Empowering Language Models with Knowledge Graph Reasoning for Question Answering

Anonymous Author(s)
Affiliation
Address
email

Abstract

Answering open-domain questions requires world knowledge about in-context entities. As pre-trained Language Models (LMs) lack the power to store required knowledge, external knowledge sources, such as knowledge graphs, are often used to augment LMs. In this work, we propose knOwledge REasOning empowered Language Model (OREOLM), which consists of a novel Knowledge Interaction Layer that can be flexibly plugged into existing Transformer-based LMs to interact with a differentiable Knowledge Graph Reasoning module collaboratively. In this way, LM guides KG to walk towards the desired answer, while the retrieved knowledge improves LM. By adopting OREOLM to RoBERTa and T5, we show significant performance gain, achieving state-of-art results in the *Closed-Book* setting. The performance enhancement is mainly from the KG reasoning's capacity to infer missing relational facts. In addition, OREOLM provides reasoning paths as rationales to interpret the model's decision.

1 Introduction

Open-Domain Question Answering (ODQA), one of the most knowledge-intensive NLP tasks, requires QA models to infer out-of-context knowledge to the given single question. Following the pioneering work by Chen et al. (2017), ODQA systems often assume to access an external text corpus (e.g., Wikipedia) as an external knowledge source. Due to the large scale of such textual knowledge sources (e.g., 20GB for Wikipedia), it cannot be encoded in the model parameters. Therefore, most works retrieve relevant passages as knowledge and thus named *Open-Book* models (Roberts et al., 2020), with an analogy of referring to textbooks during an exam. Another line of *Closed-book* models (Roberts et al., 2020) assume knowledge could be stored implicitly in parameters of Language Models (LM, e.g. BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020)). These LMs directly generate answers without retrieving from an external corpus and thus benefit from faster inference speed and simpler training. However, current LMs still miss a large portion of factual knowledge (Pörner et al., 2020; Lewis et al., 2021a), and are not competitive with *Open-Book* models.

To improve the knowledge coverage of LM, one natural choice is to leverage knowledge stored in Knowledge Graph ($\mathcal{KG}$, e.g. FreeBase (Bollacker et al., 2008) and WikiData (Vrandecic and Krötzsch, 2014)), which explicitly encodes world knowledge via relational triplets between entities. There are several good properties of $\mathcal{KG}$: 1) a $\mathcal{KG}$ triplet is a more abstract and compressed representation of knowledge than text, and thus $\mathcal{KG}$ could be stored in memory and directly enhance LM without using an additional retrieval model; 2) the structural nature of $\mathcal{KG}$ could support logical reasoning (Ren et al., 2020) and infer missing knowledge through high-order paths (Lao et al., 2011; Das et al., 2018). Taking the question "what cheese is used to make the desert cannoli?" as an example, even if this relational fact is missing in $\mathcal{KG}$, we could still leverage high-order relationships, e.g., both Ricotta Cheese and Cannoli are specialties in Italy, to infer the answer "Ricotta Cheese."

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

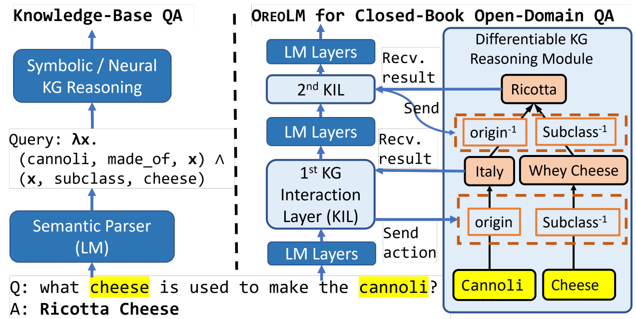


Figure 1: An Illustrative figure of OREOLM. Compared with previous KBQA systems that stack reasoner on top of LM, OREOLM enables interaction between the two.

In light of the good properties of $\mathcal{KG}$, there are several efforts to build Knowledge Base Question Answering (KBQA) systems. As is illustrated in Figure 1(a), most KBQA models use LM as a parser to map textual questions into a structured form (e.g., SQL query or subgraph), and then based on $\mathcal{KG}$, the queries could be executed by symbolic reasoning (Berant et al., 2013) or neural reasoning (e.g. Graph Neural Networks) (Sun et al., 2019) to get the answer. Another recent line of research Verga et al. (2021); Yu et al. (2022b) tries to encode the knowledge graph as the *memory* into LM parameters. However, for most methods discussed above, LM is not interacting with $\mathcal{KG}$ to correctly understand the question, and the answer is usually restricted to a node or edge in $\mathcal{KG}$.

In this paper, we propose knOwledge REasOning empowered Language Model (OREOLM), a model architecture that can be applied to Transformer-based LMs to improve *Closed-Book* ODQA. As is illustrated in Figure 1(b), the key component is the Knowledge Interaction Layers (KIL) inserted amid LM layers, which is like cream filling within two waffles, leading to our model's name OREO. KIL interacts with a $\mathcal{KG}$ reasoning module, in which we maintain different reasoning paths for each entity in the question. We formulate the retrieval and reasoning process as a contextualized *random walk* over the $\mathcal{KG}$, starting from the in-context entities. Each KIL is responsible for one reasoning step. It first predicts a relation distribution for every in-context entity, and then the $\mathcal{KG}$ reasoning module traverses the graph following the predicted relation distribution. The reasoning result in each step is summarized as a weighted averaged embedding over the retrieved entities from the traversal.

By stacking T layers of KIL, OREOLM can retrieve entities that are T-hop away and help LM to answer open questions that require out-of-context knowledge or multi-hop reasoning. The whole procedure is fully differentiable, and thus OREOLM learns and infers in an end-to-end manner. We further introduce how to pre-train OREOLM over unlabelled Wikipedia corpus. In addition to the salient entity span masking objective, we introduce two self-supervised objectives to guide OREOLM to learn better entity and relation representations and how to reason over them.

We test OREOLM with RoBERTa and T5 as our base LMs. By evaluating on several single-hop ODQA datasets in *closed-book* setting, we show that OREOLM outperforms existing baselines with fewer model parameters. Specifically, OREOLM helps more for questions with missing relations in $\mathcal{KG}$, and questions that require multi-hop reasoning. We further show that OREOLM can serve as a backbone for *open-book* setting and achieves comparable performance compared with the state-of-the-art QA systems with dedicated design. In addition, OREOLM has better interpretability as it can generate reasoning paths for the answered question and summarize general rules to infer missing facts.

This key contributions are as follows: (1) We propose OREOLM to integrate symbolic knowledge graph reasoning with neural LMs. Different from prior works, OREOLM can be seamlessly plugged into existing LMs. (2) We pretrain OREOLM with RoBERTa and T5 to on the Wikipedia corpus. OREOLM can bring significant performance gain on ODQA. (3) OREOLM offers interpretable reasoning paths for answering the question and high-order reasoning rules as rationales.

2 Methodology

Preliminary We denote a Knowledge Graph $\mathcal{KG} = \big(\mathcal{E}, \mathcal{R}, \mathcal{A} = \{A_r\}_{r\in\mathcal{R}}\big)$, where each $e \in \mathcal{E}$ and $r \in \mathcal{R}$ is entity node and relation label. $A_r \in \{0,1\}^{|\mathcal{E}|\times|\mathcal{E}|}$ is a sparse adjacency matrix

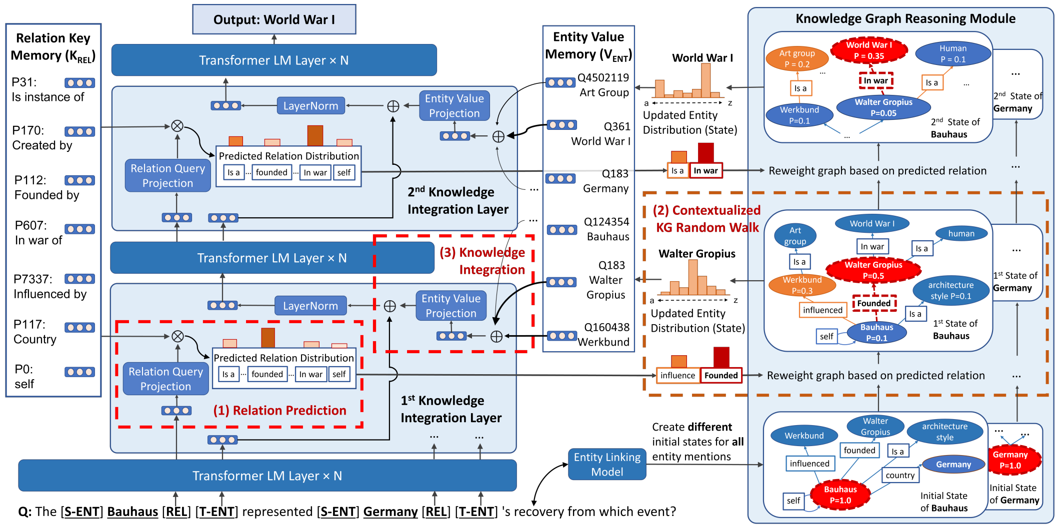


Figure 2: **Model architecture of OREOLM**. Three key procedures are highlighted in red dotted box: 1) **Relation Prediction** (Sec. 2.1.1): Knowledge Interaction Layers (KIL) predicts relation action for each entity mention. 2) **One-step State Transition** (Sec. 2.1.2): Based on the predicted relation, $\mathcal{KG}$ re-weights each graph and conduct contextualized random walk to update entity distribution state. 3) **Knowledge Integration** (Sec. 2.2): An weighted aggregated entity embedding is added into a placeholder token as retrieved knowledge.

indicating whether relation r holds between a pair of entities. The task of knowledge graph reasoning aims at answering a factoid query $(s, r, ?)$, i.e., which target entity has relation r with the source entity s. If $\mathcal{KG}$ is complete, we could simply get answers by checking the adjacency matrix, i.e., $\{\forall t : A_r[s,t] = 1\}$. For incomplete $\mathcal{KG}$ where many relational facts are missing, path-based reasoning approaches Lao et al. (2011); Xiong et al. (2017); Das et al. (2018) have been proposed to answer the one-hop query via finding multi-hop paths. For example, to answer the query $(s, \text{Mother}, ?)$, a path $s \xrightarrow{\text{Father}} j \xrightarrow{\text{Wife}} t$ could reach the target answer t. In this paper we try to integrate symbolic $\mathcal{KG}$ reasoning into neural LMs and help it deal with ODQA problems.

Overview of OREOLM We illustrate the overall architecture of OREOLM in Figure 2. All the light blue blocks are our added components to support $\mathcal{KG}$ reasoning, while the dark blue Transformer layers are knowledge-injected LM. The key component of OREOLM for conducting $\mathcal{KG}$ reasoning is the Knowledge Interaction Layers (KIL), which are added amid LM layers to enable deeper interaction with the $\mathcal{KG}$.

Given a question q = "The Bauhaus represented Germany's recovery from which event?", QA model needs to extract knowledge about all n in-context entity mentions $M = \{m_i\}_{i=1}^n$, e.g., the history of "Germany" at the time when "Bauhaus" is founded, to get the answer a = "World War I". Such open-domain Q&A can be abstracted as $P(a|q, M)$. Starting from each mentioned entity m_i, we desire the model to learn to walk over the graph to retrieve relevant knowledge and form a T-length reasoning path for answering this question, where T is a hyper-parameter denote the longest reasoning path required to answer the questions. We define each reasoning path starting from the entity mention m_i as a chain of entities (states) random variables $\rho_i = \{e_i^t\}_{t=0}^T$, where each mentioned entity is the initial state, i.e., $e_i^0 = m_i$. The union of all paths for this question is defined as $\boldsymbol{\varrho} = \{\rho_i\}$, which contains the reasoning paths from each mentioned entity to answer the question.

OREOLM factorizes $\boldsymbol{P}(a|q, M)$ by incorporating possible paths $\boldsymbol{\varrho}$ as a latent variable, yielding:

$$\boldsymbol{P}(a|q, M) = \sum\nolimits_{\boldsymbol{\varrho}} \boldsymbol{P}(\boldsymbol{\varrho}|q, \{m_i\}_{i=1}^n) \cdot \boldsymbol{P}(a|q, M, \boldsymbol{\varrho}) = \sum\nolimits_{\boldsymbol{\varrho}} \Big(\prod_{i=1}^{n} \boldsymbol{P}(\rho_i|q, m_i) \Big) \cdot \boldsymbol{P}\Big(a|q, \{m_i, \rho_i\}_{i=1}^n\Big)$$

$$= \sum\nolimits_{\boldsymbol{\varrho}} \Big(\prod_{i=1}^{n} \prod_{t=1}^{T} \underbrace{\boldsymbol{P}(e_i^t|q, e_i^{<t})}_{\mathcal{KG}\text{ Reasoning (2.1)}} \Big) \underbrace{\boldsymbol{P}\Big(a|q, \{e_i^{0:T}\}_{i=1}^n\Big)}_{\text{knowledge-injected LM (2.2)}}$$

We assume (1) reasoning paths starting from different entities are generated independently; and (2) reasoning paths can be generated autoregressively. In this way, the QA problem can be decomposed into two entangled steps: 1) $\mathcal{KG}$ Reasoning, which autoregressively walks through the graph to get a path ρ_i starting from each entity mention m_i; and 2) knowledge-injected LM, which benefits from the reasoning paths to obtain the out-context knowledge for answer prediction.

The relational path ρ_i in $\mathcal{KG}$ Reasoning requires the selection of next entity e_i^t at each step t. We further decompose it into two steps: 1.a) relation prediction, in which LM is involved to predict the next-hop relation based on the current state and context; and 1.b) the non-parametric state transition, which is to predict the next-hop entity based on the $\mathcal{KG}$ and the predicted relation. Formally:

$$\underbrace{\boldsymbol{P}\big(e_i^t|q, e_i^{<t}\big)}_{\mathcal{KG}\text{ Reasoning (2.1)}} = \sum_r \underbrace{\boldsymbol{P}_{rel}\big(r_i^t|q, e_i^{<t}\big)}_{\text{relation prediction (2.1.1)}} \cdot \underbrace{\boldsymbol{P}_{walk}\big(e_i^t|r_i^t, e_i^{<t}\big)}_{\text{contextualized random walk (2.1.2)}}$$

We keep track of the entity distribution at each step t via the probability vector[1] $\boldsymbol{\pi}_i^{(t)} \in \mathcal{R}^{|\mathcal{E}|}$, with $\boldsymbol{\pi}_i^{(t)}[e]$ being the probability of staying at entity e, i.e., $\boldsymbol{P}\big(e_i^t = e|q, e_i^{<t}\big)$.

We highlight the three procedures in red dotted box in Figure 2. We take the first reasoning step starting from entity mention "Bauhaus" as an example. In the first red box within KIL, we predict which relation action should be taken for entity "Bauhaus", and send the prediction (e.g. "Founded") to $\mathcal{KG}$. In the second red box, $\mathcal{KG}$ re-weights the graph and conducts contextualized random walk to update entity distribution, where "Walter" has the highest probability. Finally, weighted by the entity distribution, an aggregated entity embedding is sent back to KIL and added into a placeholder token as the knowledge, so the later LM layer knows to focus on the retrieved "Walter". We introduce these steps in the following.

Input. Initially, we first identify all N entity mentions $\{m_i\}_{i=1}^N$ in the input question q as well as the corresponding $\mathcal{KG}$ entities[2].. For each mention m_i we add three special tokens as the interface for Knowledge Interaction Layers (KIL) to send instruction and receive knowledge: we add a [S-ENT] token before, and [REL], [T-ENT] tokens after each entity mention m_i. KIL can be flexibly inserted into arbitrary LM intermediate layer. By default, we just insert each KIL every N Transformer-based LM layers, thus the input to the t-th KIL are contextualized embeddings of each token k as $\texttt{LM}_k^{(t)}$, including added special tokens.

2.1 LM involved $\mathcal{KG}$ Reasoning

We first introduce the reasoning process $\boldsymbol{P}\big(e_i^t|q, e_i^{<t}\big) = \sum_r \boldsymbol{P}\big(r_i^t|q, e_i^{<t}\big) \cdot \boldsymbol{P}\big(e_i^t|r_i^t, e_i^{<t}\big)$.

2.1.1 Relation Prediction.

For each entity mention m_i, we desire to predict which relation action should take r_i^t as instruction to transit state. We define the predicted relation probability vector $\boldsymbol{\gamma}_i^{(t)} = \boldsymbol{P}_{rel}\big(r_i^t|q, e_i^{<t}\big) \in \mathcal{R}^{|\mathcal{R}|}$ representing the relation distribution to guide walking through the graph. Denote the corresponding [REL] token as $\texttt{REL}[i]$ (and similarly for other special tokens). The contextual embedding $\texttt{LM}_{\texttt{REL}[i]}^{(t)}$ encode the relevant information in question q that hints next relation. We maintain a global relation key memory $\texttt{K}_{rel} \in \mathbb{R}^{|\mathcal{R}|\times d}$ storing each relation's d-dimentional embedding. To calculate similarity, we first get relation query $Q_{\texttt{REL}[i]}^{(t)}$ by projecting relation token's embedding into the same space of key memory via a projection head Q-Proj[3] followed by a LayerNorm (abbreviated as LN), and then calculate dot-product similarity followed by softmax:

$$Q_{\texttt{REL}[i]}^{(t)} = \text{LN}^{(t)}\big(\text{Q-Proj}^{(t)}(\texttt{LM}_{\texttt{REL}[i]}^{(t)})\big), \quad \boldsymbol{\gamma}_i^{(t)} = \boldsymbol{P}_{rel}\big(r_i^t|q, e_i^{<t}\big) = \text{Softmax}\big(Q_{\texttt{REL}[i]}^{(t)}\, \texttt{K}_{rel}^T\big).$$

Note that the relation queries $\texttt{LM}_{\texttt{REL}[i]}^{(t)}$ are different for every mention m_i and reasoning step t depending on the context, and thus the the relation distributions $\boldsymbol{\gamma}_i^{(t)}$ gives contextualized predictions

[1]Throughout the paper, all vectors are row-vectors.

[2]For Wikipedia pretraining, we use the ground-truth entity label as one-hot initialization for $\boldsymbol{\pi}_i^0$. For downstream tasks we use GENRE (Cao et al., 2021) to get top 5 entity links.

[3]We denote different non-linear MLP projections as X-Proj$(h) = W_2^X\sigma(W_1^X h + b_1) + b_2$.

based on the question q. The predicted relations are sent to the knowledge graph reasoning module as instruction to conduct state transition.

2.1.2 Contextualized KG Random Walk

Next, we introduce how we conduct state transition $\boldsymbol{P}_{walk}\big(e_i^t | r_i^t, e_i^{<t}\big)$. One classic transition algorithm is random walk, which is a special case of markov chain, i.e. the transition probability only depends on previous state. Consider a state at entity s, the probability walking to target t is $\frac{1}{deg(s)}$ if $A[s,t] = 1$. Based on it, we define the Markov transition matrix for random walk as $M_{rw} = \boldsymbol{D}_A^{-1} A$, where the degree matrix $\boldsymbol{D}_A \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is defined as the diagonal matrix with the degrees $deg(1), \dots, deg(|\mathcal{E}|)$ on the diagonal. With random walk Markov matrix M_{rw} we can transit the state distribution as: $\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(t-1)} M$, The limitation of random walk is that the transition strategy is not dependent on the question q. We thus propose a Contextualized Random Walk (CRW).

Based on the predicted relation distribution $\boldsymbol{\gamma}_i^{(t)}$, we calculate a different weighted adjacency matrix $\widetilde{A}_i^{(t)} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ by adjusting the edge weight:

$$\widetilde{A}_i^{(t)} = \sum\nolimits_{r\in\mathcal{R}} w_r \cdot \boldsymbol{\gamma}_{i,r}^{(t)} \cdot A_r, \quad {M_{crw}}_i^{(t)} = \boldsymbol{D}^{-1}_{\widetilde{A}_i^{(t)}} \widetilde{A}_i^{(t)}, \ \ \forall i \in [1, N],$$

where w_r is a learnable importance weight for relation r that helps solving downstream tasks, and $\boldsymbol{\gamma}_{i,r}^{(t)}$ is the probability corresponding to relation r in $\boldsymbol{\gamma}_i^{(t)}$. With the transition matrix ${M_{crw}}_i^{(t)}$, the state transition is defined as $\boldsymbol{\pi}_i^{(t)} = \boldsymbol{\pi}_i^{(t-1)} {M_{crw}}_i^{(t)}$.

CRW allows each reasoning path ρ_i to have its transition matrix. However, as the total number of entity nodes $|\mathcal{E}|$ could be huge (e.g., 5M for WikiData), we cannot afford to update the entire adjacency matrix for every in-batch mention. We thus adopt a scatter-gather pipeline to implement graph walking, as shown in Algorithm 1 in Appendix. The complexity is # of in-batch entities times # of edges in T-hop subgraph starting from these entities, i.e., $\mathcal{O}(n \times \#\text{edge})$, and thus this operation is not expensive. Another concern is why not using Graph Neural Networks (GNNs). We provide discussion in Sec. D.3 in Appendix.

2.2 Knowledge-Injected `LM`

After we get the updated entity distribution $\boldsymbol{\pi}_i^{(t)}$, we want to inject such information back to the `LM` without harming its overall structure. We maintain a global entity embedding value memory $\mathtt{V}_{ent} \in \mathbb{R}^{|\mathcal{E}|\times d}$ storing entity embeddings. We only consider the entities within the sampled local subgraph in each batch. We thus get an entity index list $\boldsymbol{I}$ as the query to sparsely retrieve a set of candidate entity embeddings and then aggregate them weighted by entity distribution and embedding table. We then use a Value Projection block to map the aggregated entity embedding into the space of `LM`, and then directly add the transformed embedding back to the output of `T-ENT`.

$$V_i^{(t)} = \text{V-Proj}^{(t)}\big(\boldsymbol{\pi}_i^{(t)} \cdot \mathtt{V}_{ent}[\boldsymbol{I}]\big), \quad \widehat{\mathtt{LM}}^{(t)}_{\mathtt{T-ENT}[i]} = \text{LN}^{(t)}\big(\mathtt{LM}^{(t)}_{\mathtt{T-ENT}[i]} + V_i^{(t)}\big). \tag{1}$$

Then, we just take all $\widehat{\mathtt{LM}}^{(t)}_{\mathtt{T-ENT}}$ as input to next Transformer-based `LM` layer to learn the interaction between the retrieved knowledge with in-context words via self-attention.

By repeating the `KIL` for T times, the final representation $\widehat{\mathtt{LM}}^{T}$ is conditioned on the reasoning paths $\rho_i = e_i^{0:T}$, which reaches entities that are T-hop away from initial entity m_i in the question. Finally, we can predict the answer of open questions $\boldsymbol{P}\big(a | q, \{e_i^{0:T}\}_{i=1}^n\big)$ by taking knowledge-injected representation $\widehat{\mathtt{LM}}^{T}$ for span extraction, entity prediction or direct answer generation.

2.3 Pre-Train OREOLM to Reason

The design of OREOLM allows end-to-end training given QA datasets. However, due to the small coverage of knowledge facts for existing QA datasets, we need to pretrain OREOLM on a large-scale corpus to get good entity embeddings.

Salient Span Masking. One straightforward approach is to use Salient Span Masking (SSM) objective (Guu et al., 2020) masks out entities or noun tokens requiring specific out-of-context knowledge. We mainly mask out entities for guiding OREOLM to reason. Instead of randomly masking entity mentions, we explicitly sample a set of entity IDs and mask every mentions linking to these entities. This could prevent the model copy the entity from the context to fill in the blank. We also follow (Yang et al., 2019) to mask out consecutive token spans. We then calculate the cross-entropy loss on each salient span masked (SSM) token as $\mathcal{L}_{SSM}$.

2.3.1 Weakly Supervised Training of KIL

Ideally, OREOLM can learn all the entity knowledge and how to access the knowledge graph by solely optimizing $\mathcal{L}_{SSM}$. However, without a good initialization of entity and relation embeddings, KIL makes a random prediction, and the retrieved entities by $\mathcal{KG}$ reasoning are likely to be unrelated to the question. In this situation, KIL does not receive meaningful gradients to update the parameters, and LM learns to ignore the knowledge. To avoid this cold-start problem and provide entity and relation embedding a good initialization, We utilize the following two external signals as self-supervised guidance.

Entity Linking Loss. To initialize the large entity embedding tables in $\mathtt{V}_{ent}$, we use other entities that are not masked as supervision. Similar to Févry et al. (2020), we force the output embedding of [S-ENT] token before the first KIL followed by a projection head E-Proj to be close to its corresponding entity embedding:

$$E_{\texttt{S-ENT}[i]} = \text{LN}\big(\text{E-Proj}(\texttt{LM}^{(1)}_{\texttt{S-ENT}[i]})\big), \quad \mathcal{L}_{ent} = \sum\nolimits_i -\text{logSoftmax}\big(E_{\texttt{S-ENT}[i]}\ \mathtt{V}_{ent}[\boldsymbol{I}]^T\big)\boldsymbol{\pi}_i^0[\boldsymbol{I}]. \quad (2)$$

Similar to Section 2.2, we only consider entities within the batch, denoted by index $\boldsymbol{I}$. This contrastive loss guides each entity's embedding $\mathtt{V}_{ent}[e]$ closer to all its previously mentioned contextualized embedding, and thus memorizes those context as a good initialization for later knowledge integration.

Weakly Supervised Relation Path Loss. Entity mentions within each Wikipedia passage are naturally grounded to WikiData $\mathcal{KG}$. Therefore, after we mask out several entities, we can utilize the $\mathcal{KG}$ to get all paths from other entities to the masked entities as weakly supervised relation labels.

Formally, we define a **Grounded Dependency Graph** $\mathcal{DG}$, which contains all reasoning paths within T-step from other in-context entities to masked entities, and then define $R_{\mathcal{DG}}(m_i, t)$ as the set of all relations over every edges for entity mention m_i at t-th hop. Based on it, we define the weakly supervised relation label $q_i^{(t)} \in \mathbb{R}^{|\mathcal{R}|}$ as the probabilistic vector which uniformly distributed on each relation in set. Note that we call uniformly-weighted $q_i^{(t)}$ as weakly supervised because 1) some paths lead to multiple entities rather than only the target masked entity; 2) the correct relation is dependent on the context. Therefore, $q_i^{(t)}$ only provides all potential candidates for reachability, and more fine-grained signals for reasoning should be learned from unsupervised $\mathcal{L}_{SSM}$. We adopt a list-wise ranking loss to guide the model to assign a higher score on these relations than others.

$$\mathcal{L}_{rel} = \sum\nolimits_{m_i} \sum\nolimits_{t=1}^{T} -\log \boldsymbol{P}^{(t)}_{rel}\big(r|m_i, q\big) \cdot q_i^{(t)}.$$

Overall, $\mathcal{L}_{ent}$ and $\mathcal{L}_{rel}$ provide OREOLM with good initialization of the large $\mathcal{KG}$ memory. Afterward, via optimizing $\mathcal{L}_{SSM}$, the reasoning paths that provide informative knowledge receive a positive gradient, guiding OREOLM to reason.

3 Experiments

The proposed KIL layers can be pugged into most Transformer-based Language Models without hurting its original structure. In this paper, we experiment with both encoder-based LM, i.e. RoBERTa-base ($d = 768, l = 12$), and encoder-decoder LM, i.e. T5-base ($d = 768, l = 12$) and T5-large ($d = 1024, l = 24$). For all LMs, add 1 KIL layer or 2 KIL layers to the encoder layers. The statistics of $\mathcal{KG}$ are shown in Table 4 in Appendix. Altogether, it takes about 0.67B parameter for $\mathcal{KG}$ memory, which is affordable to load as model parameter. We pre-train all LMs using the combination of $\mathcal{L}_{SSM}$, $\mathcal{L}_{ent}$ and $\mathcal{L}_{rel}$ for 200k steps on 8 V100 GPUs, with a batch size of 128 and default optimizer and learning rate in the original paper, taking approximately one week to finish pre-training of T5-large model, and 1-2 days for base model.

Models	**#param**	**NQ**	**WQ**	**TQA**	**ComplexWQ**	**HotpotQA**
T5 (Base)	0.22B	25.9	27.9	29.1	11.6	22.8
+ **OREOLM** (T=1)	0.23B + 0.68B	28.3	30.6	32.4	20.8	24.1
+ **OREOLM** (T=2)	0.24B + 0.68B	28.9	31.2	33.7	23.7	26.3
T5 (Large)	0.74B	28.5	30.6	35.9	16.7	25.3
+ **OREOLM** (T=1)	0.75B + 0.68B	30.6	32.8	39.1	24.5	28.2
+ **OREOLM** (T=2)	0.76B + 0.68B	**31.0**	**34.3**	**40.0**	**27.1**	**31.4**
T5-3B (Roberts et al., 2020)	3B	30.4	33.6	43.4	-	27.8
T5-11B (Roberts et al., 2020)	11B	32.6	37.2	50.1	-	30.2

Table 1: **Closed-Book Generative QA** of Encoder-Decoder LM on (Single/Multi)-hop Dataset.

3.1 Evaluate for *Closed-Book* QA

OREOLM is designed for improving *Closed-Book* QA, so we first evaluate it in this setting. **Generative QA** Following the hyperparameters and setting in Roberts et al. (2020), we directly fine-tune the T5-base and T5-large augmented by our OREOLM on the three single-hop ODQA datasets: Natural Question **(NQ)** (Kwiatkowski et al., 2019), WebQuestions **(WQ)** (Berant et al., 2013) and TriviaQA **(TQA)** (Joshi et al., 2017). To test OREOLM's ability to solve complex questions, we also evaluate on two multi-hop QA datasets, i.e. **Complex WQ** (Talmor and Berant, 2018) and **HotpotQA** (Yang et al., 2018). Detailed dataset statistics and experimental setups are in Appendix C.

Experimental results are shown in Table 7. We use Exact Match accuracy as the metric for all the datasets. On the three single-hop ODQA datasets, OREOLM with 2 KIL blocks achieves 3.3 absolute accuracy improvement to T5-base, and 3.4 improvement to T5-large. Compared with T5 model with more model parameters (e.g., T5-3B and T5-11B), our T5-large augmented by OREOLM could outperform T5-3B on NQ and WQ datasets. In addition, OREOLM could use the generated reasoning path to interpret the model's prediction. We show examples in Table 10 in Appendix.

For the two multi-hop QA datasets, the performance improvement brought by OREOLM is more significant, i.e., 7.8 to T5-base and 8.2 to T5-large. Notably, by comparing the T5-3B and T5-11B's performance on HotpotQA (we take results from (Chen et al., 2022)), T5-large augmented by OREOLM achieves 1.2 higher than T5-11B. This shows that OREOLM is indeed very effective for improving *Closed-Book* QA performance, especially for complex questions.

Entity Prediction. Encoder-based LM (i.e. RoBERTa) in most cases cannot be directly used for *Closed-Book* QA, but more serve as reader to extract answer span. However, Verga et al. (2021) propose a special evaluation setting as *Closed-Book Entity Prediction*. They add a single [MASK] token after the question, and use its output embedding to classify WikiData entity ID. This restricts that answers must be entities that are covered by WikiData, which they call *WikiData-Answerable* questions. We follow Verga et al. (2021) to use such reduced version of WebQuestionsSP **(WQ-SP)** (Yih et al., 2015) and TriviaQA **(TQA)** as evaluation dataset, and finetune the RoBERTa (base) model augmented by OREOLM to classify entity ID. . We mainly compare OREOLM with EaE (Févry et al., 2020) and FILM (Verga et al., 2021), which are two $\mathcal{KG}$ memory augmented LM. We also run experiments on KEPLER (Wang et al., 2019), a RoBERTa model pre-trained with knowledge augmented task.

Experimental results are shown in Table 2. Similar to the observation reported by Verga et al. (2021), adding $\mathcal{KG}$ memory for this entity prediction task could significantly improve over vanilla LM, as most of the factual knowledge required to predict entities are stored in $\mathcal{KG}$. By comparing with FILM (Verga et al., 2021), which is the state-of-the-art model in this setup, OREOLM with reasoning step ($T = 2$) outperforms FILM by 2.9, with smaller memory consumption.

3.2 Analyze $\mathcal{KG}$ Reasoning Module

In our previous studies, we find that using a higher reasoning step, i.e. $T = 2$, generally performs better than $T = 1$. We hypothesize that the KG we use has many missing one-hop facts, and high-order reasoning helps recover them and empowers the model to answer related questions. To test whether OREOLM indeed can infer missing facts, we use **EntityQuestions (EQ)** (Sciavolino et al., 2021), which is a synthetic dataset by mapping each WikiData triplet to natural questions. We take RoBERTa-base model augmented by OREOLM trained on NQ as entity predictor and directly test its transfer performance on EQ dataset without further fine-tuning.

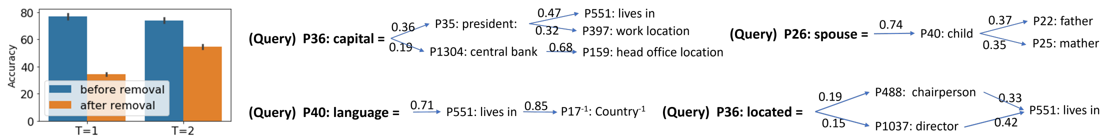


Figure 3: **Testing the reasoning capacity of OREOLM to infer missing relations**. On the **left**, the barplot shows the transfer performance on EQ before and after removing relation, OREOLM ($T = 2$) is less influenced. On the **right** shows reasoning paths (rules) automatically generated by OREOLM.

Models	#param (B)	WQ-SP	TQA
EaE (Févry et al., 2020)	0.11 + 0.26	62.4	24.4
FILM (Verga et al., 2021)	0.11 + 0.72	78.1	37.3
KEPLER (Wang et al., 2019)	0.12	48.3	24.1
RoBERTa (Base)	0.12	43.5	21.3
+ OREOLM (T=1)	0.12 + 0.68	80.1	39.7
+ OREOLM (T=2)	0.13 + 0.68	**80.9**	**40.3**
Ablation Studies			
RoBERTa + Concat KB + $\mathcal{L}_{SSM}$	0.12	47.1	22.6
+ OREOLM (T=2) **w/o PT**	0.13 + 0.68	46.9	22.7
w. $\mathcal{L}_{SSM}$	0.13 + 0.68	51.9	26.8
w. $\mathcal{L}_{SSM} + \mathcal{L}_{ent}$	0.13 + 0.68	68.4	35.7

Table 2: **Closed-Book Entity Prediction** performance of Encoder LM on WikiData-Answerable Dataset.

Models	#param (B)	NQ	TQA
Graph-Retriever Min et al. (2019)	0.11	34.7	55.8
REALM Guu et al. (2020)	0.33 + 16	40.4	-
DPR Karpukhin et al. (2020) + BERT	0.56 + 16	41.5	56.8
+ OREOLM (DPR, T=2)	0.57 + 17	43.7	58.5
FiD (Base) = DPR + T5 (Base)	0.44 + 16	48.2	65.0
+ OREOLM (T5, T=2)	0.45 + 17	49.3	67.1
+ OREOLM (DPR & T5, T=2)	0.46 + 17	51.1	68.4
FiD (Large) = DPR + T5 (Large)	0.99 + 16	51.4	67.6
+ OREOLM (T5, T=2)	0.99 + 17	52.4	68.9
+ OREOLM (DPR & T5, T=2)	1.00 + 17	**53.2**	69.5
KG-FiD (Base) (Yu et al., 2022a)	0.44 + 16	49.6	66.7
KG-FiD (Large) (Yu et al., 2022a)	0.99 + 16	**53.2**	69.8
EMDR^2 (Sachan et al., 2021b)	0.44 + 16	52.5	**71.4**

Table 3: **Open-Book QA** Evaluation.

To test whether OREOLM could recover missing relation, we mask **all** the edges corresponding to each relation separately and make the prediction again. The average results before and after removing edges are shown on the left part of Figure 3. When we remove all the edges to each relation, OREOLM with $T = 1$ drops significantly, while $T = 2$ could still have good accuracy. To understand why OREOLM ($T = 2$) is less influenced, in the right part of Figure 3, we generate a reasoning path for each relation by averaging the predicted probability score at each reasoning step and pick the relation with the top score. For example, to predict the "Capital" of a country, the model learns to find the living place of the president, or the location of a country's central bank. Both are very reasonable guesses. Many previous works (Xiong et al., 2017) could also learn such rules in an ad-hoc manner and require costly searching or reinforcement learning. In contrast, OREOLM could learn such reasoning capacity for all relations end-to-end during pre-training.

3.3 Evaluate for *Open-Book* QA

Though OREOLM is designed for *Closed-Book* QA, the learned model can serve as backbone for *Open-Book* QA. We take DPR and FiD models as baseline. For DPR retriever, we replace the question encoder to RoBERTa + OREOLM, fixing the passage embedding and only finetune on each downstream QA dataset. For FiD model, we replace the T5 + OREOLM. We also changed the retriever with our tuned DPR. Results in Table 3 show that by augmenting both retriever and generator, OREOLM improves a strong baseline like FiD, for about 3.1% for Base and 1.8% for Large, and it outperforms the very recent KG-FiD model for 1.6% in base setting, and achieve comparative performance in a large setting. Note that though our results is still lower than some recent models (e.g., EMDR^2), these methods are dedicated architecture or training framework for *Open-Book* QA. We may integrate OREOLM with these models to further improve their performance.

4 Conclusion

We presented OREOLM, a novel model that incorporates symbolic $\mathcal{KG}$ reasoning with existing LMs. We showed that OREOLM can bring significant performance gain to open-domain QA benchmarks, both for closed-book and open-book settings, as well as encoder-only and encoder-decoder models. Additionally, OREOLM produces reasoning paths that helps interpret the model prediction.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. https://openreview.net/forum?id=SJgVHkrYDH Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. https://www.aclweb.org/anthology/D13-1160/ Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. https://doi.org/10.1145/1376616.1376746 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. ACM.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data 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.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. https://openreview.net/forum?id=5k8F6UU39V Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. https://doi.org/10.18653/v1/P17-1171 Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Wenhu Chen, Pat Verga, Michiel de Jong, John Wieting, and William Cohen. 2022. https://doi.org/10.48550/arXiv.2204.04581 Augmenting pre-trained language models with qa-memory for open-domain question answering. *CoRR*, abs/2204.04581.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. 2020. https://openreview.net/forum?id=ryxjnREFwH Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. https://openreview.net/forum?id=Syg-YfWCW Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rajarshi Das, Ameya Godbole, Ankita Naik, Elliot Tower, Robin Jia, Manzil Zaheer, Hannaneh Hajishirzi, and Andrew McCallum. 2022. http://arxiv.org/abs/2202.10610 Knowledge base question answering by case-based reasoning over subgraphs. *CoRR*, abs/2202.10610.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021. https://doi.org/10.18653/v1/2021.emnlp-main.755 Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9594–9611. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. https://doi.org/10.18653/v1/n19-1423 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. Association for Computational Linguistics.

Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. 2020. https://openreview.net/forum?id=SJxstlHFPH Differentiable reasoning over a virtual knowledge base. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. https://doi.org/10.18653/v1/p19-1259 Cognitive graph for multi-hop reading comprehension at scale. 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 2694–2703. Association for Computational Linguistics.

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. https://www.aclweb.org/anthology/2020.emnlp-main.99/ Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1295–1309. Association for Computational Linguistics.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. http://arxiv.org/abs/2004.07202 Entities as experts: Sparse memory access with entity supervision. *CoRR*, abs/2004.07202.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. http://arxiv.org/abs/2002.08909 REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. https://doi.org/10.18653/v1/P17-1147 Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. http://arxiv.org/abs/2004.04906 Dense passage retrieval for open-domain question answering. *CoRR*, abs/2004.04906.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. https://doi.org/10.18653/v1/2021.findings-acl.223 Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. 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 2526–2538. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. https://transacl.org/ojs/index.php/tacl/article/view/1455 Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Ni Lao, Tom M. Mitchell, and William W. Cohen. 2011. https://aclanthology.org/D11-1049/ Random walk inference and learning in A large scale knowledge base. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 529–539. ACL.

Patrick S. H. Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021a. https://www.aclweb.org/anthology/2021.eacl-main.86/ Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1000–1008. Association for Computational Linguistics.

Patrick S. H. Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021b. http://arxiv.org/abs/2102.07033 PAQ: 65 million probably-asked questions and what you can do with them. *CoRR*, abs/2102.07033.

Chen Liang, Jonathan Berant, Quoc V. Le, Kenneth D. Forbus, and Ni Lao. 2017. https://doi.org/10.18653/v1/P17-1003 Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 23–33. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. https://doi.org/10.18653/v1/D19-1282 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2829–2839. Association for Computational Linguistics.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William W. Cohen. 2020. http://arxiv.org/abs/2010.14439 Differentiable open-ended commonsense reasoning. *CoRR*, abs/2010.14439.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. https://ojs.aaai.org/index.php/AAAI/article/view/16796 KG-BART: knowledge graph-augmented BART for generative commonsense reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6418–6425. AAAI Press.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. http://arxiv.org/abs/1911.03868 Knowledge guided text retrieval and reading for open domain question answering. *CoRR*, abs/1911.03868.

Nina Pörner, Ulli Waltinger, and Hinrich Schütze. 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.71 E-BERT: efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 803–818. 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. http://jmlr.org/papers/v21/20-074.html Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/807 Transforming dependency structures to logical forms for semantic parsing. *Trans. Assoc. Comput. Linguistics*, 4:127–140.

Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. https://openreview.net/forum?id=BJgr4kSFDS Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. https://doi.org/10.18653/v1/2020.emnlp-main.437 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 2020, Online, November 16-20, 2020*, pages 5418–5426. Association for Computational Linguistics.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021a. End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6648–6662. Association for Computational Linguistics.

Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. 2021b. https://proceedings.neurips.cc/paper/2021/hash/da3fde159d754a2555eaa198d2d105b2-Abstract.html End-to-end training of multi-document reader and retriever for open-domain question answering. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 25968–25981.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. https://doi.org/10.18653/v1/2021.emnlp-main.496 Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6138–6148. Association for Computational Linguistics.

Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. 2019. https://doi.org/10.18653/v1/D19-1242 Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. 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 2380–2390. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. https://doi.org/10.18653/v1/n18-1059 The web as a knowledge-base for answering complex questions. 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 641–651. Association for Computational Linguistics.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. 2021. https://doi.org/10.18653/v1/2021.naacl-main.288 Adaptable and interpretable neural memory over symbolic knowledge. 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 3678–3691. Association for Computational Linguistics.

Denny Vrandecic and Markus Krötzsch. 2014. https://doi.org/10.1145/2629489 Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2019. http://arxiv.org/abs/1911.06136 KEPLER: A unified model for knowledge embedding and pre-trained language representation. *CoRR*, abs/1911.06136.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. https://doi.org/10.18653/v1/d17-1060 Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 564–573. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. https://doi.org/10.18653/v1/d18-1259 Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. https://doi.org/10.18653/v1/2021.naacl-main.45 QA-GNN: reasoning with language models and knowledge graphs for question answering. 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 535–546. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. https://doi.org/10.3115/v1/p15-1128 Semantic parsing via staged query graph generation: Question answering with knowledge base. 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 1321–1331. The Association for Computer Linguistics.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022a. https://aclanthology.org/2022.acl-long.340 Kg-fid: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4961–4974. Association for Computational Linguistics.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2022b. http://arxiv.org/abs/2010.00796 JAKET: joint pre-training of knowledge graph and language understanding. Conference on Artificial Intelligence, AAAI,.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. http://arxiv.org/abs/2201.08860 Greaselm: Graph reasoning enhanced language models for question answering. *CoRR*, abs/2201.08860.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. https://doi.org/10.18653/v1/p19-1139 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. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. http://arxiv.org/abs/1709.00103 Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.

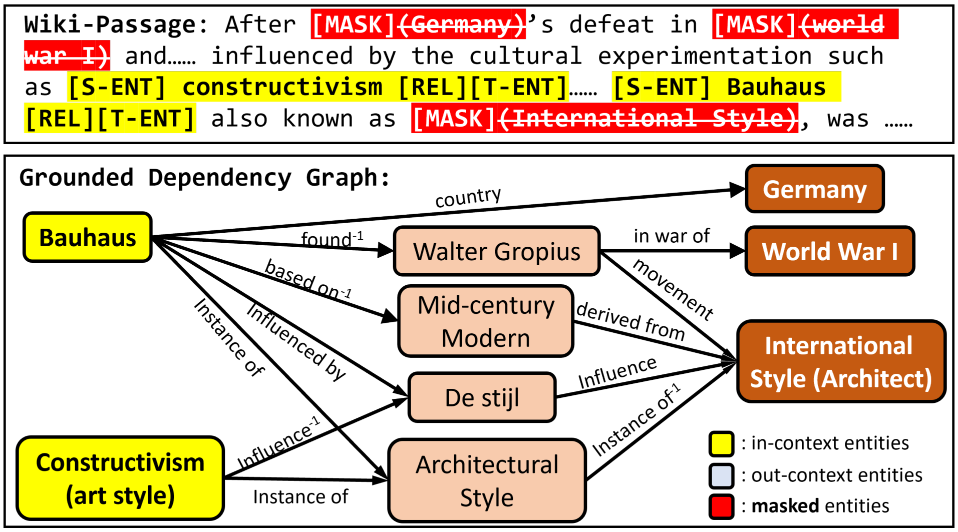


Figure 4: Pre-training sample w/ golden reasoning path. More real examples are shown in Table 8 in Appendix.

Name	**Number**	**dimension**	**#param (M)**
Number of Entity	4,947,397	128	633
Number of Relation	2,008	768	1.5
Number of Edges	45,217,947	-	47

Table 4: Statistics and parameter of $\mathcal{KG}$ Memory.

A Related Work

To encode knowledge (significantly smaller than the web corpus) as *memory* into LM parameter, a line of works try compressed knowledge including QA pairs (Chen et al., 2022; Lewis et al., 2021b), entity embedding (Févry et al., 2020) and reasoning cases (Das et al., 2021, 2022). There's also several works utilizing Knowledge Graph(KG). FILM (Verga et al., 2021) turns KG triplets into memory. Given a question, LM retrieves most relevant triplet as answer. GreaseLM (Zhang et al., 2022) propose to interact LM with KG via a interaction node. JAKET (Yu et al., 2022b) encode text and KG independently and fuse information at late stage. We introduce and discuss with other related works in Sec. D in Appendix.

B Implementation Details

Entity Linking durine pre-training We use the 2021 Jan. English dump of Wikidata and Wikipedia. For each wikipedia page, we link all entity mentions with hyperlinks to WikiData entity entry, augment all other mentions with same aliases, tokenize via each LM's tokenizer and split into chunks with maximum token length allowed. We then construct induced k-hop subgraphs connecting entities within each chunk for quickly get grounded computational graph.

For entities, Wikipedia provides hyperlinks with ground-truth entity ID, but it doesn't cover all the entity mentions, mostly hyperlinks only appear when this entity appears for the first time. Therefore, we first collect all entities appeared in hyperlinks as well as their aliases stored in WikiData, and then search any mentions that have any of these alias and link it to the corresponding entity.

Implementation of Contextualized Random Walk We first gather the entity and relation probability to each edge, and then scatter the probability to target nodes. This allows us to simultaneously conduct message passing with modified adjacency weight $\widetilde{A}_i^t$ for all entity mention m_i in parallel.

Dataset	Train	Dev	Test
Natural Questions	58880	8757, 3610	
Trivia QA	60413	8837	11313
Web Questions	2474	361	2032
HotpotQA			
Complex WebQ	27623	3518	3531
WebQ-SP (Wiki-answerable)	1388	153	841
FreebaseQA (Wiki-answerable)	12535	2464	2440

Table 5: Dataset Train/Valid/Test splits.

Models	**#param (B)**	WQ-SP	TQA
RoBERTa (Base)	0.12	47.5	40.3
+ **OREOLM** (T=1)	0.12 + 0.68	89.7	61.4
+ **OREOLM** (T=2)	0.13 + 0.68	**92.4**	**66.8**

Table 6: **Closed-Book Entity Prediction** validation performance of Encoder RoBERTa on WikiData-Answerable Dataset.

Algorithm 1: Pytorch Pseudocode of CRW

```
def ContextualizedRandomWalk(
    i_init, KG,    # initial entity index and Graph
    w_deg, w_rel,  # inv(degree) and relation weights
    p_ent, p_rel   # entity and predicted relation dis-
                   # tribution tensor @ t-th step.
): -> FloatTensor
    # Get <src, rel, tgt> edge list of k-hop subgraph
    i_src, i_rel, i_tgt = k_hop_subgraph(i_init, KG)
    # Gather entity and relation probability to edge
    p_src  = (p_ent * w_deg)[:, i_src] # N x n_edge
    p_rel  = (p_rel * w_rel)[:, i_rel] # N x n_edge
    p_edge = l1_normalize(p_src * p_rel, dim=1)
    # Scatter edge probability to target node
    p_ent  = scatter_add(src=p_edge, idx=i_tgt, dim=1)
    return p_ent  #(t+1)-th step's entity distribution
```

Hyperparameters In this work, we don't have too much hyperparmaters to be tuned, as most parameters as well as optimizing setting of LM is fixed. Our random walk part is non-parametric. The only tunable hyperparamter is hidden dimension size. We simply choose one setting, which is 128 for entity embedding, and 768 for relation embedding. The former is because entity is super large (over 5M), so we use a reletively smaller dimension size. Detailed statistics about wikidata memory is in Table 4.

C Dataset Details

Below shows details for each dataset, and the detailed dataset split is shown in Figure 5

Natural Questions Kwiatkowski et al. (2019) contains questions from Google search queries, and the answers are text spans in Wikipedia. We report short answer Exact Match (EM) performance. The open version of this dataset is obtained by discarding answers with more than 5 tokens.

WebQuestions (WQ) Berant et al. (2013) contains questions from Google Suggest API, and the answers are entities in Freebase.

TriviaQA Joshi et al. (2017) contains trivia questions and answers are text spans from the Web. We report Exact Match (EM) performance. We use its unfiltered version for evaluation.

HotpotQA Yang et al. (2018) is a multi-hop QA dataset. There are two evaluation settings. In the *distractor setting*, 10 candidate paragraphs are provided for each question, of which there are two

Models	#param	NQ	WQ	TQA	ComplexWQ	HotpotQA
T5 (Large)	0.74B	-	-	-	-	-
+ OREOLM (T=2)	0.76B + 0.68B	33.6	38.9	42.7	29.6	35.5

Table 7: **Closed-Book Generative QA** validation performance of T5.

golden paragraphs. In the *full-wiki setting*, a model is required to extract paragraphs from the entire Wikipedia. We report Exact Match (EM) on full-wiki setting.

Complex WebQuestions Talmor and Berant (2018) is a dataset that composite simple one-hot questions in WebQuestionsSP by extending entities or adding constraints, so that each question eequires complex reasoning to solve.

WebQuestionsSP Yih et al. (2015) is annotated dataset from WebQuestions, such taht each quetsion is answerable using Freebase via a SQL query.

D Other Related Works

D.1 Introduce other related works

Open-Domain Question Answering aims at answering factoid questions by refering to a large-scale corpus. Most works adopt a two-stage pipeline proposed in Chen et al. (2017) that combines a retriever with a neural reader. There also exists several QA works using $\mathcal{KG}$ to help ODQA. For example, Asai et al. (2020) and Min et al. (2019) expand the entity graph following wikipedia hyperlinks or triplets in knowledge base. Ding et al. (2019) extract entities from current context via entity-linking and turn them into a cognitive graph, and a graph neural network is applied on top of it to extract answer. Dhingra et al. (2020) and Lin et al. (2020) construct an entity-mention bipartite graph and then model the QA reasoning as graph traversal by filtering only the contexts that are relevant to the question.

Knowledge-Base Question Answering Traditional parsing-based methods parse the question into some intermediate query (e.g., SQL language, query graphs), which can execute on a knowledge base to get answer (Berant et al., 2013; Yih et al., 2015; Reddy et al., 2016; Zhong et al., 2017; Liang et al., 2017). However, existing knowledge bases suffer from low coverage of entities and relations required for open-ended questions. As an alternative, several works try to incorporate the structured knowledge into neural QA models for differentiable reasoning. Lin et al. (2019) and Feng et al. (2020) parse the question into a sub-graph of knowledge base, and apply graph neural networks as reasoner to extract answers. Chen et al. (2020) integrates general symbolic operations as basic units, and parse questions into compositional programs to answer general questions.

Knowledge-augmented Language Models explicitly incorporate external knowledge (e.g. knowledge graph) into `LM`. Overall, these approaches can be grouped into two categories: The first one is to explicitly inject knowledge representation into language model pre-training, where the representations are pre-computed from external sources (Zhang et al., 2019; Liu et al., 2021). For example, ERNIE Zhang et al. (2019) encodes the pre-trained TransE Bordes et al. (2013) embeddings as input. The second one is to implicitly model knowledge information into language model by performing knowledge-related tasks, such as entity category prediction (Yu et al., 2022b) and graph-text alignment Ke et al. (2021). For example, JAKET jointly pre-trained both the KG representation and language representation by adding two self-supervised learning objectives (i.e., entity category prediction, relation type prediction) on knowledge graphs (Yu et al., 2022b).

D.2 Discussion with Previous Works

Compare with FILM Though FILM has the advantage of end-to-end training and easily modification of knowledge memory, it simply stacks $\mathcal{KG}$ module on top of `LM` without interaction, and can only handle one-hop relational query that is answerable by $\mathcal{KG}$. Our approach, OREOLM, follows the same *memory* idea by encoding $\mathcal{KG}$ into LM parameter, and we desire LM and $\mathcal{KG}$ reasoning module could interact and collaboratively improve each other.

Notably, OREOLM with $T = 1$ shares a similar design with FILM. The major differences are: 1) they store every triplet as a key-value pair, while we explicitly keep the $\mathcal{KG}$ adjacency matrix and conduct a random walk, which has smaller search space and is more controllable. 2) They add the memory on top of LM, and thus the knowledge could not help language understanding, and FILM could mainly help wikipedia-answerable questions. Instead, we insert the KIL layer amid LM layers to encourage interaction, and thus the model could also benefit encoder-decoder model (as shown above).

Compare with Previous Path-Based Reasoning and Retrieval Pre-Training Note that as our definition of entity state $\boldsymbol{\pi}_i$ and relation action $\boldsymbol{\gamma}_i$ are both continuous probabilistic vector, the whole $\mathcal{KG}$ Reasoning is fully differentiable and thus could be integrated into LM seamlessly and trained end-to-end. This is different from previous path traversal works such as DeepPath Xiong et al. (2017) and MINERVA Das et al. (2018), which defines state and action as discrete and could only be trained via reinforcement learning rewards. The reasoner training is also different from passage retrieval pre-training Guu et al. (2020); Sachan et al. (2021a), as the passage are naturally consisted of discrete tokens, and thus the reader is still required to re-encode the question with each passage, and different objectives are required to train retriever and reader separately.

D.3 Discussion of Graph Walking-based Reasoning vs Graph Neural Networks

Recently, Graph Neural Networks (GNNs) have shown superior performance for structured representation learning. There's also a lot of works trying to use GNNs for Question Answering (Yasunaga et al., 2021; Zhang et al., 2022). The one that has very similar motivation with us is GreaseLM. Therefore, a natural question is, whether could we use GNN instead of the non-parametric random walk module, for ODQA?

To answer this question, let's consider a simplest setup of GNN. We could identify initial entities, connected them via a k-hop subgraph, and encode graph with text (Zhang et al., 2022) or independently (Yu et al., 2022b). When we want to retrieve knowledge from graph to LM, normally we just take the contextualized node embedding as input for knowledge fusion.

In this setup, say the answer is K-hop away from an initial entity, the ground-truth reasoning path is $e_0, r_1, e_1, r_2, ..., e_{k-1}, r_k, e_k = a$. Using our method, we first predict $r1$, transit to e_1, and step by step conduct reasoning via walking. However, if we use GNN's final embedding, it requires to pass information from neighbor to itself. Therefore, suppose we have a K-layer GNN, the first step should be identify r_k, and pass information from answer $e_k = a$ to e_{k-1}. This is conter-intuitive as we normally cannot assume to know the answer, nor knowling the last step to reach the answer. In situations where all candidate answer is given, like CommonSenseQA, where GreaseLM mainly works on, this problem is less harmful as it's guaranteed to contain the answer in a restricted small graph. However, in open-domain setup, we need to try best to narrow down the search space by following the forward reasoning instead of the backward manner. Therefore, in this work we adopt walking-based reasoning.

E Illustration of Pre-Trained Data and Reasoning Paths

The pre-training samples and reasoning paths (generated by T5-large on NQ dataset) is shown from Table 8-11.

F Ablation Studies

We conduct several ablation studies to evaluate which model design indeed contributes to the model. As shown in the bottom blocks in Table 2, we first remove the $\mathcal{KG}$ reasoning component and provide RoBERTa base model via concatenated KB triplets and train such a model using $\mathcal{L}_{SSM}$ over the same WikiDataset. Such a model's results are close to the KEPLER results but much lower than other models with explicit knowledge memory. We further investigate the role of pre-training tasks. Without pre-training, the OREOLM only performs slightly better than RoBERTa baseline, due to the cold-start problem of entity and relation embedding. We further show that removing $\mathcal{L}_{ent}$ and $\mathcal{L}_{ent}$

could significantly influence final performance. The current combination is the best choice to train OREOLM to reason.

G Limitations

Limited Reasoning Steps In our experiments, we show that using reasoning step $T = 2$ has better performance to $T = 1$ on one-hop and multi-hop (mostly two) QA datasets. Thus, it's a natural question about whether we could extending reasoning steps more? As previous KG reasoning mostly could support very long path (with LSTM design)

Though we didn't spend much time exploring before the paper submission, we indeed try using $T = 3$, but currently it didn't get better results. We hypothesize the following reasons: 1) A large portion of our current model's improvement relies on the weakly supervised relation pre-training. To do it, we construct a K-hop (K=2 now) subgraph, and sample dependency graph based on it. The larger K we choose, the more noise is included into the generated relation label, in an exponential increasing speed. Thus, it's harder to get accurate reasoning path ground-truth for high-order T. Another potential reason is that within Transformer model, the representation space in lower and upper layer might be very different, say, encode more syntax and surface knowledge at lower layers, while more semantic knowledge at upper layers. Currently we adopt a MLP projection head, wishing to map integrated knowledge into the same space, but it might have many flaws and need further improvement.

Large Entity Embedding Table requires Pre-Training and GPU resources Our current design has a huge entity embedding table, which should be learned through additional supervision and could not directly fine-tune to downstream tasks. This is restricts our approach's usage.

Require Entity Linking Current model design requires an additional step of entity linking for incoming questions, and then add special tokens as interface. A truly end-to-end model should identify which elements to start conducting reasoning by its own without relying on external models.

Only support relational path-based reasoning Though there are lots of potential reasoning tasks, such as logical reasoning, commonsense reasoning, physical reasoning, temporal reasoning, etc. Our current model design mainly focus on path-based relational reasoning, and it should not work for other reasoning tasks at current stage.

Unreasonable Assumption of Path In-dependency When we derive equation 1, we have the assumption that reasoning paths starting from different entities should be independent. This is not always correct, especially for questions that require logical reasoning, say, have conjunction or disjunction operation over each entity state. And thus our current methods might not work for those complex QA with logical dependencies.

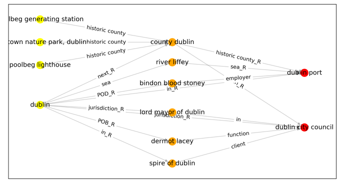
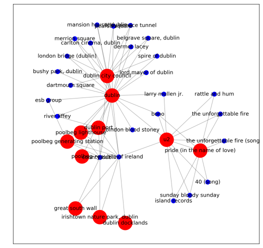
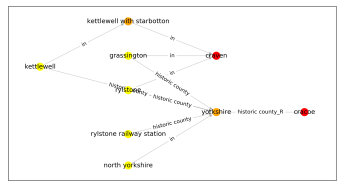
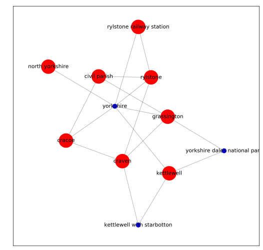
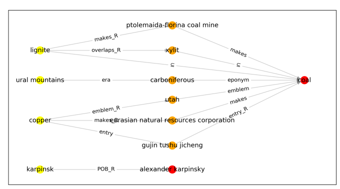
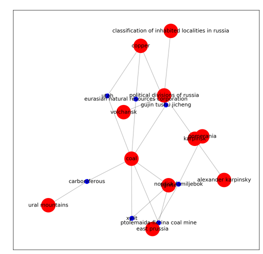
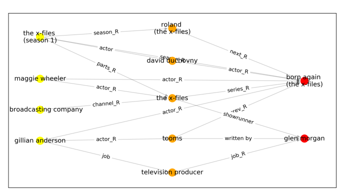
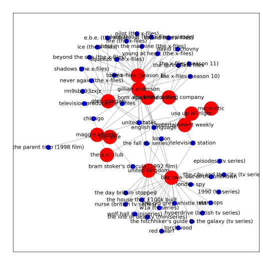
Title	Masked Text	Ground Truth	Dependency Graph	2-Hop Graph
Poolbeg	the lighthouse was [mask] [s-ent] [mask] [rel] [t-ent] completed in 1795. overview. the [s-ent] poolbeg[rel] [t-ent] “peninsula” is home to a number of landmarks including the [s-ent] [mask][rel] [t-ent] , the [s-ent] pool[mask] lighthouse[rel] [t-ent] , the [s-ent] irishtown nature park[rel] [t-ent] , the southern part of [s-ent] [mask][rel] [t-ent] ...	[’ connected to land by the’, ’ great south wall’, ’ great south wall’, ’beg’, ’ dublin port’, ”’s main power station,”, ’ structures in’, ’48’, ’ a process to list the’, ’ after the station’, ’, including 3,’, ’ dublin city council’, ’ quarter” on the’]		
Rylstone	it is situated very near to [s-ent] [mask][rel] [t-ent] and about 6 miles south west[mask] [s-ent] [mask]ington[rel] [t-ent] . the population of the [s-ent] civil parish[rel] [t-ent] as of the 2011 census was 160. [s-ent] rylstone railway station[rel] [t-ent] opened in 1902, closed to passengers in 1930, and closed completely in 1969....	[’ craven’, ’ cracoe’, ’ of’, ’ grass’, ’ the inspiration for’, ’ tour de france’, ’stone’, ’ by will’...]		
Karpinsk	ologist [s-ent] [mask] [rel] [t-ent] . history.[mask]the settlement of bogoslovsk () was founded in either 1759 or in 1769. it remained one of the largest [s-ent] copper[rel] [t-ent] production centers in the [s-ent] urals[rel] [t-ent] [mask] [s-ent] [mask][rel] [t-ent] deposits started to be mined in 1911.....	[’ alexander karpinsky’, ”, ’ until 1917.’, ’ coal’, ’erman civilians, who’, ’ and’, ’ years of’, ’ forest laborers. moreover’, ’ in’, ’ the’, ’ framework of the’, ’ districts’, ’ karpinsk’, ’insk’...]		
3 (The X-Files)	[s-ent] [mask][mask][rel] [t-ent] ". [s-ent] gillian anderson[rel] [t-ent] is absent[mask][mask] episode as she was on leave to give birth to her daughter piper at the time. this episode was the first[mask] not appear. reception. ratings. "3" premiered on the [s-ent] fox network[rel] [t-ent] on, and was first broadcast in the [s-ent] united kingdom[rel] [t-ent].....	[’ny had’, ’ episode’, ’born again’, ’ from the’, ’ in which scully did’, ’. it was’, ’egall’, ’ metacritic’, ’ as "wretched’, ’ fact that’, ’ background noise for a’, ’ heavy-handed attempts at’, ’ glen morgan’, ’ doing an episode on’]		

Table 8: Example of Pre-training data points (Part 1).

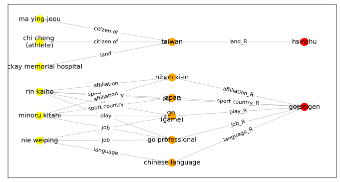
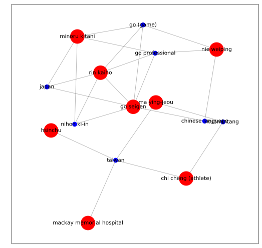
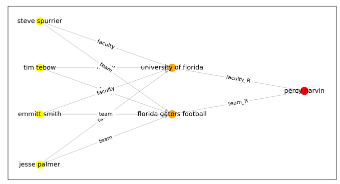
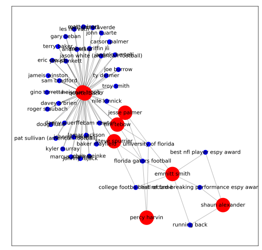
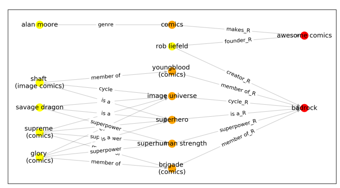
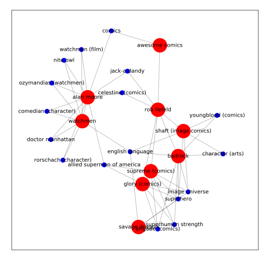
Title	Masked Text	Ground Truth	Dependency Graph	K-Hop Graph
Shen Chun-shan	his memoirs, he suffered his second stroke[mask][mask], even after his second stroke, he continued writing; his series of biographies of five go masters [s-ent] [mask][mask][mask][rel] [t-ent] , [s-ent] minoru kit[mask][rel] [t-ent]	['. however', ' go seigen', 'ani', ' 2007, he', ' was hospital', ' hsinchu', 'after surgery', ' scale', ' continuing to improve.', ' his coma. in'...]		
2007 Florida Gators football team	[s-ent] tim[mask][mask][rel] [t-ent] completed 22 of 27 passes for 281 yards passing and also ran for[mask] yards on 6 carries. [s-ent] [mask] [rel] [t-ent] carried the ball 11 times for 113 yards[mask] two touchdowns and also caught 9 passes for 110[mask] receiving, becoming the first player in school history	[' tebow', ' 35', ' percy harvin', ' and', ' yards', ' 30–9', ' renewed their budding', ' gamecocks', 'gator', ' quarterback', ' set a career-high', ' of these five rushing', '.', ' percy harvin', ' sinus infection.', 'ators', ' touchdown']		
Judgment Day (Awesome Comics)	[s-ent] alan moore[rel] [t-ent] used "judgment day" to reject the violent, deconstructive clichés of 1990s comics inadvertently caused by his own work on " [s-ent] watchmen[rel] [t-ent] ", "" and " [s-ent] saga of the[mask][mask][rel] [t-ent] " and uphold the values of classic superhero comics. the series deals with a metacommentary of the notion of retcons to super-hero histories as [s-ent] alan moore[rel] [t-ent] [mask] for the characters of [s-ent] [mask][mask][rel] [t-ent] , to replace the shared universe they left when [s-ent] rob liefeld[rel] [t-ent] left image several years earlier. plot. in[mask], mick tombs/ [s-ent] knightsabre[rel] [t-ent].....	[' swamp thing', ' himself creates a new backstory', ' awesome comics', ' 1997', 'riptide', ' knightsabre appears to be', ' and sw', ' badrock', ' supreme', 'by', ' analyzing', ' cybernetic young', ' it, and it has', 'ue out', ', administrator for youngblood']		

Table 9: Example of Pre-training data points (Part 2).

Question	Answer	Reasoning Paths as Rationale
southern soul was considered the sound of what independent record label	['Motown']	soul music $\xrightarrow{\text{genre-R}}$? $\xrightarrow{\text{label}}$? independent record label $\xrightarrow{\text{belong}}$? $\xrightarrow{\text{is a-R}}$?
who is the bad guy in lord of the rings	['Sauron']	the lord of the rings (film series) $\xrightarrow{\text{theme}}$? $\xrightarrow{\text{characters}}$?
where was the mona lisa kept during ww2	['the Ingres Museum', "Château d'Amboise", 'Château de Chambord', 'the Loc - Dieu Abbey']	mona lisa $\xrightarrow{\text{creator}}$? $\xrightarrow{\text{tomb}}$? world war 2 $\xrightarrow{\text{take place}}$? $\xrightarrow{\text{located-R}}$?
who have won the world cup the most times	['Brazil']	fifa world cup $\xrightarrow{\text{parts}}$? $\xrightarrow{\text{land}}$?
who wrote the song the beat goes on	['Sonny Bono']	song $\xrightarrow{\text{album type-R}}$? $\xrightarrow{\text{author}}$?
who plays mrs. potato head in toy story	['Estelle Harris']	toy story $\xrightarrow{\text{series}}$? $\xrightarrow{\text{VO}}$?
who plays caroline on the bold and beautiful	['Linsey Godfrey']	the bold and the beautiful $\xrightarrow{\text{in work-R}}$? $\xrightarrow{\text{actor}}$?
where are the fruits of the spirit found in the bible	['Epistle to the Galatians']	bible $\xrightarrow{\text{parts}}$? $\xrightarrow{\text{parts}}$?
who is the only kaurava who survived the kurukshetra war	['Yuyutsu']	kaurava $\xrightarrow{\text{in work}}$? $\xrightarrow{\text{in work-R}}$? Kurukshetra War $\xrightarrow{\text{location}}\xrightarrow{\text{live in-R}}$
what is the deepest depth in the oceans	['Mariana Trench']	ocean $\xrightarrow{\text{in}}$? $\xrightarrow{\text{lowest point}}$?
where did the french national anthem come from	['Strasbourg']	national anthem $\xrightarrow{\text{is a-R}}$? $\xrightarrow{\text{released in}}$?

Table 10: Example of QA prediction with reasoning path on NQ (part 1).

Question	Answer	Generated Reasoning Paths as Rationale
who sings the song where have all the flowers gone	['Pete Seeger']	song $\xrightarrow{\text{album type-R}}$? $\xrightarrow{\text{actor}}$?
who discovered some islands in the bahamas in 1492	['Christopher Columbus']	the bahamas $\xrightarrow{\text{entry}}$? $\xrightarrow{\text{entry-R}}$?
which type of wave requires a medium for transmission	['mechanical waves', 'heat energy', 'Sound']	wave $\xrightarrow{\text{belong-R}}$? $\xrightarrow{\text{belong-R}}$?
land conversion through burning of biomass releases which gas	['traces of methane', 'carbon monoxide', 'hydrogen']	gas $\xrightarrow{\text{belong-R}}$? $\xrightarrow{\text{as-R}}$?
the sum of the kinetic and potential energies of all particles in the system is called the	['internal energy']	kinetic energy $\xrightarrow{\text{belong}}$? $\xrightarrow{\text{belong-R}}$? potential energy $\xrightarrow{\text{belong}}$? $\xrightarrow{\text{belong-R}}$?
who did seattle beat in the super bowl	['Denver Broncos']	super bowl $\xrightarrow{\text{organizer}}$? $\xrightarrow{\text{league-R}}$?
what is the name of the girl romeo loved before juliet	['Rosaline']	romeo $\xrightarrow{\text{in work}}$? $\xrightarrow{\text{in work-R}}$?
who will get relegated from the premier league 2016/17	['Hull City', 'Sunderland', 'Middlesbrough']	premier league $\xrightarrow{\text{league-R}}$? $\xrightarrow{\text{POB}}$?
actress in the girl with the dragon tattoo swedish	['Noomi Rapace']	sweden $\xrightarrow{\text{speaking}}$? $\xrightarrow{\text{mother tongue-R}}$?

Table 11: Example of QA prediction with reasoning path on NQ (part 2).