

LM-CORE: Language Models with Contextually Relevant External Knowledge

Jivat Neet Kaur^{*α}, Sumit Bhatia^β, Milan Aggarwal^β,
Rachit Bansal^{*γ}, and Balaji Krishnamurthy^β

^α Microsoft Research, India

^β Media and Data Science Research Lab, Adobe, India

^γ Delhi Technological University, India

t-kaurjivat@microsoft.com, sumit.bhatia@adobe.com
milaggar@adobe.com, racbansa@gmail.com, kbalaji@adobe.com

Abstract

Large transformer-based pre-trained language models have achieved impressive performance on a variety of knowledge-intensive tasks and can capture factual knowledge in their parameters. We argue that storing large amounts of knowledge in the model parameters is sub-optimal given the ever-growing amounts of knowledge and resource requirements. We posit that a more efficient alternative is to provide explicit access to contextually relevant structured knowledge to the model and train it to use that knowledge. We present LM-CORE – a general framework to achieve this– that allows *decoupling* of the language model training from the external knowledge source and allows the latter to be updated without affecting the already trained model. Experimental results show that LM-CORE, having access to external knowledge, achieves significant and robust out-performance over state-of-the-art knowledge-enhanced language models on knowledge probing tasks; can effectively handle knowledge updates; and performs well on two downstream tasks. We also present a thorough error analysis highlighting the successes and failures of LM-CORE. Our code and model checkpoints are publicly available¹.

1 Introduction

Large pre-trained language models (PLMs) (Peters et al., 2018; Devlin et al., 2019; Brown et al., 2020) have achieved state-of-the-art performance on a variety of NLP tasks. Much of this success can be attributed to the significant semantic and syntactic information captured in the contextual representations learned by PLMs. In addition to applications requiring linguistic knowledge, PLMs have also

been useful for a variety of tasks involving factual knowledge and it has been shown that models such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) store significant world knowledge in their parameters (Petroni et al., 2019).

PLMs are typically fed a large amount of unstructured text which leads to the linguistic nuances and world knowledge being captured in the model parameters. This *implicit storage* of the knowledge in the form of the parameter weights not only leads to poor interpretability while analyzing model predictions but also poses constraints on the amount of knowledge that can be stored. It is not practical to pack all the ever-evolving world knowledge in the language model parameters due to the great financial and environmental costs incurred by training of the PLMs. Further, since the PLMs acquire knowledge from the text corpora they are trained on, they tend to become sensitive to the contextual and linguistic variations (Jiang et al., 2020). Moreover, PLMs do not contain explicit grounding to real world entities, and hence, often find it difficult to *recall* factual knowledge (Logan et al., 2019). For example, the model may not be able to *recall* correct information and successfully complete the sentence, “*The birthplace of Barack Obama is ___*”, if the LM has seen this fact in a different context during training (e.g., “*Barack Obama was born in Honolulu, Hawaii.*”).

Large scale structured knowledge bases (KBs) such as YAGO (Suchanek et al., 2007) and Wikidata (Vrandečić and Krötzsch, 2014) offer a rich resource of high quality structured knowledge that can provide the PLMs with explicit grounding to real world entities. Consequently, efforts have been made to integrate factual knowledge into PLMs and create entity-enhanced language models (Peters et al., 2019; Zhang et al., 2019; Sun et al., 2020; Liu et al., 2020; Wang et al., 2021a,b). However,

^{*}Work done as an intern at the Media and Data Science Research Lab, Adobe, India.

¹<https://github.com/sumit-research/lmcore>

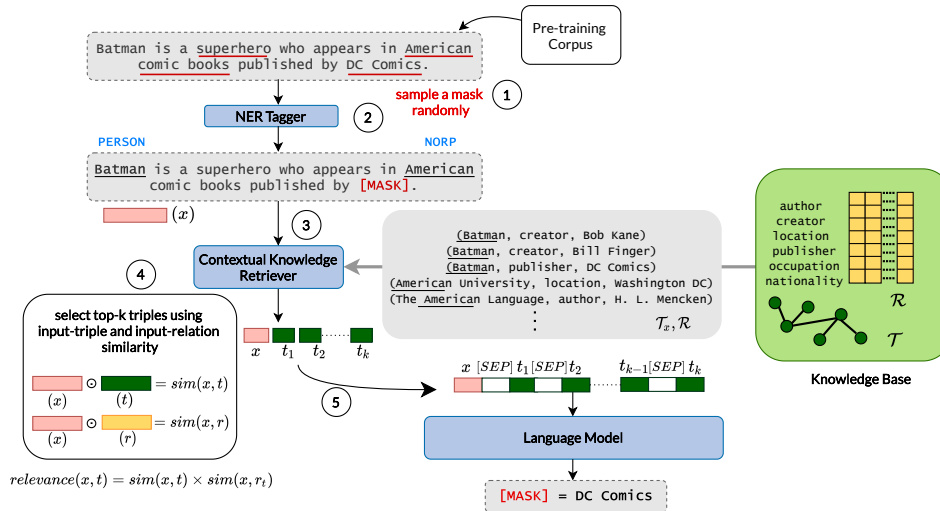


Figure 1: **Language Model Pre-Training with Contextually Relevant External Knowledge:** ① Using a sentence sampled from the pre-training corpus, an input (x) is created by selecting an entity mention at random from the potential mask candidates (underlined in red). ② An NER tagger is then applied to the masked input sequence (x) to identify named entities (underlined in black). ③ For the identified entities, the Knowledge Retrieval module fetches the set \mathcal{T}_x of all the triples from the Knowledge Base and then ④ scores all the retrieved triples using input-triple and input-relation similarity (details in Section 3.2). ⑤ The top- k triples are fed to the Language Model encoder along with the input sequence (x) and the model is trained to predict the masked token.

these works either update the PLM parameters or modify the architecture to facilitate the storage of factual knowledge in the model layers and parameters, making it expensive to update knowledge.

In this work, we step back and ask – *what if instead of focusing on storing the knowledge in the language model parameters, we provide the model with contextually relevant external knowledge and train it to use this knowledge?* This approach offers several potential advantages – (i) we can utilize the already available high-quality large-scale knowledge bases such as YAGO and Wikidata; (ii) not all the knowledge needs to be packed in the parameters of the model resulting in lighter, smaller and greener models; and (iii) as new knowledge becomes available, the knowledge base can be updated independently of the language model.

Our Contributions: We present LM-CORE, a framework for augmenting language models with contextually relevant external knowledge. The LM-CORE framework is summarized in Figure 1 and consists of a contextual knowledge retriever that fetches relevant knowledge from an external KB and passes it to the language model along with the input text. The language model is then trained with a modified entity-oriented masked language modeling objective (Section 3). Our proposed solution is simple, yet highly effective. Experiments on benchmark knowledge probes show that the pro-

posed approach leads to significant performance improvements over base language models as well as state-of-the-art knowledge enhanced variants of the language models (Section 4). We find that with access to contextually relevant external knowledge, LM-CORE is less sensitive to the contextual variations in input text. We also show how LM-CORE can handle knowledge updates without any re-training and compare the performance of LM-CORE on two knowledge-intensive downstream tasks. Finally, we present an in-depth analysis of cases where our proposed approach gives incorrect answers paving the way for further research in this direction (Section 4.7).

2 Related Work

Augmenting Additional Knowledge in PLMs: Previous works on augmenting PLMs with additional knowledge can be grouped into two categories. One line of work adopts a retrieve and read framework where the model is trained to retrieve relevant information followed by a reading comprehension step to perform the downstream task (Lee et al., 2019a; Guu et al., 2020; Agarwal et al., 2021). While our proposal has similarities with this line of work in terms of *retrieving* the contextual knowledge, there are two major differences. First, most of these works consider external knowledge in the form of unstructured text (such as

Wikipedia documents). However, extracting factual knowledge from unstructured text is hard and error-prone due to ambiguities in natural language and infrequent mentions of entities of interest (Peters et al., 2019). This issue can be alleviated by using a structured knowledge base where the knowledge is represented (mostly) unambiguously – each fact is a triple in the knowledge base. Further, these approaches employ *explicit supervision during pre-training* to train the model to fetch relevant passages from the text. This results in systems that are more complex and resource-hungry than the base PLMs used and also make it difficult to reuse or adapt the models to different sources of knowledge.

The second body of work has focused on injecting the factual knowledge directly into the model parameters by feeding more data to the model during pre-training (Poerner et al., 2020; Roberts et al., 2020). A promising direction explored recently is utilizing structured knowledge bases to augment Transformer-based LMs. ERNIE (Zhang et al., 2019) and KnowBERT (Peters et al., 2019) are notable efforts in this direction where the entity information from knowledge bases is explicitly linked with the input text during pre-training yielding entity-enhanced variants of BERT models with entity representations integrated within the Transformer layers. An alternative way of training entity-aware language models is illustrated by frameworks such as CoLAKE (Sun et al., 2020) and KEPLER (Wang et al., 2021b) that jointly learn the language and knowledge representations thereby producing language models augmented with factual knowledge and knowledge embeddings enhanced with textual context. However, these approaches, by design, will lead to larger and larger models to store the ever-growing abundant knowledge. Further, due to the strong coupling between the knowledge and language signals, updating or adding knowledge requires re-training of the model.

Examining the knowledge contained in PLMs: Petroni et al. (2019) posit that while training over large amounts of input text, PLMs may also be storing (implicit) relational knowledge in their parameters and proposed the Language Model Analysis (LAMA) framework to measure the relational knowledge stored in a PLM. Jiang et al. (2020) argue that due to the sensitivity of the PLMs on the input context, such manually created prompts are sub-optimal and might fail to retrieve facts that the PLM *does know*, thus providing only a lower bound

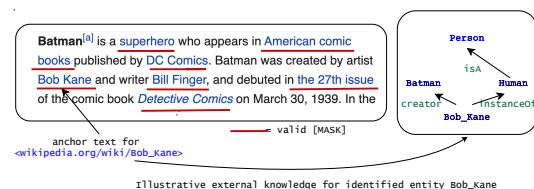


Figure 2: We create our pre-training corpus from Wikipedia by masking entity spans detected using anchor text of hyperlinks.

estimate of the knowledge contained in it. Subsequent work (Shin et al., 2020; Zhong et al., 2021) has attempted to generate better prompts in order to tighten this estimate. Poerner et al. (2020) introduced LAMA-UHN (UnHelpfulNames), a much harder subset of LAMA where the input probes provide little or no helpful contextual signals from other tokens in the probe, thus measuring the innate ability of the PLM to recall information.

3 LM-CORE: Knowledge Retrieval and Training Framework

Task setting and Overview: Consider a language model \mathcal{L} (such as BERT and RoBERTa) and a knowledge base $\mathcal{K} = \{t_{hrt} = \langle h, r, t \rangle \mid h, t \in \mathcal{E}; r \in \mathcal{R}\}$. Here, we consider the knowledge base \mathcal{K} as a set of triples such that each triple t_{hrt} represents the relationship r between entities h and t . \mathcal{E} is the set of all the entities, and \mathcal{R} is the set of all the relationship types present in the knowledge base. Given a text input x , the proposed LM-CORE framework retrieves a set of triples $\mathcal{T}_x \in \mathcal{K}$ such that the triples in \mathcal{T}_x are contextually relevant to x . The language model is then presented with the original input x and the contextually relevant knowledge in the form of \mathcal{T}_x and is trained to make predictions using this additional knowledge. We posit that the model essentially needs to learn relevant semantic associations between natural language input text and various relation types present in the knowledge base. Identifying the correct relation types will help the model leverage the corresponding relevant facts in order to make an accurate prediction. This is accomplished via a modified Masked Language Modeling (MLM) (Devlin et al., 2019) pre-training objective. Figure 1 summarizes the complete workflow of our proposed LM-CORE framework and we describe the three main components in detail in the following sub-sections.

3.1 Entity span masking

Masked Language Modelling is a popular task used for training PLMs where the objective is to predict the masked token in the input sequence. In order to improve model’s grounding to real world entities, previous works have adopted different strategies for explicitly masking entity information in the input text by using entity representations obtained by knowledge base embeddings (Zhang et al., 2019), using named entity recognizers (NER) and regular expressions (Guu et al., 2020), and verbalizing knowledge base triples (Agarwal et al., 2021). These approaches often result in noisy masks due to the limitations of underlying rules, and NER and entity linking systems. To overcome these limitations, we propose a novel way of creating high-quality and accurate entity masks by using Wikipedia as the base corpus for training. Note that in order to create entity masks, we need to identify corresponding entity mentions in the input text for which we utilize the human-annotated links in Wikipedia. The official style guidelines of Wikipedia require the editors to link mentions of topics to their corresponding Wikipedia pages. In Figure 2, the left textbox shows a screenshot of the Wikipedia article about Batman where various other related topics, or concepts, are linked to their corresponding Wikipedia pages (underlined in red in the figure, and displayed as blue anchor-text in Wikipedia). This information provides us with high-quality human annotation of entity mentions in the input text. As illustrated in Figure 2, the underlined tokens (such as DC Comics, Bob Kane, Bill Finger) constitute the set of entity tokens that could be masked. For each such mask, we can also obtain the corresponding contextual knowledge from the external knowledge base (illustrated for Bob Kane in the right text box). By masking only the entity tokens (instead of randomly sampled words) and providing contextually relevant knowledge to the model retrieved from the knowledge base (as described in next subsection), we expect the model to learn to predict the masked entity tokens by utilizing the external knowledge.

3.2 Contextual Knowledge Retrieval

After preparing the masked input for training, the second component in our framework fetches contextually relevant knowledge to feed to the language model. Consider the sentence, “Warren Buffett is the chairman of [MASK]”, where the masked

token is *Berkshire Hathaway*. In the typical MLM setting, the model only has access to the linguistic and contextual clues present in the input text to predict the masked token. However, if contextually relevant information is available as additional input, the model can use it to output the correct token.

We consider the problem of finding contextually relevant facts given the input query text as an information retrieval (IR) problem and adopt a retrieve and re-rank approach that has empirically been found to perform well in a variety of tasks (Chen et al., 2017; Wang et al., 2017; Das et al., 2019; Yang et al., 2019). Recall the example input discussed above – “Warren Buffett is the chairman of [MASK]”. Intuitively, in this input text, there are two important signals that the retriever needs to utilize – *entity* and *relation* information. The entity mention *Warren Buffett* indicates that we need to fetch facts related to Warren Buffet from the knowledge base. Typically, there are numerous facts related to a given entity in the knowledge base, especially for popular entities such as Warren Buffett. Thus, the retriever also needs to utilize the presence of the word chairman to retrieve facts (KB triples) representing the management or executive relation.

Given an input text, our retriever pipeline performs Named Entity Recognition (NER) to identify named entity mentions in the input text. We use the NER model from FLAIR (Akbik et al., 2019) to identify named entity mentions and then select KB entities having maximum overlap with the mention-span of the identified entities. For instance, if the input query is “Buffett was born in [MASK]”, all of the entities containing Buffett - Warren_Buffett, Howard_Warren_Buffett, Howard_Graham_Buffett, Volcano_(Jimmy_Buffett_song) etc. are selected, but if the query is “Warren Buffett was born in [MASK]”, only the first two entities will be chosen. Once these entities are selected, all the facts from the KB involving these entities are retrieved (denoted by \mathcal{T}_x in Figure 1).

After retrieving the facts involving the entities mentioned in the input, we next need to rank these triples based on their relevance to the input. In order to measure the contextual relevance of a given triple t to the input x , we compute the following two scores.

Query-Triple similarity: We obtain representations of the input text x as well as the triple t and compute the inner product of the representations to

obtain the similarity score as follows.

$$\text{sim}(x, t) = \text{Emb}(x)^T \text{Emb}(t), t \in \mathcal{T}_x \quad (1)$$

Here, $\text{Emb}(\cdot)$ is obtained using the Sentence Transformer (Reimers and Gurevych, 2019). While it is straightforward to obtain representations of input x , sentence transformer can not be applied directly to KB triples. Application of KB embeddings such as TransE (Bordes et al., 2013) is also not feasible as then the representations of the input text and triples will be in different embedding spaces. To overcome this, we adopt a simple approach of verbalizing the knowledge base triples by concatenating the head entity, relationship and the tail entity, and obtain the representation of the verbalized triple from the sentence transformer. For example, the triple (Warren_Buffett, hasOccupation, Investor) is verbalized as *Warren Buffett has occupation Investor* and is fed as input to the sentence transformer.

Relation-based scoring: A triple is highly relevant for the input text if the triple represents the same relationship that is being talked about in the text. To capture this intuition, we embed all the relation types in the KB in the same embedding space as triples using the sentence transformer and compute the similarity between the input text and the relation type of the triple as follows.

$$\text{sim}(x, r) = \text{Emb}(x)^T \text{Emb}(r), r \in \mathcal{R} \quad (2)$$

where \mathcal{R} is the set of all relations in the KB. The final relevance score for the triple t , $\text{relevance}(x, t)$ is obtained by taking a product of the above two scores. Based on this final score, we select the top- k triples that constitute the contextual knowledge to be fed as input along with x to the LM. We use $k = 8$ in this work (See Appendix 4.3 for effect of varying k). Some illustrative examples of the final retrieved knowledge base triples are presented in Appendix A.3.

3.3 Language Model Pre-training with Contextual Knowledge

With the masked training corpus and the module to fetch contextually relevant knowledge, we now train the model to utilize the additional contextual knowledge to predict the masked token. From the masked corpus, we select a sentence and a valid entity span is chosen at random out of all the potential spans in the sentence. We mask this span to create the input text x . We filter out sentences starting with pronouns such as *he*, *she*, *her*, and

they as we observed that most of such sentences do not contain other useful signals to unambiguously predict the masked words. For instance, if the input example is - "*He developed an interest in investing in his youth, eventually entering the Wharton School of the University of Pennsylvania*" and *Wharton School of the University of Pennsylvania* is masked, the remaining words in the sentence are not providing any informative signals to the model to predict the masked tokens. Given the input sentence thus selected, the contextual knowledge retriever fetches the relevant triples from the knowledge base. The representations of the input sentence and the retrieved triples are then concatenated and fed to the model and the model is trained to minimize the following MLM loss.

$$L_{MLM} = \frac{1}{M} \sum_{m=1}^M \log p(x_{ind_m} | x, t_1, t_2, \dots, t_k) \quad (3)$$

where M is the total number of [MASK] tokens in x and ind_m is the index of the m^{th} masked token.

With the additional contextual information available to the model, we expect the model to learn the associations between linguistic cues in the input text and relevant relationship information in the triples. For example, we expect the model to associate different ways in which someone’s date of birth could be mentioned in natural language (such as *X was born on*, *the birthday of X is*, and numerous other linguistic variations) to the KB relation `birthDate` and utilize the information from the corresponding triple. Note that since the types of relations in the knowledge base are relatively small in number, and do not change often, we expect the model to generalize well and be more robust to linguistic variations.

4 Experiments and Discussions

Data Sources and Pre-processing: We create our pre-training corpus using the December 20, 2018 snapshot of English Wikipedia that contains about 5.5M documents. Processing all the articles following the masking strategy described in Section 3.1 resulted in a total of ~ 46.3 M sentences with valid masks, from which we randomly sample sentences to create input examples.

In order to illustrate the general nature of LM-CORE, we used two different PLMs as our LM encoders – BERT-base (uncased) model and RoBERTa-base (cased) model. We use YAGO and

Table 1: Mean precision at one (P@1) of various models on LAMA probe. We group all the models based on the base language model used (BERT or RoBERTa). For LM-CORE, (\cdot, \cdot) indicates the variant – (b, r corresponds to BERT and RoBERTa, respectively, and y, w indicate YAGO and Wikidata5M, respectively). Best results in each column are highlighted in bold and the second best performance is underlined

	Complete	Google-RE				T-REx				SQuAD	Concept Net
		DoB	PoB	PoD	All	1-1	N-1	N-M	All		
<i>BERT-based models</i>											
BERT-base	24.73	<u>1.59</u>	15.46	10.33	9.12	67.94	32.67	23.54	30.83	14.29	15.88
BERT-large	25.44	<u>1.59</u>	15.53	12.16	9.76	74.23	31.30	25.30	31.05	17.61	18.72
ERNIE	22.16	1.42	13.48	4.97	6.62	61.51	28.57	21.93	27.58	13.62	14.83
LM-CORE(b,y)	<u>39.64</u>	64.44	52.71	50.98	56.04	<u>74.37</u>	<u>51.18</u>	<u>34.57</u>	<u>45.83</u>	15.61	14.78
LM-CORE(b,w)	42.83	0.66	<u>37.62</u>	<u>31.11</u>	<u>23.13</u>	81.79	59.86	45.48	55.32	<u>17.28</u>	<u>16.15</u>
<i>RoBERTa-based models</i>											
RoBERTa	20.46	<u>1.85</u>	12.98	1.23	5.35	57.49	23.14	21.59	24.21	12.94	<u>18.47</u>
RoBERTa-large	24.24	1.41	12.48	0.46	4.78	70.24	29.08	23.28	28.82	18.88	22.09
KEPLER	19.36	1.47	11.73	3.08	5.43	52.32	21.58	21.41	23.01	9.10	17.25
CoLAKE	23.38	1.79	15.72	10.79	9.43	64.08	29.40	23.54	28.80	8.39	17.17
LM-CORE(r,y)	<u>34.60</u>	46.33	43.47	<u>26.35</u>	38.71	<u>68.21</u>	<u>45.30</u>	<u>30.40</u>	<u>40.60</u>	13.29	17.53
LM-CORE(r,w)	41.96	0.38	<u>33.11</u>	28.20	<u>20.56</u>	70.21	60.30	43.18	54.11	<u>15.73</u>	18.38

Wikidata as two different knowledge bases giving us four variants of LM-CORE ($\{bert, roberta\} \times \{yago, wikidata\}$). We use the English Wikipedia version of YAGO 4 (Suchanek et al., 2007) and preprocess it to obtain our retrieval corpus consisting of roughly 17M triples spanning over 4.9M entities and 131 unique relations. For Wikidata, we used the Wikidata5M version (Wang et al., 2021b) that consists of roughly 21M triples covering 821 unique relations and 4.8M entities. Further details regarding retrieval corpus generation and processing can be found in the Appendix (Section A.2). For computing triple representations for retrieval (Section 3.2), we concatenate the subject (head), relation, and object (tail) of triples and embed them using the Sentence Transformers (Reimers and Gurevych, 2019) and obtain the 768-dimensional embeddings (same as LM encoder dimensions).

4.1 Does External Knowledge Help PLMs in Knowledge Intensive Tasks?

We now present an analysis of how much, and if, having access to external knowledge can help PLMs in knowledge-intensive tasks. A popular way of assessing a model’s ability to perform at such tasks is by using benchmark knowledge probes. We use the LAMA probe (Petroni et al., 2019) that provides a cloze-style sentence representation of facts and the model being evaluated is required to predict the masked words in these sentences (e.g., *Barack Obama was born in ____*).

Table 1 reports the performance of various PLMs

on the LAMA probe as measured by Precision at 1 (P@1). The numbers in the Table are grouped based on the base language model used by different models. We use ERNIE (Zhang et al., 2019) (based on BERT and Wikidata), and KEPLER (Wang et al., 2021b) and CoLAKE (Sun et al., 2020) (based on RoBERTa) as the representative knowledge enhanced language models. Both KEPLER and CoLAKE have used Wikidata5M as the knowledge base. We used author provided code and checkpoints for obtaining the reported numbers. For LM-CORE, we use four variants with different knowledge base and language encoder combinations as described above.

We observe that our approach of providing external knowledge to the PLMs leads to substantially improved performance over the base language models and their SoTA knowledge enhanced variants. LM-CORE(b,w) achieves P@1 of 42.83% compared to 25.44% for BERT-large. Likewise, LM-CORE(r,w) achieves a P@1 of 41.96% significantly outperforming RoBERTa-large (24.24%). We also report the numbers on the four different subsets of LAMA revealing interesting insights. For all the models considered, we note that the performance on T-REx subset is higher than the Google-RE subset. We attribute this to the nature of knowledge required for probes in the four subsets. Note especially the column for Date of Birth (DoB) in the Table. All the models, except for LM-CORE(b,y) and LM-CORE(r,y) perform extremely poorly. This is because the Wikidata5M KB does

Table 2: P@1 for different models on LAMA-UHN.

	LAMA	LAMA UHN	Percentage Change
<i>BERT-based models</i>			
BERT-base	24.73	18.72	-24.31
BERT-large	25.44	19.92	-21.67
ERNIE	22.16	15.81	-28.66
LM-CORE(b,y)	39.64	<u>41.33</u>	+4.26
LM-CORE(b,w)	42.83	45.50	+6.23
<i>RoBERTa-based models</i>			
RoBERTa-base	20.46	13.66	-33.24
RoBERTa-large	24.24	17.99	-25.78
KEPLER	19.36	12.46	-35.64
CoLAKE	23.38	17.16	-13.74
LM-CORE(r,y)	34.60	<u>34.25</u>	-1.01
LM-CORE(r,w)	41.96	44.75	+6.65

not have *date* entity type and hence, the poor performance of models using Wikidata. We also note that on the SQuAD and Concept Net subsets, the knowledge enhanced models do not offer significant improvements over the base language models. While Google-RE and T-REx focus more on factual world knowledge (present in abundance in YAGO and Wikidata), SQuAD and ConceptNet concentrate more on commonsense knowledge (limited in YAGO and Wikidata). This is a major focus of our continuing work on enhancing the external knowledge with commonsense knowledge bases such as ConceptNet (Speer et al., 2017) Atomic (Sap et al., 2019).

4.2 Sensitivity to Contextual Signals in Input

PLMs are often sensitive to the linguistic variations in the input and are overly reliant on the surface form of entity names for making its predictions Porer et al. (2020). For example, BERT can predict that a person with an Italian-sounding name was born in Italy even if this is factually incorrect. In order to evaluate the sensitivity and robustness of different models, we report the P@1 numbers for the LAMA-UHN (UnHelpfulNames) probing benchmark (Table 2) – a much harder subset of LAMA where input probes with helpful entity names are removed and the PLM has little or no helpful contextual signals from other tokens in the probe. We observe that the LM-CORE variants significantly outperform the base language models and their knowledge enhanced variants. Further, note that while all the baseline models suffer a significant fall in performance (expected due to the hardness of LAMA-UHN), the drop in performance of LM-CORE variants is much less. This indicates that

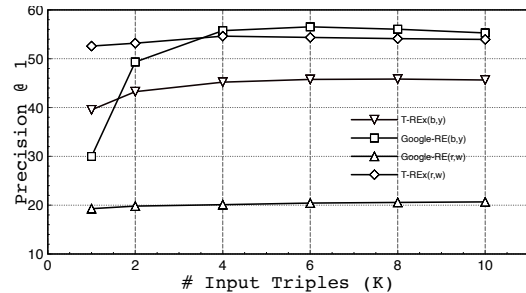


Figure 3: Effect of k on performance of different models.

having access to relevant external knowledge helps reduce the dependence on linguistic signals and results in the *robust* outperformance of LM-CORE variants.

4.3 Effect of Varying Number of Input Triples to LM-CORE

We analyze the effect of varying the number of candidates (k) during retrieval in Figure 3. We discuss with respect to Google-RE and T-REx subsets as our factual knowledge triples are most relevant for answering queries in these subsets (in comparison to commonsense queries in ConceptNet).

We plot the Precision@1 (P@1) against increasing k values from 1 to 10 for LM-CORE(b,y) and LM-CORE(r,w) variants. We do not observe any consistent optimal k value across variants and data subsets. To add, there is no significant difference between P@1 values as k varies from 4 to 10. Hence, in order to maximize our recall while keeping the computational expense in mind, we select $k = 8$ for our experiments.

4.4 Role of LM-CORE Pre-training and Retrieved Knowledge

We now study the role LM-CORE pre-training plays in helping the model access and utilize the retrieved knowledge and ensure that the model does not just rely on the knowledge stored in its parameters. We also study the effect of augmenting the base LMs with knowledge retrieved by LM-CORE. In addition to providing an insight into the quality of the knowledge retrieved by LM-CORE, this will also help us better understand the ability of LM-CORE to utilize the retrieved knowledge.

We consider the following four variants on the LAMA probe (Table 3):

1. RoBERTa-base

Table 3: Ablation study analyzing the effectiveness of LM-CORE pre-training and contextually retrieved knowledge (Precision@1 values).

	All	Google RE	T-REx	SQuAD	Concept Net
RoBERTa (base)	20.46	5.35	24.21	12.94	18.47
RoBERTa-base + LM-CORE triples	30.06	9.79	38.71	7.69	16.26
LM-CORE(r,w) + random triples	19.51	9.05	22.74	13.99	16.92
LM-CORE(r,w)	41.69	20.56	54.11	15.73	18.38

2. RoBERTa-base + triples retrieved by LM-CORE’s Contextual Knowledge Retriever
3. LM-CORE(r,w) + random triples
4. LM-CORE(r,w)

We observe that LM-CORE(r,w)’s performance (41.69 P@1) significantly exceeds RoBERTa-base’s performance using the same triples in input (30.06 P@1), demonstrating that our training procedure equips the model with the capability of identifying and using relevant external knowledge effectively. There is a large drop in performance (from 41.69 P@1 to 19.51 P@1) when LM-CORE(r,w) is provided with random triples in input. This shows that the model exclusively accesses external knowledge to answer queries correctly. While the performance drops, it is important to note that the P@1 is similar to RoBERTa-base (20.46 P@1), highlighting that our training procedure does not lead to *catastrophic forgetting* and the model is able to rely on the knowledge stored in its parameters when semantically relevant triples are not provided in the input. Finally, although RoBERTa-base when augmented with contextually relevant triples does not perform competitively with LM-CORE, it demonstrates considerable improvement over the base RoBERTa model. This shows that high-quality relevant external knowledge has the potential to improve factual prediction, further reinforcing our motivation to train models to efficiently retrieve and use this knowledge.

4.5 Downstream Tasks

We consider two downstream tasks to study the effectiveness of LM-CORE for different NLP applications. We take Zero-Shot Relation Extraction (ZSRE) (Levy et al., 2017) and open-domain question answering over Web Questions (WQ) (Berant et al., 2013) dataset as the representative knowledge-intensive tasks. Tables 4 and 5 re-

port the performance of LM-CORE and various other baselines for the two tasks, respectively. We use the LM-CORE(b,w) variant for these experiments as most baselines use BERT as the LM and Wikipedia as the knowledge base. For the ZSRE task, we use the data splits and evaluation systems provided as part of the KILT benchmark (Petroni et al., 2021). We find that for the ZSRE task, LM-CORE achieves a significantly higher F-1 score (**74.80**) compared to the second-best RAG model (49.95). Also, note that the online evaluator for the task considers exact string match (including casing, punctuations, etc.) for computing accuracy numbers but not for computing other metrics. Hence, the reported accuracy number for LM-CORE represent a lower bound as we don’t have access to the same pre-processing pipeline to process its output. For the WQ dataset, we find that LM-CORE outperforms BERT with BM25 and neural retrievers, and the DrQA system. We observe that LM-CORE is outperformed by ORQA, designed explicitly for this task, and RAG (a retrieval augmented generative model). However, do note that all the models except LM-CORE have access to much larger knowledge source (complete Wikipedia corpus. \approx 2B words), whereas LM-CORE only has access to the KB triples (21M triples, \approx 140M words). As we show in the following subsection, with access to additional external knowledge, the performance of LM-CORE can improve significantly.

Table 4: F1 and Accuracy on Zero Shot RE. * The accuracy for LM-CORE is the lower-bound number as the online evaluator considers exact string match to compute accuracy.

	#params	F1	Accuracy
BERT+DPR (Karpukhin et al., 2020)	330M	37.28	6.93
T5 (base)	220M	13.52	9.02
BART (large)	406M	12.21	9.14
BART+DPR	626M	34.47	30.43
RAG (Lewis et al., 2020)	626M	49.95	44.74
LM-CORE(b,w)	110M	74.80	14.24*

4.6 Handling Knowledge Updates

Once PLMs have been trained, it is expensive to retrain them with new and updated knowledge. LM-CORE, on the other hand, can easily access updated knowledge as it is *external* to the model. Instead of storing all the KB facts in the PLM, LM-CORE essentially learns semantically relevant associations between the input text and KB relations and can easily *query* for the relevant knowledge by using the learned relationship associations. We illustrate this ability to handle dynamic knowledge by in-

Table 5: Accuracy on Web Questions. BM25, Neur. Retriever, DRQA and LM-CORE perform static retrieval.

	Accuracy
BM25 + BERT	17.7
Neur. Retriever + BERT	7.3
DrQA (Chen et al., 2017)	20.7
LM-CORE (b,w)	21.9
ORQA (Lee et al., 2019b)	36.4
RAG (Lewis et al., 2020)	45.5

roducing new triples in the KB and verifying if LM-CORE is able to leverage this new information to correct its earlier predictions. We consider the LM-CORE(b,y) variant for this experiment. We randomly sample 100 instances from the LAMA probe where the model failed and manually analyze these instances to identify the cases where the corresponding fact was not present in the YAGO KB. There were a total of 41 such instances and we manually added the correct facts needed to answer the corresponding questions in YAGO. We then presented the 41 inputs again to the model with the updated KB. This time, the model used this newly added knowledge and was able to correct its prediction *without any re-training* for 36 out of 41 cases (87.8%). As discussed in the following sub-section (4.7), a majority of errors made by LM-CORE are due to missing facts in the KB and we expect that most of such errors can be corrected by having access to a larger, more comprehensive Knowledge Base.

4.7 Discussions

We now present some representative examples to illustrate the successes and failures of LM-CORE. Consider a test probe from the Google-RE subset of LAMA – *Phil Mogg is a member of ____*. Here, the correct output token is *UFO*, the band and BERT model incorrectly predicts *parliament* as the output token. This highlights the sensitivity of PLMs on context; BERT’s prediction seems to be derived from its memorization of the frequently encountered phrase *member of parliament* during pre-training. We argue that the contextual knowledge retrieved by LM-CORE which includes the relevant fact $\langle \text{Phil Mogg}; \text{member of}; \text{UFO (band)} \rangle$ has helped the model to produce the correct output. We present more such successful examples in the Appendix (Tables 10 and 11).

Next, we analyzed the cases where the proposed framework produced incorrect output and observed three major reasons for errors – (i) the required knowledge was not present in the knowledge base;

(ii) the required knowledge was not retrieved despite being present in the knowledge base; and (iii) the system made errors after retrieving the relevant knowledge. The first problem cause could be addressed by enhancing the knowledge base as shown in Section 4.6. The other two causes of failure highlight the scope of improvement in our retrieval module as well as pre-training module, where further training could help the model make better use of the retrieved knowledge. Some representative examples of these different cases are presented in the Appendix (Table 12). Finally, we noticed some errors that could be attributed to the characteristics of the LAMA probe. Specifically, there are input probes that refer to entities without providing any additional context for disambiguation. For example, the sentence “*James Johnson was born in ____*” has no clues to determine whether the prompt is referring to the basketball player, Virginia congressman, or the Governor of Georgia with this name. We also noticed certain probes where there are multiple correct completions and the benchmark considers only one of these as the correct answer. For example, “*Michelangelo is a ____ by profession*” can be correctly completed by *poet*, *painter* or *architect*, but the evaluation considers only *poet* as the correct answer. We also noticed some input examples with highly unambiguous language. For example, “*X died in ____*”, can refer to either X’s place of death or date of death but only the former is accepted as the correct answer. Lastly, there are cases where slight (and correct) variations of the expected answer are evaluated as incorrect by the probe. For example, for the prompt “*Harashima is ____ citizen.*” *Japan* is provided as the correct answer while the prediction made by LM-CORE (*Japanese*) is considered incorrect.

5 Conclusion

We presented LM-CORE, a framework to train language models with contextually relevant external knowledge. We show that having access to external knowledge leads to significant and robust out-performance over base language models and their knowledge enhanced versions on knowledge probing and two downstream tasks. We also showed how LM-CORE can handle knowledge updates and presented a thorough error analysis that helped us identify possible directions of future work.

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, Online. Association for Computational Linguistics.
- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.
- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544. ACL.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, page 2787–2795, Red Hook, NY, USA. Curran Associates Inc.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading Wikipedia to answer open-domain questions](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
- Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *ICLR*.
- 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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.
- 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.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019a. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019b. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.
- Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel 0001, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. [K-BERT: enabling language representation with knowledge graph](#).

- In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2901–2908. AAAI Press.
- 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, Florence, Italy. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Robert L Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP*.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick S. H. Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel 0001. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2523–2544. Association for Computational Linguistics.
- 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, Hong Kong, China. Association for Computational Linguistics.
- Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. [E-BERT: Efficient-yet-effective entity embeddings for BERT](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.
- 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(140):1–67.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- 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, Online. Association for Computational Linguistics.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. [AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.
- Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.
- Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. [Yago: A core of semantic knowledge](#). In *Proceedings of the 16th International Conference on World Wide Web, WWW ’07*, page 697–706, New York, NY, USA. Association for Computing Machinery.
- Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. [CoLAKE: Contextualized language and knowledge embedding](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Denny Vrandečić and Markus Krötzsch. 2014. [Wiki-data: A free collaborative knowledgebase](#). *Commun. ACM*, 57(10):78–85.
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing 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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1405–1418. Association for Computational Linguistics.
- Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald

- Tesauro, Bowen Zhou, and Jing Jiang. 2017. [R³: Reinforced reader-ranker for open-domain question answering](#).
- 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.
- Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. [End-to-end open-domain question answering with BERTserini](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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, Florence, Italy. Association for Computational Linguistics.
- Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. [Factual probing is \[MASK\]: Learning vs. learning to recall](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

Appendix

A LM-CORE Training Details

We used the Hugging Face Transformer² models BERT and RoBERTa as base models for LM-CORE pre-training. We use the `BASE` (12-layer, 768-hidden, 12-heads) size models and initialize from `bert-base-uncased` and `roberta.base` parameters, respectively. This is consistent with the initialization in Peters et al. (2019); Zhang et al. (2019); Wang et al. (2021b) and is a common practice to save up on pre-training time.

We used the Adam (Kingma and Ba, 2015) optimizer and a learning rate of $3e-5$ across all settings. We could not perform a lot of hyperparameter tuning owing to the computational requirements of the task. Pre-training was done using 8 Nvidia A100 GPUs with a batch size of 512 using gradient accumulation. The masked LM loss continued to decrease at the end of pre-training, suggesting further improvement in performance can be expected. The pre-trained checkpoints for all four variants of LM-CORE can be found [here](#).

A.1 Pre-Training corpus

We use the English Wikipedia (December 20, 2018) snapshot³ to create our pre-training corpus and WikiExtractor⁴ to process the dumps. This Wikipedia version contains about 5.5M documents. We retain the hyperlinks while extracting Wikipedia articles as we use them for creating entity masks (Section 3.1). Following the entity masking strategy described in Section 3.1, we obtain our pre-training corpus which contains ~ 46.3 M sentences in total.

During pre-training the base LMs, we sample sentences containing valid masks. The pre-training corpus is maintained consistent across all LM-CORE variants.

A.2 Retrieval corpus

We use two popular knowledge bases (KBs) in LM-CORE - YAGO 4 (Suchanek et al., 2007) and Wikidata5M (Wang et al., 2021b). The statistics of the KBs – number of facts, entities and relations

²<https://github.com/huggingface/transformers>

³<https://archive.org/download/enwiki-20181220/enwiki-20181220-pages-articles.xml.bz2>

⁴<https://github.com/attardi/wikiextractor>

can be found in Table 6. We describe the preprocessing steps followed to obtain the respective final retrieval corpora in the following subsections.

A.2.1 YAGO

YAGO 4⁵ is in RDFS format. YAGO facts are derived from Wikidata, however, all the entities are arranged in a taxonomy mapped to schema.org.

We pre-process YAGO to remove triples involving relationships such as *image*, *logo* and *url* that point to meta-data such as images and other files. We also filter out triples that point to RDF literals or Wikidata URLs.

A.2.2 Wikidata5M

We use the Wikidata5M subset of Wikidata as made available by Wang et al. (2021b)⁶. This subset of Wikidata is aligned with Wikipedia such that each entity in Wikidata5M has a corresponding entry in Wikipedia. We used the `raw` graph as provided in the dataset, the statistics of which are reported in Table 6.

Table 6: Statistics of the knowledge bases used for retrieval in terms of number of triples, entities and relations

	Facts	Entities	Relations
Yago	17,421,942	4,927,897	131
Wikidata5M	21,285,880	4,797,808	821

A.3 Example Retrieved Triples

We provide a closer look into our pre-training approach by showing examples of masked input sentences and the retrieved triple candidates from the knowledge base (Table 7). We observe that the facts retrieved are highly relevant for predicting the masked entities in the input context.

B Additional Experiments

B.1 How LM-CORE Compares with Other Retrieval Paradigms

Table 8 also reports results of REALM (Guu et al., 2020) – a retrieval-based language model that retrieves relevant documents from a text corpus during pre-training. We observe that LM-CORE outperforms REALM on the ConceptNet,

⁵<https://yago-knowledge.org/downloads/yago-4>

⁶<https://deepgraphlearning.github.io/project/wikidata5m>

Input	Masked Token	Candidates (correct fact in bold)
Henri Jules Louis Marie Rendu (24 July 1844 – 16 April 1902) was a French physician born in [MASK].	Paris	(Henri Jules Louis Marie Rendu; birth date; 1844-07-24) (Henri Jules Louis Marie Rendu; birth place; Paris) (Henri Jules Louis Marie Rendu; death date; 1902-04-16) (Henri Jules Louis Marie Rendu; nationalit;y France) (Henri Jules Louis Marie Rendu; given name; Henri)
Weisenborn attended the [MASK].	University of Chicago	(Gordon Weisenborn; alumni of; University of Chicago) (Clara Weisenborn; member of; Republican Party (United States)) (Günther Weisenborn; nationality; Germany) (Günther Weisenborn; death place; West Berlin) (Clara Weisenborn; nationality; United States)
Dehorokkhi (English: Bodyguard) is a Bangladeshi [MASK] directed by Iftakar Chowdhury.	action film	(Dehorokkhi; director; Iftakar Chowdhury) (Dehorokkhi; in language; Bengali language) (Dehorokkhi; genre; Action film) (Bangladeshi Idol; in language; Bengali language) (British Bangladeshi Who’s Who; in language; English language)
Palaemon macrodactylus is a [MASK] of shrimp of the family Palaemonidae.	species	(Palaemon macrodactylus; parent taxon Palaemon (genus)) (Palaemon macrodactylus; parent taxon; Palaemon (genus)) (Palaemon macrodactylus; taxonomic rank; Species) (Palaemonidae; taxonomic rank; Family (biology)) (Palaemonidae; parent taxon; Palaemonoidea)

Table 7: Examples of masked input sentences (from Wikipedia) and top-5 retrieved candidates during pre-training.

DoB (Google-RE), and 1-1 (T-REx) subsets, while REALM outperforms the proposed solution in other subsets of the LAMA probe. We specifically highlight an absolute 15 points improvement on the `date-of-birth` relation despite REALM using explicit date masks while training whereas our training corpus only has entity masks. This indicates that our model can use the contextual knowledge provided by the retriever module even though it is not explicitly shown such knowledge during training.

Note that while REALM is similar to our proposed solution as far as the idea of retrieving relevant knowledge is concerned, the key difference in the two approaches lies in the source of knowledge being used. REALM relies on an unstructured text corpus (Wikipedia) as the source of knowledge and employs a computationally complex retrieve and read paradigm requiring additional training of the knowledge retriever model. Our proposed solution, on the other hand, uses structured knowledge which offers the advantage of being (almost) unambiguous and less resource-hungry compared to unstructured text. We present the resource requirements of our approach and REALM in Table 9. Note that the size of the external knowledge (in number of words) used by REALM is an order of magnitude greater, and requires three times the number of parameters compared to our model. Fur-

thermore, REALM was trained for 200K steps with a batch size of 512 on an 80 TPU cluster, whereas our proposed solution is much more efficient being trained for 1K steps with a batch size of 512 on a machine with 8 Nvidia A100 GPUs. This computational efficiency of our proposed solution allows us to continue further work on improving our performance by enhancing the structured knowledge base and bridge the performance gap with more complex and computationally expensive models such as REALM.

C LAMA Evaluation

We use the official LAMA data code⁷ for evaluating P@1 numbers in Table 1. All the BERT-based models are evaluated using this repository. The LAMA code provides functionality for evaluating RoBERTa models trained in the fairseq framework. Hence, we evaluate RoBERTa-base, RoBERTa-large and KEPLER (Wang et al., 2021b) using this code. The KEPLER repo also points to this code for evaluation. CoLAKE (Sun et al., 2020), has adapted the official code⁸ to allow huggingface transformer checkpoints as input, and hence this code is used for CoLAKE, LM-CORE(r,y) and LM-CORE(r,w) evaluation. We ensure the model

⁷<https://github.com/facebookresearch/LAMA>

⁸<https://github.com/txsun1997/CoLAKE>

Table 8: Precision at one (P@1) on LAMA probe. We consider only LM-CORE(b,y) and LM-CORE(b,w) as REALM is trained on BERT. Best results in each column are highlighted in bold and the second best performance is underlined.

	Google-RE				T-REx				SQuAD	Concept Net
	DoB	PoB	PoD	All	1-1	N-1	N-M	All		
BERT-base	1.59	15.46	10.33	9.12	67.94	32.67	23.54	30.83	14.29	15.88
BERT-large	1.59	15.53	12.16	9.76	74.23	31.30	25.30	31.05	<u>17.61</u>	18.72
LM-CORE(b,y)	64.44	<u>52.71</u>	<u>50.98</u>	<u>56.04</u>	74.37	51.18	34.57	45.83	15.61	14.78
LM-CORE(b,w)	0.66	37.62	31.11	23.13	81.79	<u>59.86</u>	<u>45.48</u>	<u>55.32</u>	17.28	<u>16.15</u>
REALM	<u>49.06</u>	79.56	64.13	67.36	<u>55.81</u>	69.54	66.98	68.18	27.96	4.78

Model	no. of params	Retrieval corpus size	Resources used
LM-CORE(b,y)	110M	17M KB triples (~100M words)	8 GPUs
LM-CORE(b,w)	110M	21M KB triples (~140M words)	8 GPUs
REALM	330M	5.5M documents (~2B words)	80 TPUs

Table 9: Resource requirements of LM-CORE and REALM. REALM requires additional ICT pre-training over all Wikipedia documents for initialization.

vocabularies and data is consistent across evaluation. We have used author provided/recommended code and publicly available checkpoints from the official code repositories for all baselines.

C.1 LAMA: Qualitative Analysis

Table 10 and 11 show examples spanning different relationships in LAMA where LM-CORE(b,y) and LM-CORE(r,w) are able to make correct predictions. We also compare the predictions with BERT-base and RoBERTa-base respectively and highlight how these PLMs struggle to make knowledgeable predictions.

C.2 LAMA: Error Analysis

We present various failure cases for LM-CORE in Table 12. These are representative of the type of errors we encountered, however, we observed that majority of the errors resulted due to correct facts missing from the KB.

C.3 Complete LAMA-UHN results

The complete LAMA-UHN results over all subsets of Google-RE and T-REx can be found in Table 13.

D Downstream Evaluation

We discuss the experimental setup and hyperparameter settings for our downstream tasks.

D.1 Zero Shot RE

We consider the open domain version of Zero Short RE (Levy et al., 2017) from Petroni et al. (2021). The dataset is split into three disjoint sets – train (147,909 samples, 84 relations), dev (3,724 samples, 12 relations) and test (4,966 samples, 24 relations). The systems are evaluated on relations never seen during training.

We fine tune our model for 2 epochs with a batch size of 96. We use the Adam (Kingma and Ba, 2015) optimizer and a learning rate of 3e-5. We performed multiple trials by tuning the number of epochs in {1, 2, 5}.

D.2 Web Questions

Web Questions (Berant et al., 2013) was created using questions that were sampled from the Google Suggest API. We used the same splits as Lee et al. (2019b) with training, dev and test sets containing 3417, 361 and 2032 samples respectively.

We fine tuned our model for 20 epochs – we experimented with number of epochs in {10, 20, 30, 50}. We use the Adam (Kingma and Ba, 2015) optimizer and a learning rate of 3e-5.

E Risks Statement

This work considers training of large language models using large textual corpora as well as structured knowledge bases. The model learns the nuances of the language and correlations between different real-world entities based on the data that is being used for training the model. Hence, there is a chance that the biases and noise in the training data will creep into the model parameters as well that can lead to a biased model behavior. We need

to be careful in deploying the model and extrapolating the output of the model in applications such as search, conversational systems and recommendation systems where model's inherent biases can lead to catastrophic impacts on the user.

	Relation	Input query	LM-CORE(b,y) prediction	BERT-base prediction	Candidates
Google-RE	birth-place	Stanley Corrsin was born in ____.	Philadelphia	London	(Stanley Corrsin birth date 1920-04-03) (Stanley Corrsin nationality United States) (Stanley Corrsin birth place Philadelphia) (Stanley Corrsin given name Stanley (given name)) (Stanley Corrsin death date 1986-06-02) (Stanley Corrsin has occupation Physicist) (Stanley Corrsin alumni of University of Pennsylvania) (Stanley Corrsin member of American Academy of Arts and Sciences)
	birth-date	Tom Coppola (born ____).	1945	1975	(Tom Coppola birth date 1945-06-06) (Tom Coppola given name Tom (given name)) (Tom Coppola nationality United States) (Tom Coppola family name Coppola (surname)) (Tom Coppola alumni of USC Thornton School of Music) (Christopher Coppola birth date 1962-01-25) (Anton Coppola nationality United States) (Chris Coppola birth date 1962-01-25)
	death-place	Aglaja Orgeni died in ____.	Vienna	Bucharest	(Aglaja Orgeni death date 1926-03-15) (Aglaja Orgeni death place Vienna) (Aglaja Orgeni birth date 1841-12-17) (Aglaja Orgeni birth place Rimavská Sobota) (Aglaja Orgeni nationality Austria) (Aglaja Orgeni nationality Hungary) (Aglaja Orgeni has occupation Opera singer) (Aglaja Orgeni death place Vienna)
T-REx	P106	Cigoli is a ____ by profession.	architect	lawyer	(Cigoli has occupation Architect) (Cigoli nationality Italy) (Cino Cinelli has occupation Businessperson) (Francesco Cirio has occupation Businessperson) (Cigoli birth place Cigoli, San Miniato) (Emilio Cigoli has occupation Stage actor) (Francesco Cigalini has occupation Mathematician) (Ciputra has occupation Businessperson)
	P463	Phil Mogg is a member of ____.	UFO	parliament	(Phil Mogg member of UFO (band)) (Phil Mogg nationality United Kingdom) (Phil Mogg birth date 1948-04-15) (Phil Mogg birth place London) (Mo Mozzali member of Minneapolis Millers) (John Mogg, Baron Mogg nationality United Kingdom) (Jamie Moyer member of Colorado Rockies) (Jamie Moyer member of Philadelphia Phillies)
	P407	Summerfolk was written in ____.	russian	english	(Summerland (novel) in language English language) (Summerfolk in language Russian language) (The World That Summer genre Neofolk) (Summerfolk author Maxim Gorky) (Summer (novel) in language English language) (Summertime (novel) in language English language) (A Summer Tale date published 2000) (Summerteeth in language English language)
	P1303	Nigel Pulsford plays ____.	guitar	sgt	(Nigel Pulsford has occupation Guitarist) (Nigel Pulsford given name Nigel) (Nigel Pulsford birth date 1963-04-11) (Nigel Pulsford nationality United Kingdom) (Nigel Pulsford nationality Wales) (Nigel Pulsford birth place Newport, Wales) (William Pulsford nationality United Kingdom of Great Britain and Ireland) (Reginald Purdell has occupation Actor)

Table 10: Illustrative examples of cases where LM-CORE(b,y) model successfully output the correct completions for various probes in LAMA. Candidates containing correct fact are in bold.

	Relation	Input query	LM-CORE(r,w) prediction	RoBERTa-base prediction	Candidates
Google-RE	birth-place	Sebastiano Maffettone was born in ____.	Naples	Rome	(Sebastiano Maffettone place of birth Naples) (Sebastiano Mazzoni place of birth Florence) (Sebastiano Mocenigo place of birth Venice) (Sebastiano Martinelli place of birth Italy) (Sebastiano Baggio place of birth Italy) (Stanley Corrsin has occupation Physicist) (Sebastiano Vassalli place of birth Genoa) (Sebastiano Poma place of birth Parma)
	P413	Rivaldo plays in ____ position.	midfielder	the	(Rivaldo position played on team forward) (Rivaldo position played on team midfielder) (Rivaldo Gonzalez position played on team midfielder) (Rivaldo Coetzee position played on team defender) (Rivaldo Vítor Mosca Ferreira Júnior position played on team forward) (Rivaldo member of sports team brazil national football team) (Rivaldo member of sports team brazil national under-20 football team) (Rivaldo member of sports team São Paulo fc)
T-REx	P176	Amiibo is produced by ____.	Nintendo	Samsung	(amiibo manufacturer nintendo) (amiibo tap: nintendo's greatest bits publisher nintendo) (animal crossing: amiibo festival publisher nintendo) (amiibo tap: nintendo's greatest bits platform wii u) (animal crossing: amiibo festival developer nintendo entertainment planning & development) (animal crossing: amiibo festival platform wii u) (amiibo instance of internet protocol) (animal crossing: amiibo festival genre party game)
	P138	Uraninite is named after ____.	uranium	the	(uraninite named after uranium) (uraniborg named after urania) (uranopilite named after compound) (uranopilite named after uranium) (uraniinae instance of taxon) (urania parent taxon uraniinae) (uranocircite-ii named after uranium) (30 urania named after urania)
	P159	The headquarter of Stelco is in ____.	Hamilton	Madrid	(Stelco headquarters location Hamilton) (Stelco lake erie works located in the administrative territorial entity Ontario) (Stelco owned by U.S. steel) (Stelco lake erie works country Canada) (Stelco industry ferrous metallurgy) (Stelco instance of business) (Stec, inc. headquarters location California) (Stekey located in the administrative territorial entity louisiana)
	P37	The official language of Virrat is ____.	Finnish	English	(Virrat official language Finnish) (Virrat country Finland) (Virrat located in the administrative territorial entity Pirkanmaa) (Virrat located in time zone utc+2) (Virrat located in time zone utc+03:00) (Virrat instance of municipality of Finland) (Virrat instance of town) (Virrat instance of city)

Table 11: Illustrative examples of cases where LM-CORE(r,w) model successfully output the correct completions for various probes in LAMA. Candidates containing correct fact are in bold.

Input Query	Expected Answer	Model Output	Retrieved Candidates	Comments
Hans Gefors was born in ____.	Stockholm	Hamburg	(Hans Raj Hans birth date 1953-11-30) (Hans Raj Hans given name Hans (name)) (Hans Raj Hans nationality India) (Hans Raj Hans has occupation Politician) (Hans Raj Hans member of Indian National Congress) (Claus Gerson birth place Hamburg) (Hans Geister birth date 1928-09-28) (Hans Gericke nationality Germany)	Corresponding fact not present in KB. We speculate that the candidate in bold led the model to predict Hamburg. BERT predicted <i>Oslo</i> as the answer.
Victor Salvi plays ____.	harp	quarterback	(Victor Salvi given name Victor (name)) (Victor Salvi nationality United States) (Victor Salvi death place Milan) (Victor Salvi death date 2015-05-10) (Victor Salvi birth place Chicago) (Victor Salvi birth date 1920-03-04) (Joan Lui actor Francesco Salvi) (Victor Salvi birth date 1920-03-04)	Corresponding fact not present in KB
CBeebies is owned by ____.	BBC	Microsoft	(CBeebies founding date 2002) (Gigglebiz creator CBeebies) (CBEF contained in place Ontario) (CBEF location Ontario) (Bambi production company The Walt Disney Company) (CBE Software founding date 2006) (Paddington Bear (TV series) production company ITV Central) (CBS Interactive parent organization CBS Corporation)	Corresponding fact not present in KB.
Ivan Petch was born in ____.	Concord	Sydney	(Ivan Petch birth date 1939-03-01) (Ivan Petch birth place Concord, New South Wales) (Ivan Petch family name Petch) (Ivan Petch given name Ivan (name)) (Ivan Petch nationality Australia) (Ivan Petch has occupation Politician) (Ivan Petch has occupation Electrical engineer) (Ivan Petch alumni of Fort Street High School)	Correct fact is retrieved. However, the model is still not able to predict correct output.
Scientist was born in ____.	Kingston	London	(Scientist (musician) birth date 1960-04-18) (Thomas Young (scientist) has occupation Physicist) (I Am a Scientist date published 1994) (Thomas Prince (scientist) has occupation Physicist) (Bambi production company The Walt Disney Company) (Allen Taylor (scientist) nationality United States) (Lawrence Roberts (scientist) nationality United States) (David Thomas (Canadian scientist) has occupation Biochemist)	Ambiguous query, leads to poor retrieval results.
Moldova shares border with ____.	Ukraine	Romania	(Moldova shares border with Ukraine) (Moldova shares border with Romania) (Moldova shares border with aa) (Moldova shares border with Jabara) (Moldova Nouă shares border with Bela Crkva) (Moldova contains administrative territorial entity Transnistria) (Moldova diplomatic relation Russia) (Moldova diplomatic relation European Union)	Multiple answers correct, however, LAMA considers only one.

Table 12: Illustrative examples of cases where the proposed solution produced incorrect completions.

Table 13: Mean precision at one (P@1) of various models on LAMA-UHN probe. Best results are highlighted in bold and the second best performance is underlined

	Complete	Google-RE				T-REx			
		DoB	PoB	PoD	All	1-1	N-1	N-M	All
<i>BERT-based models</i>									
BERT-base	18.72	<u>1.59</u>	6.98	3.98	4.18	62.86	21.99	17.32	22.16
BERT-large	19.92	<u>1.59</u>	7.71	5.66	49.86	70.13	22.35	19.62	23.62
ERNIE	15.81	1.42	6.57	1.38	3.12	55.68	17.90	15.40	18.76
LM-CORE(b,y)	<u>41.33</u>	64.44	46.80	46.02	52.42	<u>71.04</u>	<u>44.42</u>	<u>29.11</u>	<u>39.75</u>
LM-CORE(b,w)	45.50	0.66	<u>30.93</u>	<u>23.39</u>	<u>18.33</u>	80.05	55.16	41.17	50.92
<i>RoBERTa-based models</i>									
RoBERTa	13.66	<u>1.85</u>	4.18	0.55	2.19	53.36	13.99	15.80	16.62
RoBERTa-large	17.99	1.41	5.68	0.36	2.48	<u>67.27</u>	20.47	17.86	21.74
KEPLER	12.46	1.47	4.88	0.91	2.42	48.70	12.60	14.44	15.08
CoLAKE	17.16	1.79	6.89	5.83	4.84	59.84	18.85	17.61	20.37
LM-CORE(r,y)	<u>34.25</u>	46.33	35.66	<u>19.31</u>	33.77	63.83	<u>37.74</u>	<u>25.20</u>	<u>34.12</u>
LM-CORE(r,w)	44.75	0.38	<u>24.80</u>	20.04	<u>15.07</u>	67.53	55.72	39.77	50.07