Exploiting Structured Knowledge in Text via Graph-Guided Representation Learning

Tao Shen[1]*, **Yi Mao**[2], **Pengcheng He**[2], **Guodong Long**[1], **Adam Trischler**[3] and **Weizhu Chen**[2]
[1]Australian AI Institute, School of Computer Science, FEIT, University of Technology Sydney
[2]Microsoft Dynamics 365 AI [3]Microsoft Research, Montréal
tao.shen@student.uts.edu.au, guodong.long@uts.edu.au
{maoyi,penhe,adtrisch,wzchen}@microsoft.com

Abstract

In this work, we aim at equipping pre-trained language models with structured knowledge. We present two self-supervised tasks learning over raw text with the guidance from knowledge graphs. Building upon entity-level masked language models, our first contribution is an entity masking scheme that exploits relational knowledge underlying the text. This is fulfilled by using a linked knowledge graph to select informative entities and then masking their mentions. In addition, we use knowledge graphs to obtain distractors for the masked entities, and propose a novel distractor-suppressed ranking objective that is optimized jointly with masked language model. In contrast to existing paradigms, our approach uses knowledge graphs implicitly, only during pre-training, to inject language models with structured knowledge via learning from raw text. It is more efficient than retrieval-based methods that perform entity linking and integration during finetuning and inference, and generalizes more effectively than the methods that directly learn from concatenated graph triples. Experiments show that our proposed model achieves improved performance on five benchmarks, including question answering and knowledge base completion.

1 Introduction

Self-supervised pre-trained language models (LMs) like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) learn powerful contextualized representations. With task-specific modules and finetuning, they have achieved state-of-the-art results on a wide range of natural language processing tasks. Nevertheless, open questions remain about what these models have learned and improvements can be made along several directions. One such direction is, when downstream task performance depends on relational knowledge – the kind modeled by knowledge graphs [1] (KGs) – directly finetuning a pre-trained LM often yields sub-optimal results, even though some works (Petroni et al., 2019; Davison et al., 2019) show pre-trained LMs have been partially equipped with such knowledge.

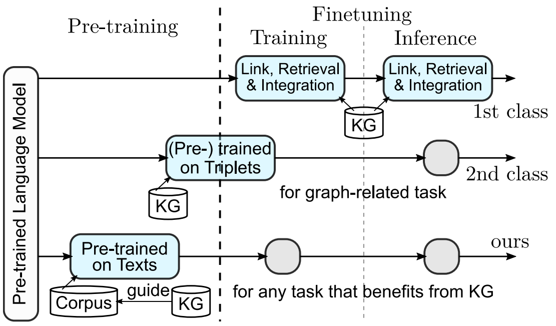
Pre-training
Finetuning
Training
Inference
Pre-trained Language Model
Link, Retrieval & Integration
Link, Retrieval & Integration
KG
1st class
(Pre-) trained on Triplets
KG
2nd class
for graph-related task
Pre-trained on Texts
ours
Corpus
guide
KG
for any task that benefits from KG

Figure 1: Taxonomy of different approaches to integrating pre-trained LMs with knowledge graphs.

To address this shortcoming, several recent works attempt to integrate KGs into pre-trained LMs. These approaches can be coarsely categorized into two classes, as shown in Figure 1. The first line of methods retrieves a KG subgraph (Liu et al., 2019a; Lin et al., 2019; Lv et al., 2019) and/or pre-trained graph embeddings (Zhang et al., 2019b; Peters et al., 2019) via entity linking during both training and inference on downstream tasks. While these methods inject domain-specific knowledge directly into language representations, they rely heavily on the performance of the linking algorithm and/or the quality of graph embeddings. Graph embeddings, to be tractable over large-scale KGs, are often learned using shallow models (e.g., TransE (Bordes et al., 2013), TuckER (Balazevic et al., 2019) and ConvE (Dettmers et al., 2018)) with limited expressive power. Besides, the linking and retrieval invoked during both finetuning and

* Work done while the author was an intern at Microsoft.

[1] "Knowledge graph" and "knowledge base" are interchangeable in this paper, denoting triple-formatted graph.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8980–8994,
November 16–20, 2020. ©2020 Association for Computational Linguistics

inference are costly, hence limiting these methods' practicality.

The second class of methods (Bosselut et al., 2019; Malaviya et al., 2019; Yao et al., 2019) uses contextualized representations from pre-trained LMs to enrich graph embeddings and thus alleviates graph sparsity issues. This is especially helpful in the case of commonsense KGs (e.g., ConceptNet (Speer et al., 2017)) that consist of non-canonicalized text and hence suffer from severe sparsity (Malaviya et al., 2019). Specifically, these methods usually feed concatenated triples (e.g., [HEAD, *Relation*, TAIL]) into LMs for training or finetuning. The drawback is that focusing on knowledge base completion tends to over-adapt the models to this specific task, which comes at the cost of generalization to text-based tasks, e.g., QA.

In this work, we equip masked language models (MLMs), e.g., BERT (Devlin et al., 2019), with structured knowledge via self-supervised pretraining on raw text. Compared to the first class, we expose LMs to structured information only during pre-training, thus circumventing costly knowledge retrieval and integration in both finetuning and inference. Also the dependency on the performance of linking algorithm is greatly reduced. Compared to the second class, we learn from free-form text through MLMs rather than triples, which fosters generalization on other downstream tasks.

Specifically, given a corpus of raw text and a KG, two KG-guided self-supervision tasks are formulated to inject structured knowledge into MLMs. First, taking inspiration from Baidu-ERNIE (Sun et al., 2019a), we reformulate the masked language modeling objective to an *entity*-level masking strategy, where entities are identified by linking their text mentions to either *concepts* in a commonsense KG or *named entities* in an ontological KG (Bollacker et al., 2008). The role of KG here is to provide a "vocabulary" of entities to be masked. To further exploit implicit relational information underlying raw text, we design a KG-guided masking scheme that selects informative entities by considering both document frequency and mutual reachability of the entities detected in the text. In addition to the new entity-level MLM task above, a novel distractor-suppressed ranking task is proposed. Negative entity samples are derived from the KG and used as distractors for the masked entities to make the learning more effective.

Note that our approach never observes the KG directly, through triples or other forms. Rather, the KG plays a guiding role in the proposed self-supervised tasks. Its guidance helps the model exploit the corpus more effectively as verified in the experiments. If a downstream task can benefit from explicit exposure to KG, a method by Davison et al. (2019) can be used to transform KG triples into natural grammatical texts for our model.

We evaluate our method on five benchmarks (including question answering and knowledge base completion) and one zero-shot testing. Results show our method achieves state-of-the-art or competitive performance on all benchmarks.

2 Vanilla Masked Language Model

To ground our approach, this section summarizes MLMs for pre-training bidirectional Transformers (Devlin et al., 2019). Compared to causal LMs (Peters et al., 2018) trained unidirectionally, MLMs randomly mask some tokens and predict the masked tokens by considering their context on both sides. Formally, given a piece of text U, a tokenizer, e.g., BPE (Sennrich et al., 2016), is used to produce a sequence of tokens $[w_1, \dots, w_n]$. A certain percentage of the original tokens are then masked and replaced: of those, 80% with the special token `[MASK]`, 10% with a token sampled from the vocabulary $\mathbb{V}$, and the remaining kept unchanged. The masked sequence, denoted as $[w_1^{(m)}, \dots, w_n^{(m)}]$, is passed into a Transformer encoder to produce contextualized representations for the sequence:

$$\boldsymbol{H}=\text{TransformerEnc}\left(\left[w_1^{(m)}, \dots, w_n^{(m)}\right]\right), \quad (1)$$

where $\boldsymbol{H} \in \mathbb{R}^{d_h \times n}$ and d_h denotes the hidden size. The training loss $\mathcal{L}_M$ for MLM task is defined as

$$\mathcal{L}_M = -\frac{1}{|\mathcal{M}|}\sum_{i \in \mathcal{M}} \log P(w_i|\boldsymbol{H}_{:,i}), \quad (2)$$

where $\mathcal{M}$ denotes the set of masked token indices, and $P(w_i|\boldsymbol{H}_{:,i})$ is the probability of predicting the original token w_i given the representations computed from the masked token sequence.

3 Proposed Method

This section begins with a description of entity-level masked language modeling. Section 3.2 proposes a KG-guided entity masking scheme for the entity-level MLM task. A novel distractor-suppressed ranking task is presented in Section 3.3,

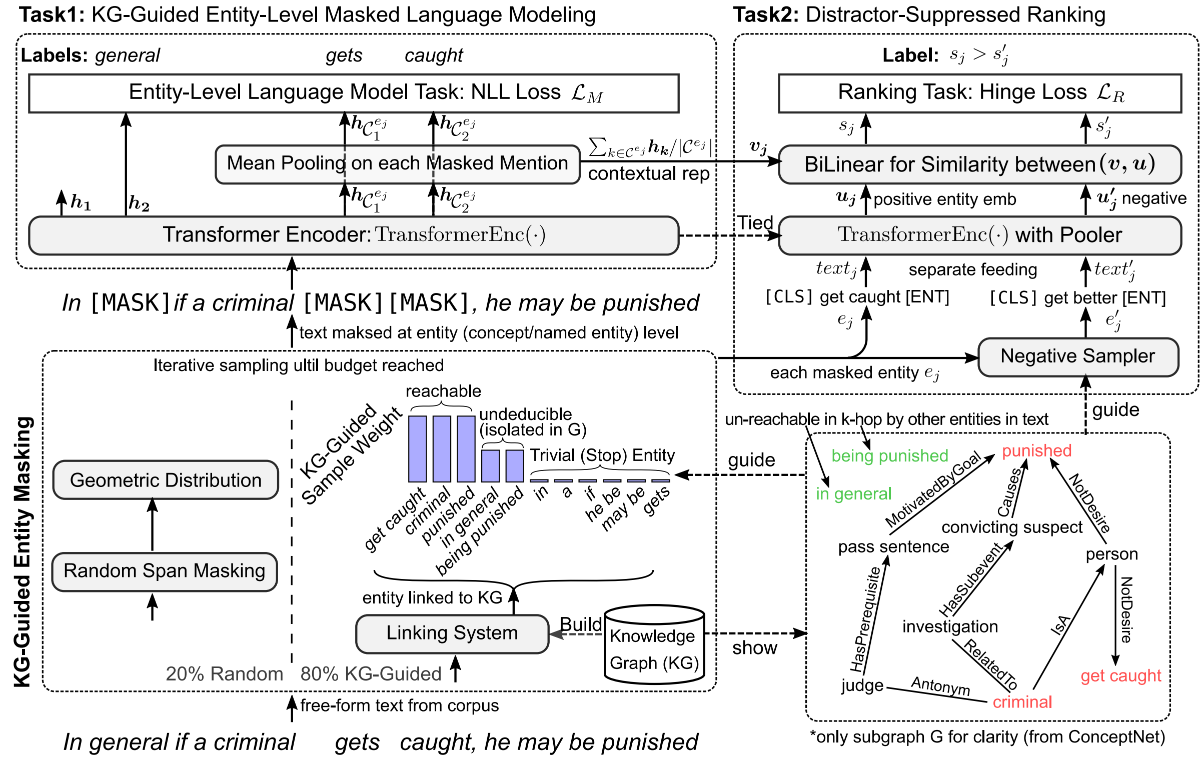
Task1: KG-Guided Entity-Level Masked Language Modeling
Labels: *general* *gets* *caught*
Entity-Level Language Model Task: NLL Loss $\mathcal{L}_M$
$\boldsymbol{h}_{\mathcal{C}_1^{e_j}}$
$\boldsymbol{h}_{\mathcal{C}_2^{e_j}}$
Mean Pooling on each Masked Mention
$\sum_{k\in\mathcal{C}^{e_j}}\boldsymbol{h_k}/|\mathcal{C}^{e_j}|$
contextual rep
$\boldsymbol{h_1}$
$\boldsymbol{h_2}$
Transformer Encoder: $\mathrm{TransformerEnc}(\cdot)$
In [MASK] *if a criminal* [MASK] [MASK], *he may be punished*
text maksed at entity (concept/named entity) level
Iterative sampling ultil budget reached
KG-Guided Entity Masking
Geometric Distribution
Random Span Masking
KG-Guided Sample Weight
reachable
undeducible (isolated in G)
Trivial (Stop) Entity
get caught
criminal
punished
in general
being punished
in
a
if
he be
may be
gets
entity linked to KG
Linking System
Build
Knowledge Graph (KG)
20% Random
80% KG-Guided
free-form text from corpus
In general if a criminal gets caught, he may be punished
Task2: Distractor-Suppressed Ranking
Label: $s_j > s'_j$
Ranking Task: Hinge Loss $\mathcal{L}_R$
s_j
s'_j
$\boldsymbol{v_j}$
BiLinear for Similarity between $(\boldsymbol{v}, \boldsymbol{u})$
$\boldsymbol{u_j}$ positive entity emb
$\boldsymbol{u'_j}$ negative
Tied
$\mathrm{TransformerEnc}(\cdot)$ with Pooler
$text_j$
separate feeding
$text'_j$
[CLS] get caught [ENT]
[CLS] get better [ENT]
e_j
e'_j
each masked entity e_j
Negative Sampler
guide
un-reachable in k-hop by other entities in text
guide
show
being punished
in general
punished
MotivatedByGoal
Causes
NotDesire
convicting suspect
pass sentence
person
HasPrerequisite
HasSubevent
investigation
IsA
NotDesire
RelatedTo
judge
Antonym
criminal
get caught
*only subgraph G for clarity (from ConceptNet)

Figure 2: **G**raph-guided Masked **L**anguage **M**odel (GLM) with two self-supervised tasks for MLM training.

which operates on the masked entities and their negative entity samples from the KG. We use a multi-task learning objective combining the two tasks above to jointly train our proposed **G**raph-guided Masked **L**anguage **M**odel (GLM). An illustration of the GLM is shown in Figure 2.

3.1 Entity-Level Masked Language Model

As aforementioned, directly training an MLM with graph triples learns structured knowledge at the cost of the model's generalization to tasks involving natural text such as question answering. Inspired by distantly supervised relation extraction (Mintz et al., 2009) assuming that any sentence containing two entities can express the relation between these two entities in a KG, we argue that it is possible for an MLM to learn structured knowledge from raw text if appropriately guided by a KG.

Roughly speaking, we take detected entity mentions as masking candidates, where the entity can be a concept/phrase in a commonsense KG or a named entity in an ontological KG. The intuition is that the mentions in text often represent knowledge-grounded, semantically meaningful text spans. Formally, we first use a KG to provide a vocabulary of entities for building an entity linking system. We then detect all entity mentions appearing in a piece of text U from a corpus. This leads to a set of linked entities $\mathcal{E} = \{e_1, e_2, \dots\} \triangleq \{e | e \in \text{KG} \land e \in U\}$ with $\mathcal{C}^{e_j}$ being the corresponding token indices in U for each entity mention e_j.

The idea of entity-level masking is not new. For example, Sun et al. (2019a) and Joshi et al. (2019) randomly mask entity candidates under uniform distribution for training MLMs. We take this idea further by building our masking scheme with the guidance of a KG, as explained below.

3.2 KG-Guided Entity Masking Scheme

In this section, we develop a new entity masking scheme to facilitate structured knowledge learning for MLMs. It explores implicit relational information underlying raw text by the guidance of a KG, and is shown to mask more informative entities compared to the previous random approaches.

In particular, the scheme is designed to avoid or reduce masking two types of entities: trivial and undeducible. Trivial entities, such as *have been* and *what do* in ConceptNet, are ubiquitous in corpora, since they are used to compose sentences. However, they express bare semantics and function similarly to stop words. On the other hand, an undeducible entity is defined as an entity that is hardly reached from any other entities detected in the same text, within certain hops over the linked KG. Examples include general modifiers and ambiguous entity linking results, as shown with green in Figure 2.

Given a masking budget (e.g., 20% of total tokens in our setting), we sample token span iteratively as follows until the budget is reached: 1) 20% of the time we sample a random token span under a geometric distribution with $p=0.2$; and 2) 80% of the time we sample an entity mention from the candidates detected in §3.1, where the probability to mask an entity mention $\mathcal{C}^{e_j}$ is defined as

$$P(\mathcal{C}^{e_j}) \propto \mathbb{I}_{\{\mathrm{DF}(e_j)<\mathrm{R}_{\text{thresh}}\}} \times \Big[\big|\mathrm{Nb}(e_j)\big|\Big]_{\mathrm{R}_{\text{min}}}^{\mathrm{R}_{\text{max}}}, \quad (3)$$

where $\mathrm{Nb}(e) \triangleq \{e' | \ \mathrm{PLen}(e' \leftrightarrow e) < \mathrm{R}_{\text{hop}} \wedge e' \in \mathcal{E}\}$. The term $\mathrm{DF}(\cdot)$ denotes document frequency, $\mathrm{PLen}(e \leftrightarrow e')$ is the length of the shortest undirected path between the two entities, $|\cdot|$ denotes the set size, and $[x]_a^b \triangleq \max(a, \min(x, b))$.

Note, the first part in Eq.(3) is designed to eliminate trivial entities that frequently appear. The second part measures whether an entity can be reached from other entities detected in the same text within R_{hop}-hops, and assigns a higher sampling weight to an entity (e.g., *criminal* in Figure 2) that could more easily be inferred by others. By guiding the model to favor masking deducible but non-trivial entities, this scheme facilitates the MLM ingesting relational knowledge into representation learning. $\mathrm{R}_{\text{hop/thresh/min/max}}$ are hyperparameters that trade off between trivial and undeducible entities.

Finally, it is worth noting, frequently appearing entities that are excluded via $\mathbb{I}_{\{\cdot\}}$ in Eq.(3) can still be masked via 20% random span masking budget, but now with much smaller probabilities.

3.3 Distractor-Suppressed Ranking Task

Empowered by the informative entity-level masks, it is natural to extend the MLM with "negative" entities sampled from the KG, by treating the masked entities as "positive". It has been verified that negative sampling is especially useful for structured knowledge learning in graph embedding approaches (Sun et al., 2019c; Cai and Wang, 2018), but how to effectively integrate negative samples from KGs into MLMs remains open.

Recently, Ye et al. (2019) propose to mask one entity mention in a sentence, and then formulate a multiple-choice QA task (Talmor et al., 2019) for representation learning, by treating the masked sentence as the question, and the masked entity plus its negative samples as answer candidates. However, this model does not quite match the MLM since the model is pre-trained by multiple-choice QA and only one entity can be masked in a sentence.

Here we propose a distractor-suppressed ranking objective that operates on each pair of a masked entity from §3.2 and its negative sample from the KG. The negative one is viewed as a distractor. We use a Transformer encoder to separately produce the embeddings of positive and negative entities using their associated node contents in the KG. We then contrast the positive and negative *entity embeddings*, $\boldsymbol{u}$ and $\boldsymbol{u}'$, against the *masked entity mention's contextualized representation*, $\boldsymbol{v}$, using vector similarity as plausible scores of both entities.

Specifically, given a set of masked entities from §3.2, $\mathcal{E}^p = \{e_1, \ldots, e_m\} \subseteq \mathcal{E}$, with the corresponding entity mentions $\mathcal{C}^{e_j}$, we gather the contextualized representation for each masked entity mention, by mean-pooling over representations of its composite tokens, where the representations are generated by the Transformer encoder of the MLM:

$$\boldsymbol{v_j} = \frac{1}{|\mathcal{C}^{e_j}|} \sum_{k \in \mathcal{C}^{e_j}} \boldsymbol{H}_{:,k}, \ j = 1, \ldots \quad (4)$$

$\boldsymbol{v_j}$ is the resulting contextualized representation for e_j. Since each entity's original mention is invisible to the encoder, $\boldsymbol{v_j}$ is rich in contextual features.

We then sample negative entity(s) from the KG for each $e_j \in \mathcal{E}^p$ and derive a set of positive-negative entity pairs $\{(e_j, e'_j)\}_{j=1}^m$. In particular, given a positive entity e_j, the sampling method randomly selects an entity e'_j from the KG as a negative sample. The sampling favors its sibling entities with the same relation, whose sample weights are twice than the others. This is similar to Ye et al. (2019) and aims to provide strong distractors. Then, another Transformer encoder separately encodes positive and negative entities, which is parameter-tied with the MLM in 3.2 but uses distinct position embeddings. To distinguish entity text coming from KG's node or natural text, we append a special token to the entity text, i.e., $text_j$ = `[CLS]` + e_j + `[ENT]`. We pass $text_j$ into the encoder to obtain the entity embedding for e_j, i.e.,

$$\boldsymbol{u_j} = \mathrm{Pool}(\mathrm{TransformerEnc}(text_j)). \quad (5)$$

Here, $\mathrm{Pool}(\cdot)$ denotes collecting the contextualized embedding from the `[CLS]` as Devlin et al. (2019). The resulting $\boldsymbol{u_j} \in \mathbb{R}^{d_h}$ is an LM-augmented entity embedding for e_j. We apply the same process to e'_j to obtain negative entity embedding $\boldsymbol{u'_j}$.

The procedure above yields a set of tuples, $\{(\boldsymbol{v_j}, \boldsymbol{u_j}, \boldsymbol{u'_j})\}_{j=1}^m$. Finally, a BiLinear layer is used as a parameterized metric for a similarity score between $\boldsymbol{v_j}$ and $\boldsymbol{u_j}$ (or $\boldsymbol{u'_j}$). The score is

$$s_j = \mathrm{BiLnr}(\boldsymbol{v_j}, \boldsymbol{u_j}), \;\; s'_j = \mathrm{BiLnr}(\boldsymbol{v_j}, \boldsymbol{u'_j}), \quad (6)$$
$$\text{where } \mathrm{BiLnr}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \boldsymbol{x}^T \boldsymbol{W} \boldsymbol{y} + b.$$

s_j and s'_j are the plausible scores for positive and negative entities, respectively. The two BiLinear layers used in Eq.(6) are parameter-tied. We then use a margin-based hinge loss to train the MLM with the formulated pairwise ranking task, i.e.,

$$\mathcal{L}_R = \frac{1}{m} \sum_{j=1}^{m} \max(\lambda - s_j + s'_j, 0), \qquad (7)$$

where the margin λ is a hyperparameter.

The proposed distractor-suppressed ranking task has several nice properties. First, only a lightweight BiLinear layer is used to measure the score. Second, training to distinguish positive from negative samples may make the model more effective. Intuitively, two sibling entities in a KG are often assigned with similar distributed representations, but express differently in subtle context; this task helps discriminate them. Lastly, in contrast to the work of Ye et al. (2019), ours is fully compatible with the entity-level MLM training task.

The final loss function to optimize is defined as a combination of the entity-level MLM loss $\mathcal{L}_M$, and the distractor-suppressed ranking loss $\mathcal{L}_R$, with the latter weighted by a hyperparameter γ:

$$\mathcal{L} = \mathcal{L}_M + \gamma \mathcal{L}_R. \qquad (8)$$

3.4 Comparison to Prior Entity-Level MLMs

Our work differs from prior entity-level MLMs, including SpanBERT (Joshi et al., 2019) and Baidu-ERNIE (Sun et al., 2019a,b) in several ways. While the motivation of previous work is moving beyond token to another text unit, our method looks for ways to introduce structured knowledge from KGs into language models. As such, named entities in prior works are recognized via NLP toolkits and the entities are simply masked in random, so relational knowledge unlikely exists among them. In contrast, entities in GLM are linked to a supporting KG, and masking has taken into consideration how an entity interacts with its neighbors in the KG. Similarly for modeling objective, previously proposed objectives, such as span boundary objective (Joshi et al., 2019), aim at learning text semantics as in traditional MLM objectives. By exploiting relational knowledge among the recognized entities, we end up with a ranking task that is specially designed for the proposed entity-level MLM to directly acquire structured information by contrastive learning.

4 Experiments

In this section we demonstrate the effectiveness of GLM on multiple benchmarks. Additional insights are gained with ablation study and other analyses.

Setups. We focus on non-canonicalized commonsense KGs in this work, specifically ConceptNet, and the proposed approach is also applicable to ontological KGs such as Freebase. For training efficiency we use two relatively small free-form corpora. One is the Open Mind Common Sense (OMCS) raw corpus[2] consisting of 800K short sentences. The other is the ARC corpus (Clark et al., 2018) containing 14M unordered, science-related sentences. Both corpora are parsed to have their entities linked to ConceptNet by using an inverted index built with fuzzy matching. In addition, we initialize GLM with either BERT or RoBERTa rather than training from scratch. We choose to match the corresponding baseline model (whether it uses BERT or RoBERTa) in each downstream task. In practice, we can initialize GLM with any state-of-the-art pre-trained bidirectional LM. Experiment code is available at `https://github.com/taoshen58/glm-codes`, and detailed settings about training and testing are provided in Appendix A.

Entity Linking for Commonsense KG. To build a fast and effective graph linking system for commonsense knowledge graph, we first apply tokenization and lemmatization to all the concepts in the graph. The data structure of the resulting phrase vocabulary is optimized as a tree-formatted inverted index to achieve good matching efficiency. During the linking phase, we apply the same tokenization and lemmatization to an input, and execute token-level match starting from every token in the input. Besides inflection-irrelevant property from lemmatization, we also consider flexible token interval to allow extra modifier or article. In case multiple candidate mentions are found, the one with the minimum Levenshtein distance to the un-lemmatized concept is taken as the result.

[2] `https://github.com/commonsense`

Dataset	# Entity	# Rel	# Train	# Dev	# Test
CommonsenseQA	N/A	N/A	9,741	1,221	1,140
SocialIQA	N/A	N/A	33,410	1,954	2,224
WN18RR	40,943	11	86,835	3,034	3,134
WN11	38,696	11	112,581	2,609	10,544
CKBC	78,334	34	100,000	1,200/1,200	2,400

Table 1: Summary statistics of five benchmarks. The first two are multiple-choice question answering tasks. The rest includes one link prediction and two triple classification tasks.

Downstream Tasks. CommonsenseQA (Talmor et al., 2019) and SocialIQA (Sap et al., 2019b) are used to evaluate GLM's performance on natural question answering (QA) task. We also experiment with three knowledge base completion (KBC) tasks: WN18RR (Dettmers et al., 2018), WN11 (Bordes et al., 2013) and commonsense knowledge base completion (Li et al., 2016), to assess whether the proposed approach can benefit graph-related tasks. Their statistics are listed in Table 1. It is worth mentioning, although WordNet (Miller, 1998) is included in ConceptNet, the triples in ConceptNet are never used during GLM training but only raw texts from OMCS (which is a standalone source of ConceptNet and independent of WordNet), so the relation labels in WordNet are never seen by GLM.

Evaluation Metrics. For multiple-choice question answering tasks and triple classification tasks, we use accuracy as the metric. For link prediction task, there are two kinds of metrics: the first includes mean rank (MR) and mean reciprocal rank (MRR), and the second is H@N (namely Hits@N) which means the proportion of correct entities in top N after being sorted w.r.t. predicted confidence. We only report results under the *filtered* setting (Bordes et al., 2013) which removes all corrupted triples appearing in training, dev and test set.

4.1 Question Answering Task Evaluation

CommonsenseQA. Table 2 reports test results of single models from the leaderboard[3] and finetuning GLM. A brief introduction to each approach without reference can be found on the leaderboard. Compared to the previous best model RoBERTa+KE which is trained with an extra in-domain corpus (i.e., OMCS) and uses retrieval during finetuning, our approach achieves 0.8% absolute improvement to deliver a new state-of-the-art result. In addition, GLM based on RoBERTa-large outperforms its corresponding baseline RoBERTa-large by 2.0%.

Method	Dev	Test
Models from pre-trained language model finetuning		
BERT-large (Devlin et al., 2019)	-	56.7
XLNet-large (Yang et al., 2019)	-	62.9
RoBERTa-large (Liu et al., 2019b)	78.5	72.1
Models w/ IR* or extra supervisions during finetuning		
CoS-E (Rajani et al., 2019)	-	58.2
AristoBERTv7 (BERT-large)	-	64.6
DREAM (XLNet-large)	-	66.9
RoBERTa + IR	78.9	72.1
RoBERTa + KE	78.7	73.3
Models w/ further self-supervision tasks + finetuning		
BERT+AMS (Ye et al., 2019)	-	62.2
BERT+OMCS	-	62.5
RoBERTa+CSPT	76.2	69.6
FreeLB-RoBERTa (Zhu et al., 2019)	78.8	72.2
GLM (RoBERTa)†	**79.8**	**74.1**

Table 2: Results on CommonsenseQA for single models."-" denotes unavailable result, and underlined score is the previous best. *IR stands for information retrieval. †GLM is initialized with RoBERTa-large and falls into the last group.

Note, methods using IR (e.g., RoBERTa+KE) must retrieve from Wikipedia during finetuning and inference, which increases the computational overhead significantly. In contrast, methods based on additional self-supervised pre-training are more efficient, but often achieve sub-optimal performance since they lack explicitly retrieved contexts. The proposed GLM falls into the latter high-efficiency group while still outperforms IR-based approaches.

Some works (e.g., RoBERTa+CSPT) find pre-training on triples from a KG can hurt the performance. This evidence supports our hypothesis that pre-training on triples can over-adapt a model to graph-related tasks and limit their generalization.

Our approach is not comparable to Lv et al. (2019) and Lin et al. (2019), the first of which achieves 75.3% accuracy. This is because during finetuning and inference, those methods explicitly find a path from question to answer concept in ConceptNet. This helps filter human-generated distractor answers since they unlikely appear in ConceptNet. In contrast, our method never uses ConceptNet during finetuning and only observes a small subgraph of ConceptNet (about 30%$\sim$40% linked concepts without relations) during pre-training.

SocialIQA. The dataset is built upon the ATOMIC knowledge graph (Sap et al., 2019a) and

[3] www.tau-nlp.org/csqa-leaderboard up to 28th Apr. 2020.

Method	Dev	Test
BERT-large (Devlin et al., 2019)	66.0	64.5
RoBERTa-large (Liu et al., 2019b)	78.2	77.1
McQueen (RoBERTa) (Anonymous)	79.5	78.0
GB-KSI (Anonymous)	77.5	78.1
GLM (RoBERTa)	**79.6**	**78.6**

Table 3: Results on SocialIQA. The results for comparative methods are copied from Sap et al. (2019b) or leaderboard.

Method	WN18RR				
	MR	MRR	H@1	H@3	H@10
TransE[1]†	2300	.243	.043	.441	.532
R-GCN[2]†	6700	.123	.080	.137	.207
ConvE[3]†	4464	.456	.419	.470	.531
ConvKB[4]†	1295	.265	.058	.445	.558
RotatE[5]	3340	.476	.428	.492	.571
QuatE[6] + TypeCons[8]	2314	**.488**	**.438**	**.508**	.582
KG-BERT[7]$_{\text{BERT-base}}$	97	.216	.041	.302	.524
KG-BERT$_{\text{GLM(BERT-base)}}$	86	.273	.086	.344	.587
+ TypeCons[8]	**42**	.370	.188	.473	**.728**

Table 4: Link prediction results on the WN18RR benchmark. †Numbers are copied from Nathani et al. (2019). [1](Bordes et al., 2013), [2](Schlichtkrull et al., 2018), [3](Dettmers et al., 2018), [4](Nguyen et al., 2018), [5](Sun et al., 2019c), [6](Zhang et al., 2019a), [7](Yao et al., 2019), [8](Krompaß et al., 2015).

Method	Dev	Test
TransE (Bordes et al., 2013)	-	75.9
DistMult-HRS (Yang et al., 2015)	-	88.9
DOLORES (Wang et al., 2018)	-	87.5
ConvKB (Nguyen et al., 2018)	-	87.6
AATE (An et al., 2018)	-	88.0
KG-BERT$_{\text{BERT-base}}$ (Yao et al., 2019)	-	93.5
KG-BERT$_{\text{GLM(BERT-base)}}$	**94.8**	**94.0**

Table 5: Test accuracy on WN11 triple classification task.

Method	Dev2	Test
Bilinear AVG (Li et al., 2016)	90.3	91.7
Bilinear AVG + Data (Li et al., 2016)	91.8	92.5
KG-BERT$_{\text{BERT-base}}$	92.9	93.2
KG-BERT$_{\text{BERT-base}}$ + Data	92.3	92.4
KG-BERT$_{\text{GLM(BERT-base)}}$	93.0	93.5
KG-BERT$_{\text{GLM(RoBERTa-large)}}$	**94.7**	**94.6**

Table 6: Test accuracy on CKBC triple classification task..

focuses on reasoning about people's actions and their social implications. It thus serves as an out-of-domain evaluation task for GLM trained using ConceptNet. Here, "out-of-domain" refers to out-of-domain KG: the model is trained with ConceptNet while SocialIQA is built upon ATOMIC. Moreover, the corpora used in continual pre-training are also out-of-domain for SocialIQA. Similar to CommonsenseQA, this task is formulated as a multiple-choice QA problem. The evaluation results listed in Table 3 demonstrate that our approach also achieves state-of-the-art performance on this out-of-domain dataset.

4.2 Graph-Related Task Evaluation

For this set of tasks we follow KG-BERT (Yao et al., 2019) that finetunes MLMs over a concatenation of a triple's head, relation, and tail, followed by an MLP to compute a score denoting if the triple is plausible. Since BERT-base model is used in KG-BERT, for fair comparison we train a GLM from BERT-base, denoted as "GLM (BERT-base)" and finetune it on KBC task following KG-BERT.

WordNet Knowledge Base Completion. Table 4 lists test results for the WN18RR link prediction task. GLM significantly outperforms KG-BERT and sets state-of-the-art or competitive results, which verifies the model's ability to retain relational knowledge. Further when type constraint is applied, "GLM+TypeCons" achieves overwhelming performance, especially in MR and Hits@10.

Test results for the WN11 triple classification task are listed in Table 5. Consistent with the results on WN18RR, finetuning our model outperforms translation-based graph embedding models and convolution-based methods, and improves state-of-the-art accuracy by 0.5%.

Commonsense Knowledge Base Completion. Lastly, we evaluate our approach on the CKBC task, which should directly benefit from commonsense knowledge. Since CKBC is derived from the OMCS corpus, for fair comparison, we provide a baseline model with equivalent training data ("+ Data" in Table 6). In addition, we remove raw sentences belonging to CKBC's test set from our GLM pre-training corpora to avoid data leakage.

Results in Table 6 show our approach outperforms KG-BERT baseline even when the latter is equipped with equivalent data (increasing training triples from 100K to 600K). There are two possible reasons why performance actually drops with more data: 1) the training triples are sorted w.r.t annotated confidence, so additional triples have a lower quality and may introduce noise; 2) more negative sampling must be done with more training triples, which introduces more false negative examples (from 1.25% to 2.42% by our observation).

Method	F1 Score
BERT-large (Davison et al., 2019)	78.8
BERT-large (our implementation)	77.1
GLM (BERT-large)	**83.4**

Table 7: Zero-shot evaluation following Davison et al. (2019).

Method	Dev Accu
BERT-large (w/o GLM continual pre-training)	65.5
GLM (BERT-large)	69.0
◇ w/o KG-guided Masking	68.6
◇ w/o Ranking	68.1
◇ w/o Ranking & Entity Mask (= BERT-large)	67.7
◇ Ranking→SBO	66.8
◇ Ranking→SBO, Entity Mask→Span	68.2

Table 8: Ablation study on CommonsenseQA dev set. Note, "SBO" denotes span boundary objective (Joshi et al., 2019) and "→" stands for module replacement.

4.3 Zero-Shot Evaluation

To explore whether GLM can indeed learn the structured information from raw text, we conduct a zero-shot evaluation on CKBC by following Davison et al. (2019). We re-train the GLM (BERT-large) on new corpora in which all raw texts containing the CKBC testing pairs are discarded. We re-implement coherency ranking and estimate PMI (Davison et al., 2019), and the results are shown in Table 7. The GLM significantly outperforms its baseline, which demonstrates the capability of retaining structured information in a language model.

4.4 Ablation Study

To evaluate the effectiveness of each component in GLM, we conduct an ablation study in Table 8 via pre-training language models with different setups and then finetuning on CommonsenseQA.

When KG-guided entity masking introduced in §3.2 is replaced with random entity masking during GLM pretraining, 0.4% accuracy drop is observed when the model is subsequently evaluated on CommonsenseQA dev set. If we set γ in Eq.(8) to zero when pre-training GLM, this leads to 0.9% accuracy drop. When both entity-level masking and distractor-suppressed ranking are removed, the setting becomes equivalent to performing continual pre-training of BERT-large on our corpora. This helps us separate the contribution of extra corpora from the proposed approach. Compared with BERT-large baseline, we observe a 2.2% (65.5% to 67.7%) accuracy improvement contributed by the corpora. Thus GLM yields an extra 1.3% (67.7% to 69.0%) improvement by better data exploitation.

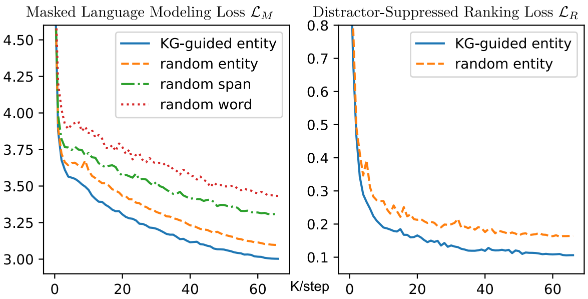

Figure 3: Dev loss in pre-training phase for different training or masking schemes. Note only entity-level masking has $\mathcal{L}_R$.

When we replace distractor-suppressed ranking with span boundary objective (Joshi et al., 2019), a significant performance decrease (-2.2%) is observed. Further replacing our entity-level masking with random span masking, however, only loses 0.8% in accuracy. It is worth noticing that the latter setup is equivalent to continual pre-training with SpanBERT (Joshi et al., 2019) on our corpora. In line with SpanBERT's conclusion that the performance of linguistic masking is not consistent, our KG-guided entity-level MLM (i.e., GLM w/o the ranking task) is worse than random span with SBO (68.1% vs. 68.2%). This suggests that objective to mask and objective to learn need to be paired, and linguistic masking can be useful if equipped with appropriate learning objective (e.g., our distractor-suppressed ranking task) during pre-training.

4.5 Analysis of Training & Masking Schemes

Given the same dev set of texts with the same masked tokens, Figure 3 compares different learning/masking schemes by plotting $\mathcal{L}_M$ (left) and $\mathcal{L}_R$ (right) defined in Eq.(8) w.r.t pre-training steps. It is observed our KG-guided entity masking is more efficient than three other masking schemes, including random entity masking (Sun et al., 2019a), random span masking (Joshi et al., 2019), and random whole-word masking (Devlin et al., 2019).

Table 9 lists a few example sentences with masked tokens highlighted according to the corresponding masking scheme. Both KG-guided and random entity masking can mask informative chunks and long-term dependency needs to be modeled in order to infer the masked tokens. In contrast, random span or word masking is likely to mask tokens that can be easily inferred from local context – a much simpler task. Furthermore, our KG-guided entity masking tends to select more informative phrases compared to random entity masking.

No.	Masked Text with different masking schemes
1	something you need to *do before* ***you*** get up **early** is **set** an alarm ← (leave the office)
2	you would talk **with** someone far ***away*** *because you* want keep in touch ← (meet strangers)
3	if you want **to** drill *a* ***hole then*** you **should** carefully plan **where** you will drill it ← (cut of beef)

Table 9: Case study for different masking schemes. Note that 1) underline: KG-guided entity masking; 2) underline wave: random entity masking; 3) *italic*: random span masking; and 4) **bold**: random whole-word masking. Text in parenthesis is a negative entity sampled for KG-guided entity masking.

4.6 Error Analysis

Compared with previous pre-trained LMs (e.g., BERT and RoBERTa), some limitations are found in our current models pre-trained on OMCS and ARC corpora, which mainly fall into three aspects:

- **Over masking:** Compared to random masking, our KG-guided masking scheme is prone to masking all key parts of a sentence, which leaves little room for MLM task.

- **Short context:** Since our employed pre-training corpora (i.e., OMCS and ARC) consist of just single, unordered sentences, information that spans across multiple sentences is not encoded effectively. When downstream tasks rely heavily on adjacent sentences, fine-tuning our model yields inferior performance. This can be empirically verified by the performance drop on the datasets (e.g., HellaSWAG in Appendix B) that involves long contexts.

- **Pipeline model:** Same as any other method aiming to integrate LMs with KG, a linking system is first applied to detect entities in text, which inevitably suffers from graph sparsity and leads to error propagation. However, our method is less sensitive to such errors compared with the methods that link entities during both finetuning and inference.

5 Related Work

Our work is related to Baidu-ERNIE (Sun et al., 2019a) and SpanBERT (Joshi et al., 2019), which both extend token level masking to the span level. For example, Baidu-ERNIE does so to improve the model's knowledge learning, using uniformly random masking for phrases and entities. As summarized in §1, existing methods for integrating knowledge into pre-trained LMs can be coarsely categorized into two classes. For example, Peters et al. (2019) retrieve entities' embeddings according to the similarity between Transformer's hidden states and pre-trained graph embeddings, then treat the retrieved embeddings as extra inputs to the next layer. In contrast, Bosselut et al. (2019) directly finetune a pre-trained LM on partially-masked triples from a KG, aiming at commonsense KBC tasks. This however limits the applications in KG-based tasks rather than natural language processing tasks like question answering (Shen et al., 2019a), sentiment analysis (Li et al., 2019), sentence classification (Shen et al., 2019b), etc. And our work is also related to using negative samples for effective learning (Cai and Wang, 2018). Moreover, our work is distinct from the works combining knowledge graph with text information via joint embedding (Wang et al., 2014; Toutanova et al., 2015; Yamada et al., 2016). They usually use the texts containing co-occurrence of entities to enrich the graph embeddings, which are specially designed for graph-related tasks. (Full in Appendix C.)

6 Conclusion

In this work, we aim at equipping pre-trained LMs with structured knowledge via self-supervised tasks. Building on entity-level MLMs, we propose an entity masking scheme under KG's guidance. It masks informative mentions and facilitates learning structured knowledge in free-form text. Moreover, we propose a distractor-suppressed ranking objective to utilize negative samples from KG as distractors for effective training. Experiments show finetuning our KG-guided pre-trained MLMs yields improved performance on related downstream tasks.

In the future, instead of pre-training on sentences, we will leverage raw text at passage or document level to alleviate the performance degeneration brought by short context. Moreover, we will use a combination of commonsense and ontological KGs, and large-scale corpora (e.g., Common Crawl) to pre-train an MLM from scratch, which we expect to benefit a wide range of tasks.

Acknowledgement

The authors would like to thank Siqi Sun for valuable suggestions about the model, Alon Talmor and Jonathan Herzig for evaluating CommonsenseQA test accuracy, and anonymous reviewers for their insightful and constructive feedback.

References

Bo An, Bo Chen, Xianpei Han, and Le Sun. 2018. Accurate text-enhanced knowledge graph representation learning. 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 745–755.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5184–5193. Association for Computational Linguistics.

Kurt D 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 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2787–2795.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. 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 4762–4779.

Liwei Cai and William Yang Wang. 2018. KBGAN: adversarial learning for knowledge graph embeddings. 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 1470–1480.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pre-trained models. 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 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 1811–1818.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, pages 640–655.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense Knowledge Base Completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Zheng Li, Xin Li, Ying Wei, Lidong Bing, Yu Zhang, and Qiang Yang. 2019. Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning. 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 4590–4600, Hong Kong, China. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-BERT: enabling language representation with knowledge graph. *CoRR*, abs/1909.07606.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. 2019. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. *CoRR*, abs/1909.05311.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2019. Exploiting structural and semantic context for commonsense knowledge base completion. *CoRR*, abs/1910.02915.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 1003–1011.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *CoRR*, abs/1909.08855.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. 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 4710–4723.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 327–333.

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.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. 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.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: Multilingual entity linking by neural type system evolution. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5406–5413.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. 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 4932–4942.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Social IQa: Commonsense reasoning about social interactions. 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 4462–4472, Hong Kong, China. Association for Computational Linguistics.

Michael Sejr Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 593–607.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019a. Multi-task learning for conversational question answering over a large-scale knowledge base. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2442–2451. Association for Computational Linguistics.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019b. Tensorized self-attention: Efficiently modeling pairwise and global dependencies together. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1256–1266. 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 Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4444–4451.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019a. ERNIE: enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019b. ERNIE 2.0: A continual pre-training framework for language understanding. *CoRR*, abs/1907.12412.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019c. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509.

Haoyu Wang, Vivek Kulkarni, and William Yang Wang. 2018. DOLORES: deep contextualized knowledge graph embeddings. *CoRR*, abs/1811.00147.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *CoRR*, abs/2002.01808.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1591–1601.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 250–259.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yi Yang and Ming-Wei Chang. 2015. S-MART: novel tree-based structured learning algorithms applied to tweet entity linking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 504–513.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *CoRR*, abs/1908.06725.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? 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 4791–4800.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019a. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 2731–2741.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. ERNIE: Enhanced Language Representation with Informative Entities. 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 1441–1451.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2019. Freelb: Enhanced adversarial training for language understanding. *CoRR*, abs/1909.11764.

A Implementation Details

Pre-trained language models implemented by Huggingface[4] are adapted for our models.

Pre-Training Hyperparameters. For continual pre-training, we do not tune hyperparameters due to the computational cost. Only a limited set of hyperparameters is tried according to empirical intuitions. Currently $R_{hop/min/max}$ in Eq.(3) of main paper are set to 3/1.0/2.0 respectively, and R_{thresh} aims to filter out entities with top 5% document frequency, thus varies with corpora. We set λ in Eq.(7) of main paper to be 1.0 and γ in Eq.(8) of main paper to be 0.2. The continual pre-training runs for 5 epochs, with a batch size of 128, a learning rate of 3e-5/1e-5 (BERT/RoBERTa), learning warmup proportion of 10%/5% (BERT/RoBERTa) and weight decay of 0.01. For both BERT and RoBERTa, max sequence length is set to be 80 and 20 for sentence-level encoding and entity embedding respectively. The masking proportion is lifted from 15% to 20% without tuning compared with BERT and RoBERTa. An intuition is that our model is initialized with well-trained language models (e.g., BERT), a slightly larger masking proportion could hold the entity with longer text span, and make the learning more efficient.

Finetuning Hyperparameters. During finetuning on downstream tasks, we conduct grid search for hyperparameters, including batch size, number of epochs/steps, learning rate, which are summarized in Table 10. As for the number of learnable parameters, the models heavily depends on the Transformer encoder since only light-weight BiLinear layer or neural classifier is introduced during finetuning. In particular, BERT-base, RoBERTa-base, BERT-large and RoBERTa-large have $\sim$ 110M, 130M, 340M and 360M parameters respectively.

Details about KG-Guided Entity Masking. In addition to finding informative and non-trivial masks, the KG-guided entity masking method can be used to filter the corpus, since if a piece of text (e.g., sentence or passage) contains only trivial and undeducible entities, the text may hardly contribute to model learning. With this strategy, most ($\sim$ 90%) sentences in ARC corpus were filtered out, which leads to a very efficient training for our models. Training our model based on BERT-large or RoBERTa-large only costs about 1 day on single V100 GPU with mixed float precision, in total about 70K steps.

Fair Comparison on WordNet KBC. There are two aspects to prevent data leakage from ConceptNet to WN18RR. First, ConceptNet is derived from OMCS, WordNet, Wiktionary, etc., and each source is independent of others. Our method only uses OMCS raw text as the training corpus and has no access to WordNet. On the other hand, even though entire ConceptNet is used to guide entity masking/sampling, unlike traditional methods taking KG's triples as training examples, our method doesn't use the relation labels but only the entities/concepts. Since WordNet's entities/concepts are also provided in the training set of WN18RR/WN11, there is no risk of leakage when testing.

Model Implementation for KBC. For CKBC dataset, we find the performance is poor when we directly finetune either BERT-base or our approach on the concatenated triples, i.e., "`[CLS]` HEAD `[SEP]` *Rel* `[SEP]` TAIL `[SEP]`". Hence, we follow Davison et al. (2019) to transform the triples to natural language sentences, and then use these sentences as input to finetune the pre-trained LMs. For WN18RR, we directly use the data processed by Yao et al. (2019), in which a description sen-

[4] https://github.com/huggingface/transformers

Hparam	GLM Model	CQA	SocialIQA	WN18RR	WN11	CKBC
Batch Size	BERT	{8, $\underline{12}$, 16}	/	{$\underline{32}$}	{$\underline{32}$, 48, 64}	{24, $\underline{32}$}
	RoBERTa	{8, $\underline{12}$, 16}	{12, $\underline{16}$}	/	/	{24, $\underline{32}$}
# Steps/Epochs	BERT	{2800, $\underline{3400}$, 4000}	/	{$\underline{5}$}	{3, $\underline{5}$}	{$\underline{6}$}
	RoBERTa	{$\underline{2800}$, 3600}	{3,$\underline{4}$}	/	/	{$\underline{6}$}
Learning Rate	BERT	{3e-5, $\underline{5e-5}$, 7e-5}	/	{$\underline{5e-5}$}	{3e-5, $\underline{5e-5}$, 7e-5}	{3e-5, $\underline{5e-5}$, 7e-5}
	RoBERTa	{1e-5, $\underline{8e-6}$}	{1e-5, $\underline{8e-6}$}	/	/	{$\underline{1e-5}$, 2e-5}
Time/Epoch	Base Model	3m	/	28m	9m	8m
	Large Model	6m	19m	/	/	25m

Table 10: Candidate values for hyperparameter search during finetuning. Note "# steps" is shown for CQA and "# epochs" for the others. The hyperparameters with underline can lead to best-performing models. Each hyperparameter grid search is conducted four times with different random seed. The time data is collected on single NVIDIA V100.

tence is attached to each entity/phrase. For WN11, we follow KG-BERT to directly concatenate triples and then use them to finetune the pre-trained LMs.

Knowledge Graph and Entity Linking. We aim at enhancing the pre-trained language models with commonsense structured knowledge, so we employ ConceptNet (Speer et al., 2017) as the backend knowledge graph. Since ConceptNet is a multi-lingual knowledge graph, we first filter out all the triples which include non-English items. In addition, we treated the KG as an undirected graph when identifying entity's mutual reachability. As for entity linking, there are many mature entity linking systems for ontological or factoid KGs, such as S-MART (Yang and Chang, 2015), DBpeida Lookup, and DeepType (Raiman and Raiman, 2018). However, for a commonsense KG whose content consists of non-canonicalized or free-form texts, there is no such a system to complete its entity linking. Therefore, we build an efficient inverted index out of lemma-based fuzzy matching as our entity linking system.

B More Experiments

PhysicalIQA. In addition to CommonsenseQA and SocialIQA shown in the main paper, a dataset named PhysicalIQA[5] is also used to evaluate our method. It is also regarded as an out-of-domain dataset compared with our training corpora. However, our implemented code base cannot reproduce the state-of-the-art results that are achieved by RoBERTa-large finetuning, possibly due to different pre-processing and feeding strategies for pre-trained LMs, e.g., special token, concatenation scheme, representation gathering method (Mitra et al., 2019). Hence, we only report re-implemented results with the same network structure, same data-preprocessing method, same random seed and same hyperparameter grid search, for fair comparison. The accuracy on dev set is 78.7% and 80.2% for RoBERTa-large baseline and GLM (RoBERTa) respectively, which further demonstrates the effectiveness of our approach on out-of-domain datasets.

HellaSWAG. We also try to apply our approach to HellaSWAG[6] (Zellers et al., 2019). It is a plausible inference task and requires reasoning over linguistic context and external knowledge. The task is to choose one plausible ending from four candidates. Same as PhysicalIQA, we only fairly report the accuracy on dev set for a fast comparison. With our implementation, finetuning RoBERTa-large on this dataset achieves 84.1% dev accuracy which is much higher than the best dev accuracy (83.5%) on leaderboard. However, finetuning our approach achieves 83.9% accuracy, which is slightly worse than the baseline. We notice that examples in HellaSWAG frequently have multiple consecutive sentences for inference, thus our model trained on single, unordered sentences may only achieve suboptimal performance. On the other hand, this is another out-of-domain dataset which may not benefit from our training knowledge graph and corpora.

C Related Work (extended)

This work is in line with Baidu-ERNIE (Sun et al., 2019a) and SpanBERT (Joshi et al., 2019) which replace word-level mask (Devlin et al., 2019) with span-level one for knowledge information and long-term dependency. In particular, Baidu-ERNIE uses

[5] https://leaderboard.allenai.org/physicaliqa/submissions/public

[6] https://leaderboard.allenai.org/hellaswag

uniformly random masking for phrases and entities, whereas SpanBERT directly masks out token spans sampled under geometric distribution.

The work of Petroni et al. (2019) finds that, without finetuning, pre-trained LMs (e.g., BERT) contains relational knowledge competitive with traditional NLP methods with oracle knowledge. Nevertheless, how to integrate the oracle knowledge into the pre-trained LMs for further performance improvement remains an open question.

As briefly summarized in the introduction of the main paper, existing methods can be coarsely categorized into two classes. For the first class, those methods retrieve a KG subgraph or/and pre-trained graph embeddings via entity linking during finetuning and inference. K-BERT (Liu et al., 2019a) retrieves a path from KG as description for each detected entity in text, and inserts such description into input sequence to Transformer encoder with carefully designed attention mask and position embedding. KnowBert (Peters et al., 2019) and THU-ERNIE (Zhang et al., 2019b) first retrieve the detected entities' embeddings from pre-trained graph embeddings (Bordes et al., 2013), and then treat these retrieved embeddings as extra inputs for each layer of Transformer encoder. Lin et al. (2019) and Lv et al. (2019) aim to solve commonsense multiple-choice QA problem. They retrieve a graph path from entities detected in question to each answer entry, and then encode (e.g., via LSTM) these paths as heterogeneous representations for higher-level modules.

The second class of methods uses contextualized representations from pre-trained LMs to enrich graph embeddings and thus alleviates graph sparsity issues. COMET (Bosselut et al., 2019) finetunes pre-trained LM on partially-masked KG triples, which aims at commonsense knowledge graph completion tasks. Malaviya et al. (2019) perform transfer learning from pre-trained language models to knowledge graphs for enhanced contextualized representation of the knowledge. KG-BERT (Yao et al., 2019) directly concatenates the head, relation and tail of a triple, and finetunes pre-trained LMs on such data with binary classification objective, i.e., whether a triple is correct or not.

More recently, K-Adapter (Wang et al., 2020) keeps the pre-trained LMs unchanged and proposes two neural adapters that are trained with relation classification and dependency parsing respectively, based on the LM's hidden states. During finetuning, these two adapters can benefit relevant downstream tasks, e.g., entity typing and QA.

How to generate and utilize negative samples is important for learning graph embeddings and structured knowledge (Sun et al., 2019c; Ye et al., 2019). For example, KBGAN (Cai and Wang, 2018) uses a knowledge graph embedding model as negative sample generator to assist the training of the desired model, which acts as the discriminator in GANs. Rather than a standalone generator, self-adversarial sampling (Sun et al., 2019c) generates negative samples according to the current entity or relation embeddings. BERT-AMS (Ye et al., 2019) and our proposed ranking task share a similar motivation that the model is able to effectively learn the structured knowledge from negative samples, but they differ in the task designs. BERT-AMS builds a multiple-choice question answering task for utilizing negative samples, which imitates the developing procedure of CommonsenseQA (Talmor et al., 2019) and aims to improve performance on that particular dataset. In contrast, our approach is more general and formulates a ranking task along with the entity-level masked language modeling objective for pre-training knowledge-aware LMs.

This work differs from the works combining knowledge graph with text information via joint embedding (Yamada et al., 2016). They usually use the texts containing co-occurrence of entities to enrich the graph embeddings, which are specially designed for graph-related tasks. For example, Wang et al. (2014) embed entities from KG and the entities' text contents in the same latent space, however, regardless of textual co-occurrences and their textual relations in natural language corpus. Further taking into account the sharing of substructure in the textual relations of a large-scale corpus, Toutanova et al. (2015) apply a CNN to the lexicalized dependency paths of the textual relation, for an augmented relation representation. The representation can be fed into any previous graph embedding approach for enhanced performance on KBC. We share similar inspirations when utilizing the texts containing entity co-occurrences and embedding entities' text contents into latent space. But beyond the shallow joint embeddings, our work takes advantage of pre-trained MLMs and equips them with structured knowledge via two self-supervised objectives built upon raw text. Hence it can produce generic text representations to benefit various downstream tasks.