmentation for zero-shot control of encoder-decoder transformers for natural language generation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 10738–10748.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. Bert-mk: Integrating graph contextualized knowledge into pre-trained language models. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2281–2290.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. Advances in neural information processing systems, 28.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In Proceedings of the 2011 conference on empirical methods in natural language processing, pages 782–792.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? Transactions of the Association for Computational Linguistics, 8:423–438.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611.

Jun Seok Kang, Robert Logan, Zewei Chu, Yang Chen, Dheeru Dua, Kevin Gimpel, Sameer Singh, and Niranjan Balasubramanian. 2019. Pomo: Generating entity-specific post-modifiers in context. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 826–838.

Jivat Kaur, Sumit Bhatia, Milan Aggarwal, Rachit Bansal, and Balaji Krishnamurthy. 2022. Lm-core: Language models with contextually relevant external knowledge. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 750–769.

Yova Kementchedjhieva, Di Lu, and Joel Tetreault. 2020. The apposcorpus: A new multilingual, multi-domain dataset for factual appositive generation. In Proceedings of the 28th International Conference on Computational Linguistics, pages 1989–2003.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2019. Sequential latent knowledge selection for knowledge-grounded dialogue. In International Conference on Learning Representations.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. In Proceedings of the 2021 Conference of the North American Chapter of the Association

for Computational Linguistics: Human Language Technologies, pages 4940–4957.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:453–466.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1595–1604.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33:9459–9474.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shuyang Li, Mukund Sridhar, Chandana Satya Prakash, Jin Cao, Wael Hamza, and Julian McAuley. 2022. Instilling type knowledge in language models via multi-task qa. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 594–603.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. Commongen: A constrained text generation challenge for generative commonsense reasoning. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1823–1840.

Linlin Liu, Xin Li, Ruidan He, Lidong Bing, Shafiq Joty, and Luo Si. 2022. Enhancing multilingual language model with massive multilingual knowledge triples. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6878–6890.

Shilei Liu, Xiaofeng Zhao, Bochao Li, Feiliang Ren, Longhui Zhang, and Shujuan Yin. 2021a. A three-stage learning framework for low-resource knowledge-grounded dialogue generation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2262–2272.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S Yu. 2021b. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 6418–6425.

Robert Logan, Nelson F Liu, Matthew E Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5962–5971.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7052–7063.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? Transactions of the Association for Computational Linguistics, 9:1047–1060.

Leora Morgenstern and Charles Ortiz. 2015. The winograd schema challenge: Evaluating progress in commonsense reasoning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 29, pages 4024–4025.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4658–4664.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2523–2544.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. arXiv preprint arXiv:2007.00655.

Robyn Speer, Catherine Havasi, et al. 2012. Representing general relational knowledge in conceptnet 5. In LREC, volume 2012, pages 3679–86.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8968–8975.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. Transactions of the Association for Computational Linguistics, 8:743–758.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355.

Cunxiang Wang, Fuli Luo, Yanyang Li, Runxin Xu, Fei Huang, and Yue Zhang. 2022. On effectively learning of knowledge in continual pre-training. arXiv preprint arXiv:2204.07994.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuan-Jing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1405–1418.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. Kepler: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics, 9:176–194.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6397–6407.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In International Conference on Learning Representations.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6442–6454.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8003–8016.

Da Yin, Li Dong, Hao Cheng, Xiaodong Liu, Kai-Wei Chang, Furu Wei, and Jianfeng Gao. 2022. A survey of knowledge-intensive nlp with pre-trained language models. arXiv preprint arXiv:2202.08772.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1441–1451.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. 2020. Pre-training text-to-text transformers for concept-centric common sense. In International Conference on Learning Representations.

A Analysis

A.1 Entity Description Probing

We analyze the quality of the knowledge injection process by evaluating the model's performance on entity description probing with structured prompts. This task is aligned with our proposed pre-training objective and reflects the effect of the continued pre-training. This can be considered as a plug-and-play process for knowledge induction by simply inserting the proposed distinct structure. We conduct evaluation on the validation set and a subset of the training set with around 10k data samples of our pre-training corpus. The data samples in the training subset are randomly selected, whereas the data samples in the validation set are not included in the training process. More specifically, the entities in the validation set may appear in the training set. However, the contexts of the entities in the paragraphs do not. We demonstrate the structured prompts for entity description probing as follows:

> **Input/Prompt:** `The Joker is a comic book series published by DC Comics starring the supervillain the <ent> Joker </ent><ent_desc> <mask> </ent_desc>.`
> **Target:** `Joker (character) <sep> Fictional character throughout the DC Universe`

The example illustrates the input sequence of the encoder, while the prompt to the decoder is the same until the `<ent_desc>` token (marked with underline). Similar to the decoder-only models, the model is expected to continue generating entity descriptions following the prompt, until the `</ent_desc>` token is generated.

Model	Train subset		Valid	
	EM	F1	EM	F1
BART-base + KILM	37.75	58.08	37.60	58.48
BART-large + KILM	42.58	61.96	42.84	62.69
BART-large + $\text{KILM}^{\dagger}_{\text{End}}$	38.64	57.97	38.59	57.71

Table A1: Results of short description generation on a subset of the training set and the validation set of the pre-training corpus. $^{\dagger}\text{KILM}_{\text{End}}$ is a variant of KILM for ablation study (Appendix A.3).

Model	G-RE	T-REx	C-Net	SQuAD
BERT-base	9.12	30.83	14.29	15.88
ERNIE	6.62	27.58	13.62	14.83
LM-CORE	23.13	55.32	17.28	16.15
KALM-base	3.27	25.96	8.61	6.64
KALM-large	5.41	28.12	10.70	11.89
BART-base	**5.70**	22.14	**13.88**	6.29
+Merge	5.50	**24.98**	13.03	7.69
+KILM	4.02	23.41	12.80	**8.39**
BART-large	**7.76**	26.00	16.07	11.19
+KILM	6.83	**26.14**	**16.96**	11.19
$+\text{KILM}_{\text{DU}}$	3.10	24.99	16.22	**12.94**

Table A2: Accuracy on the LAMA benchmark. The best results are marked with underline, while **Bold** indicates the better result of comparison between BART before and after KILM. The results of previous models except BART are taken from (Zhang et al., 2019; Rosset et al., 2020; Kaur et al., 2022).

The generated entity descriptions are evaluated with exact match (EM) and unigram F1 scores. As the results are shown in Table A1, for KILM in the primary setting, BART models with KILM achieve around 40 EM and 60 F1 scores. Interestingly, there is a marginal performance gap between the seen and unseen validation sets. The results indicate our model not only embed the knowledge with its parameters, but also can recall the injected knowledge under unseen contexts without much performance loss.

A.2 LAMA Knowledge Probing

Petroni et al. (2019) proposed the LAMA benchmark to provide an in-depth study of relational

knowledge in PLMs by probing the answers to "fill-in-the-blank" cloze statements. Different types of relational knowledge are evaluated with statements semi-manually constructed from different knowledge sources, including Google-RE (G-RE), T-REx (Elsahar et al., 2018), ConceptNet (C-Net) (Speer et al., 2012) and SQuAD (Rajpurkar et al., 2016). We follow the original LAMA settings, while only keeping the data samples whose answer length is 1 after tokenization. The probing input and output format of BART and BART+KILM is shown as followings:

Input/Prompt: The Teatr Wielki is a <MASK> .
Target: theatre

Similar to entity description probing in Appendix A.1, *"Input"* and *"Prompt"* (with underline) are inputs to BART encoder and decoder, respectively. The generation is considered to be correct only if it is exactly the same with *"Target"*. We present the probing results in Table A2. We also include the results of BERT (Devlin et al., 2019), BERT-based ERNIE (Zhang et al., 2019), BERT-based LM-CORE (Kaur et al., 2022), and GPT-2-style KALM (Rosset et al., 2020) for reference. However, because of the differences on the tokenization and pre-training process, different PLMs are not comparable on LAMA benchmark (Jiang et al., 2020). Even though KILM does not inject relational knowledge into PLMs, we still observe improvements after KILM on all the datasets except G-RE. As it's discussed in §5, the injected knowledge of person-type entities is not aligned with the knowledge required by G-RE, since the samples from G-RE are focused on `date_of_birth` and `place_of_birth` relations in the *person* domain. Under the data upscaling setting, KILM_{DU} further enhances the rational knowledge required for SQuAD, while LAMA performance is negatively impacted for other datasets. The results indicate that injecting the entity description knowledge also helps models better understand the relationships between specific entities. Moreover, the results of KILM_{DU} suggest that the injected knowledge has closer relevance to the knowledge for SQuAD, whereas far from that of G-RE and T-REx.

A.3 Ablation Study

We conduct an ablation study on the knowledge component position in KILM. We compare our method with KILM variant that moves the knowledge component (highlighted in blue in Figure 1) including <ent_desc> and </ent_desc> to the end of the target sequence. The variant of the target sequence in Figure 1 is as follows:

The Joker is a comic book series published by DC Comics starring the supervillain the <ent> Joker </ent>. It ran for nine ... </s></s> <ent_desc> Joker (character)<sep>Fictional character throughout the DC Universe </ent_desc>

We denote this KILM variant as KILM_{End}. We evaluate these two models on entity description probing and zero-shot entity disambiguation tasks. As shown in Table A1 and Table C1, BART with KILM consistently outperforms BART with KILM_{End} on both tasks. Despite the performance gap, the advantage of KILM_{End} is that KILM_{End} can also be applied to decoder-only models, such as GPT-2, for entity knowledge injection.

A.4 Data Scaling Laws

As mentioned in §4.1, we conduct continued pre-training under two settings: the primary setting and the data upscaling setting. While the primary setting only uses the paragraphs in Wikipedia summary sections, the data upscaling setting extends the training corpus to the whole Wikipedia corpus, which enlarges the training set by more than two million data samples and double the pre-training time. To study the effect of data scaling, we compare the performances of BART-large+KILM under primary and data upscaling settings on knowledge-intensive tasks, including entity disambiguation, LAMA, and closed-book QA tasks. The evaluation on entity disambiguation tasks involves six datasets and we only compare the average InKB F1 scores, since during data scaling, the performances are consistently improved across all the datasets.

In Figure A1, we show the performance difference between BART-large+KILM (or KILM_{DU}) and the corresponding baseline models on entity disambiguation, LAMA (in the first row) and QA (including three datasets under 0/5-shot in the second row) tasks. We also display the performance differences along with each bar, where a positive number denote a better performance of BART+KILM. According to the comparison,

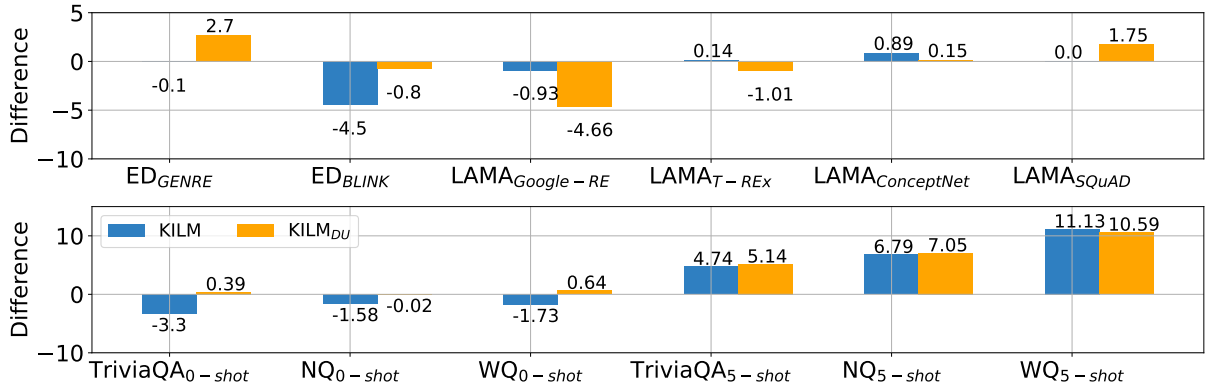

Figure A1: The performance difference between BART-large+KILM (or KILM$_{DU}$) and the corresponding baseline models on entity disambiguation, LAMA and QA (TriviaQA, NQ, and WB) tasks. More specifically, the baseline models of entity disambiguation tasks are CM3-large and BLINK with GENRE and BLINK candidates, while the baseline model of both LAMA and QA tasks is the original BART-large. We also display the performance differences along with each bar, where a positive number denotes a better performance of BART+KILM vs the baseline.

KILM in both settings shows little benefit for Google-RE and T-REx datasets in LAMA benchmark and makes it harder for the model to recall the relational knowledge in specific domains. On the other hand, for the entitiy-based tasks, such as entity disambiguation, the injected knowledge through KILM equip BART with great zero-shot ability, comparing to the strong baseline models, which we've discussed in §4.2. For QA tasks, BART+KILM in the primary setting performs worse than the original BART model in a zero-shot manner, however, BART+KILM in data upscaling setting works comparably with the original BART in this case. Together all these comparisons, we conclude that KILM, as a proposed novel technique for entity-related knowledge injection, is able to largely benefit the model in terms of zero-shot ability on entity-based knowledge-intensive tasks. However, even though we jointly pre-train the model with the original text infilling objective of BART, catastrophic forgetting of some specific knowledge is unavoidable, especially in the data upscaling setting.

A.5 Case Study

Some selected data sample from AppositionCorpus and WoW are shown in Table A3 and Table A4. For zero-shot appositive generation task, while the original BART-base model tends to generate appositives with similar surface forms to the gold ones or a piece of text that fit the context, it hallucinates a lot. BART-base+KILM is more knowledgeable on the actual meaning of the entities, however, it still make mistakes in terms of the date and specific occupation. For KGRG task with task-specific training, both models are able to generate fluent responses. At the same time, BART+KILM tends to hallucinate less by including a bit less information in some cases.

B KILM Algorithm

We denote the data transformations of the text infilling and sentence permutation objectives for BART as TEXTMASK and SENTPERM. In the original pre-training process of BART, given a target sequence with M tokens $\mathbf{Y} = \{t_1, t_2, ..., t_M\}$, and the corresponding corrupted input sequence $\mathbf{X} = \{t'_1, t'_2, ..., t'_N\}$ with N tokens, the model, parameterized by θ, is optimized by minimizing the reconstruction loss over the whole sequence $\mathbf{Y}$:

$$\mathbf{X} = \text{SENTPERM}(\text{TEXTMASK}(\mathbf{Y})) \quad (3)$$

$$\mathcal{L} = \mathbb{E}(\sum_{m=1}^{M} -\log p(t_m | t_{1:m-1}, \mathbf{X}, \theta)). \quad (4)$$

For the proposed KILM continued pre-training, the original document, the selected entity, and the corresponding injected knowledge are represented as $\mathbf{S} = \{t_1, t_2, ..., t_N\}$, E, and $\mathbf{K} = \{k_1, k_2, ..., k_L\}$, respectively. The data transformation procedure can be represented as

$$\mathbf{Y} = \text{KNINFILL}(\mathbf{S}, E, \mathbf{K}), \quad (5)$$

$$\mathbf{X} = \text{KNMASK}(\mathbf{Y}). \quad (6)$$

The final loss can be denoted as:

Algorithm 1: KILM Pre-training Process

Input: Model M_θ, Number of Epochs T, Wikipedia Corpus $\mathbb{S}$, Knowledge Corpus $\mathbb{K}$.

for $i = 1$ **to** T **do**
  for each $S_j \in \mathbb{S}$ **do**
    Sample one entity E_j^i from S_j;
    Retrieve entity knowledge: $\mathbf{K} = \textsc{LookUp}(\mathbb{K}, E_j^i)$;
    Construct training samples: $\mathbf{Y_j^i} = \textsc{KnInfill}(\mathbf{S_j}, E_j^i, \mathbf{K})$, $\mathbf{X_j^i} = \textsc{TextMask}(\textsc{KnMask}(\mathbf{Y_j^i}))$;
    Optimize M_θ with Eq. 7.
  end
end

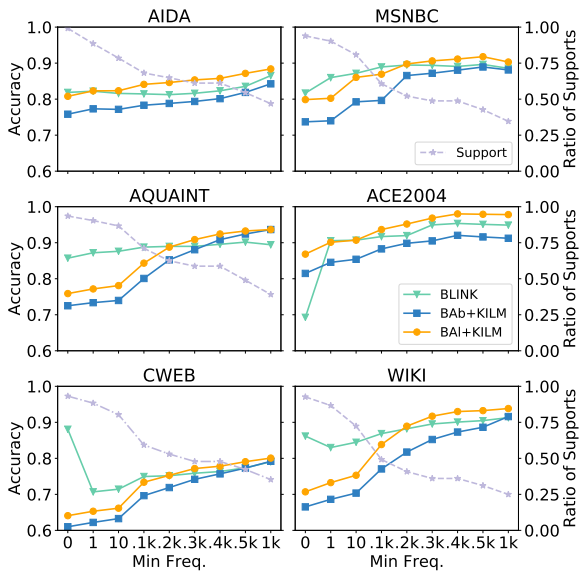

Figure C1: Results of entity disambiguation tasks with the top five candidates and different minimum frequencies at which the target entity is sampled during the continued pre-training. *BAb* and *BAl* denote BART models in base and large sizes. The primary Y-axis shows the performances of the models after KILM and the BLINK baseline on accuracy, while the Y-axis on the right shows the number of data samples that satisfy each setting.

$$\mathcal{L} = (1 - \alpha - \beta)\mathcal{L}_{copy} + \alpha\mathcal{L}_{infill} + \beta\mathcal{L}_{kn}, \quad (7)$$

where α and β are calculated based on the proportion of the corresponding spans across the entire sequence. The resulting KILM algorithm for continual pre-training is summarized in Algorithm 1.

C Additional Details for Experiments

C.1 Pre-training Settings

We initialize the model with the original BART weights and it is continually trained on eight V100 GPUs with a batch size of 8,192. The models are optimized by the Adam optimizer with a linear scheduler and weight decay as 0.01. The peak learning rate is $5e-5$. Moreover, the maximum text length of the sequences with a knowledge component is set as 640. The mask probability and the hyper-parameter λ for Poisson distribution are the same as those of BART. The implementation is mainly based on HuggingFace Transformers (Wolf et al., 2020) and Datasets (Lhoest et al., 2021) packages.

It is worth mentioning that more than 2.3 million entities with short descriptions are involved in the pre-training, and, needless to say, the occurrence of entities in Wikipedia articles is not equally distributed. For instance, while only 2,526 entities appear more than 1,000 times in the primary setting, 40.5% of the entities only appear once in the training corpus.

C.2 Pre-training Format

We use a piece of Wikipedia article to demonstrate the input and output formats of the involved pre-trained models involved in Table C6.

C.3 Zero-shot Entity Disambiguation

As shown in §4.2, we include the performance of BART and BART+Merge for reference. Due to the lack of conventional methods for evaluating BART models on zero-shot entity disambiguation tasks, we are inspired by the entity disambiguation model BLINK (Wu et al., 2020). We evaluate BART and BART+Merge by selecting the lowest perplexity candidate that generates the corresponding Wikipedia summary/short description from a given context. In addition, we also use the same datasets and the candidate sets as those in BLINK for more experiments. The InKB micro-F1 results are shown in Table C1, where BLINK is an entity linking model trained on TACKBP-2010 dataset. BLINK outperforms BART+KILM in the primary setting in all but one of the datasets, but BART+KILM$_{\mathrm{DU}}$ in data upscaling setting largely closes the performance gap between BLINK. It should be noted that both BART+KILM is a general PLM, while BLINK is not.

Models	AIDA	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	Avg	#Params
BLINK†	79.6	**80.0**	**80.3**	82.5	**64.2**	**75.5**	**77.0**	336M
BART-base	18.3	30.8	8.7	20.3	23.7	20.5	20.4	139M
BART-base+Merge	19.5	24.1	12.2	18.4	21.9	19.8	19.3	139M
BART-base+KILM	75.1	69.3	67.8	77.4	57.4	62.2	68.2	139M
BART-large	17.4	39.1	9.6	27.4	26.6	21.5	23.6	406M
BART-large+KILM	80.1	75.2	71.0	82.4	60.0	66.5	72.5	406M
BART-large+KILM$_{DU}$	**82.1**	76.4	77.8	**86.4**	62.4	72.3	76.2	406M
BART-large+KILM$^{‡}_{End}$	79.6	74.5	69.6	82.1	59.2	64.2	71.5	406M

Table C1: InKB Micro F1 on zero-shot entity disambiguation tasks with BLINK candidates. †The results are taken from https://github.com/facebookresearch/BLINK and normalized over the whole dataset. ‡KILM$_{End}$ is a variant of KILM for ablation study (Appendix A.3). *#Params* denotes the number of parameters of the models.

Model	TriviaQA	NQ	WQ
Finetuned settings			
RAG (Open-domain)	68.0	44.5	45.5
T5-base (Closed-Book)	29.1	25.9	27.9
One/Few-shot settings			
KALM-base	5.87	1.75	3.53
BART-base	9.61	2.19	3.94
+KILM	**12.55**	**6.95**	**10.38**
KALM-large	11.68	4.34	6.56
BART-large	15.74	3.80	9.25
+KILM	**16.42**	**7.83**	**12.65**

Table C2: Results on open-domain QA datasets. The best results are marked in bold. The results of the previous models except BART are taken from (Lewis et al., 2020b; Roberts et al., 2020).

Entity Frequency in Pre-training Data To study how the frequency of entities appearing in the pre-training text affects the entity linking performance, Figure C1 also shows the results of experimenting with data samples with different minimum frequencies of sampling the target entity during KILM pre-training in the primary setting. As the minimum frequency increases, the gap between BART+KILM and BLINK reduces.

C.4 Appositive Generation

We conduct zero-shot probing on AppositiveCorpus (Kementchedjhieva et al., 2020). We display the structured prompts of BART with KILM in Table 1. Following AppositiveCorpus, we use unigram F1 and METEOR (Banerjee and Lavie, 2005) for evaluation. The results under constrained and non-empty settings are listed in Table C3. Baseline results for *Person*-type entities in *News* domain come with the original AppositiveCorpus paper, while *AppositiveCorpus$_{constrained}$* denotes that the model is trained only with constrained data samples and *AppositiveCorpus$_{end2end}$* denotes that the model is trained with all the data samples in a specific domain. BART+KILM shows its advantage over BART for the *Organization*-type entities, while BART outperforms BART+KILM on all other entity types. However, as seen in Table 3, the distinction in results between human evaluation and automatic metrics demonstrate how the latter do not capture important dimensions such as hallucinations.

C.5 In-Context Few-Shot QA

In Table C2, we list the QA results when providing one example QA pairs into the inputs (1-shot) to BART models with and without KILM. Aligning with the QA example in Table 1, the general evaluation format is as follows:

```
Question: Example Q Answer: Example A\n
Question: Test Q Answer: <mask>.
```

Besides BART, we also compare our performances with KALM (Rosset et al., 2020) under an 8-shot setting, for which the eight examples are human-written, and two finetuned models with similar model sizes. Despite the performance gap with finetuned models, BART+KILM shows a significant advantage over the original model and KALM on all the datasets, especially for large-size models. The 1-shot results of BART-base+KILM are even higher than those of KALM-large, which has many more trainable parameters.

C.6 Fine-tuning Experiments

For fine-tuning experiments, including GLUE, summarization, and KGRG tasks, we conduct each experiment with random seeds 0, 42, and 852. The

Method	News ORG		News PER		Wiki ORG		Wiki PER	
	F1	METEOR	F1	METEOR	F1	METEOR	F1	METEOR
Constrained setting								
ApposCorpus$^{\dagger}_{\text{constrained}}$	-	-	<u>19.6</u>	7.9	-	-	-	-
ApposCorpus$^{\dagger}_{\text{end2end}}$	-	-	10.8	3.4	-	-	-	-
BART-base	8.4	2.4	**12.1**	**5.6**	5.2	1.9	**9.2**	**4.3**
+KILM	**17.6**	**8.1**	9.7	3.7	**9.7**	**4.4**	8.8	3.7
BART-large	10.8	4.7	**15.9**	**8.3**	8.1	3.9	**13.1**	**7.2**
+KILM	**18.0**	**7.8**	13.9	5.9	**9.5**	**4.5**	9.9	4.4
Non-empty setting								
BART-base	6.6	2.1	**11.7**	**4.8**	4.4	1.6	**7.5**	**3.5**
+KILM	**14.1**	**6.7**	7.2	2.7	**6.7**	**3.1**	5.8	2.5
BART-large	8.7	4.0	**14.9**	**6.7**	**6.8**	**3.2**	**10.6**	**6.0**
+KILM	**14.8**	**6.6**	9.7	4.1	6.6	**3.2**	6.5	3.0

Table C3: Results on zero-shot Appositive Generation under the constrained and non-empty settings. *ORG* and *PER* represent that the data samples are *Person-* and *Organization*-type entities. Bold results denote better performances of one over another with the same settings between BART and BART+KILM. †The results are taken from the original ApposCorpus paper, where *ApposCorpus$_{constrained}$* denotes that the model is trained only with constrained data samples and *ApposCorpus$_{end2end}$* denotes that the model is trained with all the data samples in a specific domain. The result hightlighted with <u>underline</u> denotes that it outperforms both BART and BART+KILM.

Model	**MNLI** m/mm	**SST** Acc	**QQP** Acc	**QNLI** Acc	**STS-B** Acc	**RTE** Acc	**MRPC** Acc	**CoLA** Mcc	**Avg** -
BART-base†	**85.7/85.8**	**93.7**	91.3	**91.6**	**89.9**	74.3	86.4	51.3	83.3
+KILM	**85.7**/85.6	93.0	**91.4**	**91.6**	89.8	**74.9**	**87.8**	**54.2**	**83.8**
BART-large†	**90.0*****/90.0**	**96.4**	92.2	**94.8**	**91.7***	82.3	89.5	57.1	87.1
+KILM	89.5/89.8	96.2	**92.3***	94.7	91.3	**87.0***	**89.6**	**58.7**	**87.7**

Table C4: Results on the GLUE benchmark. We report accuracy for the first seven tasks, the Matthews correlation for the CoLA dataset, and the average score (Avg) over all the tasks. $^{*}p < 0.05$ with pairwise t-test.

Model	**CNN Dailymail**			**XSUM**		
	R1	R2	RL	R1	R2	RL
BART-base†	42.79	**20.31**	39.93	**40.83***	**18.18***	**33.12***
+KILM	**42.86**	20.24	**39.94**	40.76	18.15	33.09
BART-large†	**44.14***	**21.43***	**41.24***	**45.17**	**22.10**	**37.06**
+KILM	43.15	20.86	40.36	45.07	21.93	36.95

Table C5: Results on summarization datasets, evaluating with ROUGE metrics. †The results of the BART models are re-run with the original settings except maximum sequence length to be 1024. $^{*}p < 0.05$ with pairwise t-test.

numbers reported in Table 6, Table C4, Table C5 and Table 4 above are the averages of the results with three random seeds. The results of BART are re-run with the original settings except maximum sequence length to be 1024 for summarization tasks. Pairwise t-tests are conducted to verify the significance level of the results of BART+KILM over the baseline model.

C.7 Human Evaluation

For both appositive generation and KGRG task, we conduct human evaluation for a comprehensive study. Pairwise A/B testing is utilized to compare the performances of BART before and after KILM (in the primary setting). For each comparison, the same context and two options generated by the models for comparison are first randomly shuffled and then are shown to the annotators. Both tasks evaluate the performances on whether the generations are hallucinated or not, named *Not Hallucinated (NH.)*. We also include two more factors for each task. For ApposCorpus, we also evaluate the generated appositives from *Is Appositive (Ap.)* and *Preference (Pref.)*, while we evaluate *Fluency (Flu.)* and *Informativeness (Info.)* for WoW. Because the dialogue task feature, we only consider the *NH.* factor when the generated response is informative for WoW task. Pairwise A/B testing is utilized to compare the performances of BART be-

fore and after KILM on both AppposCorpus and WoW. Human evaluation is done among a group of experts fluent in English coming from countries across Asia. For each comparison, the same context and the generations from both models for comparison are shown to the annotators. The annotators are supposed to choose among "generation A", "generation B", "both", and "neither". Especially for the factor *NH.*, the annotators are asked to search on the Internet for hallucination validation. Each comparison requires three judgments. We randomly sample 50 data samples from each subsets of AppposCorpus and 100 data samples from each WoW test set. Finally, 600 annotations are collected in total for both tasks.

D Datasets

A number of datasets for downstream task evaluation are involved in this work:

GLUE Benchmark GLUE benchmark is a collection of text classification datasets, which is widely used to evaluate the language modeling ability of large PLMs. In this benchmark, nine datasets are involved, including binary QA and NLI tasks. In this paper, we exclude WNLI (Morgenstern and Ortiz, 2015) task during evaluation because there are label conflicts in the dataset.[12]

Summarization Datasets Text summarization is considered an essential NLG task, which requires the model to generate short summaries of long texts. In this paper, we test our models on two summarization datasets, CNN/DailyMail and XSUM. Summaries in the CNN/DailyMail tend to be more extractive, whereas XSUM contains highly abstractive summaries.

Entity Disambiguation Datasets The entity disambiguation task is a subtask of entity linking. Given an entity mention in the context, the model is expected to select the correct entity among a set of similar candidates. Following BLINK (Wu et al., 2020) and GENRE (De Cao et al., 2020), we test our models on six entity disambiguation datasets, including AIDA-CoNLL dataset (Hoffart et al., 2011), MSNBC, AQUAINT, ACE2004, WNED-CWEB (CWEB) (Gabrilovich et al., 2013) and WNED-WIKI (WIKI) (Guo and Barbosa, 2018). We use the candidate sets from BLINK and GENRE respectively, where those of GENRE are originally from Le and Titov (2018).

ApposCorpus Appositives are phrases that appear next to a named entity to provide background information (Bauer, 2017; Kang et al., 2019). They help the readers understand the semantics of the named entities in the context. AppposCorpus (Kementchedjhieva et al., 2020) is constructed as the first end-to-end dataset for the appositive generation task. The selected entities are *Person* and *Organization* entities from *Wikipedia (Wiki)* and *News* articles. Three types of appositives are included: constrained, empty, and a third type denoted as non-empty in this paper. Constrained appositive samples leverage WikiData for appositive generation, while empty appositive samples do not require the model to generate any appositives and non-empty samples require more general knowledge for the appositive generation. In this paper, since we do not conduct task-related training, we only evaluate our models on constrained and non-empty appositive samples.

Open-domain Question Answering Datasets We further evaluate our models on three open-domain QA datasets to test the knowledge capacity: TriviaQA (Joshi et al., 2017), Natural Questions (NQ) (Kwiatkowski et al., 2019), and Web Questions (WQ) (Berant et al., 2013). TriviaQA collects the question-answer pairs from 14 trivia and quiz-league websites, where web pages and Wikipedia articles are matched to each question. NQ is a dataset of questions from web queries that can be answered with a span of Wikipedia articles. While NQ has two types of gold answers, we only evaluate the generations with the short gold answers. WQ consists of questions constructed with web queries and FreeBase (Bollacker et al., 2008)

Wizard of Wikipedia (WoW) dataset WoW is a common crowd-sourcing KGRG dataset that relies on Wikipedia knowledge to augment the dialogue responses when discussing various topics. Two speakers are provided with an initial topic during the data collection to start the conversation. There are two test sets, *seen test* and *unseen test* set, split for evaluation, where the initial topics of the dialogue samples in seen test set appear in the training set and vice versa.

[12] https://gluebenchmark.com/faq

Model	Source	Input/Output Format
BART+KILM (ours)	***Article with Entities:*** The Joker is a comic book series published by [[DC Comics]] starring the supervillain the [[Joker]]. It ran for nine issues from May–June 1975 to Sep.–Oct. 1976. ***Entities & Short Descriptions:*** **DC Comics, Inc.:** American comic book publisher and the flagship unit of DC Entertainment, a subsidiary of Warner Bros. Discovery. **Joker (character):** fictional character throughout the DC Universe.	***Sample 1*** **Input:** The Joker <mask>book series published by </ent> DC Comics </ent><ent_desc><mask></ent_desc>starring the <mask>the Joker. It ran for nine issues from May–June 1975 to Sep <mask>. **Output:** The Joker is a comic book series published by DC Comics</ent><ent_desc>DC Comics, Inc. <sep> American comic book publisher and the flagship unit of DC Entertainment, a subsidiary of Warner Bros. Discovery. </ent_desc>. starring the supervillain the Joker It ran for nine issues from May–June 1975 to Sep.–Oct. 1976. ***Sample 2*** **Input:** The Joker is a comic <mask>by DC Comics starring <mask>supervillain the <ent>Joker </ent><ent_desc> <mask></ent_desc>. It ran for nine issues from May <mask> Sep. – Oct. 1976. **Output:** The Joker is a comic book series published by DC Comics starring the supervillain the <ent>Joker </ent> <ent_desc>Joker (character) <sep>fictional character throughout the DC Universe </ent_desc>. It ran for nine issues from May–June 1975 to Sep.–Oct. 1976.
BART+Merge (baseline)		***Sample 1*** **Input:** The Joker <mask>book series published by DC Comics starring the <mask>the Joker. It ran for nine issues from May–June 1975 to Sep <mask>. **Output:** The Joker is a comic book series published by DC Comics. starring the supervillain the Joker. It ran for nine issues from May–June 1975 to Sep.–Oct. 1976. ***Sample 2*** **Input:** DC Comics, Inc. is American <mask>and the flagship unit of DC <mask>, a subsidiary of <mask>Discovery. **Output:** DC Comics, Inc. is American comic book publisher and the flagship unit of DC Entertainment, a subsidiary of Warner Bros. Discovery. ***Sample 3*** **Input:** Joker <mask>fictional character <mask>Universe. **Output:** Joker (character) is fictional character throughout the DC Universe.
Original BART		**Input:** It ran for nine issues from May <mask>Sep. – Oct. 1976. The Joker is a comic <mask>by DC Comics starring <mask>supervillain the Joker. **Output:** The Joker is a comic book series published by DC Comics starring the supervillain the Joker. It ran for nine issues from May–June 1975 to Sep.–Oct. 1976.

Table C6: Demonstrations of input and output formats of the pre-trained models involved in this work. "BART+KILM" denotes the models that are continued pre-trained with our proposed method; "BART+Merge" denotes the situation when BART model is continued pre-trained on a merge of Wikipedia corpus and the entity short descriptions; "BART" row shows the input and output formats of the original pre-training process of BART models.

Target Entity:	Burma Socialist Programme Party
Context:	He related the quality decline to the socialist system being practiced by Burma Socialist Programme Party <appos> at the time. As cassette players became popular, people were selling their old gramophones to dealers.
Appostive: Gold BART-base BART-base+KILM	 BSPP PSP founded in 1948 by Ne Win, was one of the two major political parties in Burma during the 1980–1990s

Target Entity:	Sultana Kamal
Context:	On April 27 , 2013 an alliance of 68 non-governmental organisations working for women 's rights held a rally in Dhaka to speak out against the proposed subjugation of women. Sultana Kamal <appos> told reporters there: The people are non-communal and religious tolerant.
Appostive: Gold BART-base BART-base+KILM	 a human rights activist the head of the alliance Bangladesh politician

Table A3: Selected data samples from ApposCorpus, along with generated appositives from both BART-base and BART-base+KILM. <appos> marks the position where an appositive is needed to enhance the target entity. In the above two tables, text in orange is not factual correct, while text in blue is not fully accurate.

Dialogue History:	B: Skiers used to only use one pole or spear instead of two. Have you every gone skiing? A: I haven't ever gone skiing, but my kids do and love it. Maybe one of these days I'll try it out. Do you ski?
Response: BART-base BART-base+KILM	 B: I do, but I don't know if I can do it. I do know that skiing is the most popular sport in the world . B: I do, but I don't ski much. I'm not a big fan of snow skiing.

Dialogue History:	A: Nice, I walk my dog at least once a day. B: I like watching the dog on "The Simpsons". He is funny along with the family and friends. A: I love The Simpsons.
Response: BART-base BART-base+KILM	 B: The Simpsons is a great show. It was created by Homer Simpson and Matt Groening. B: I love the Simpsons too. It was created by Matt Groening.

Table A4: Selected data samples from WoW, along with generated responses from both BART-base and BART-base+KILM. The highlighted generations in pink is annotated to contain verifiable information.

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.