KoRC: Knowledge oriented Reading Comprehension Benchmark for Deep Text Understanding

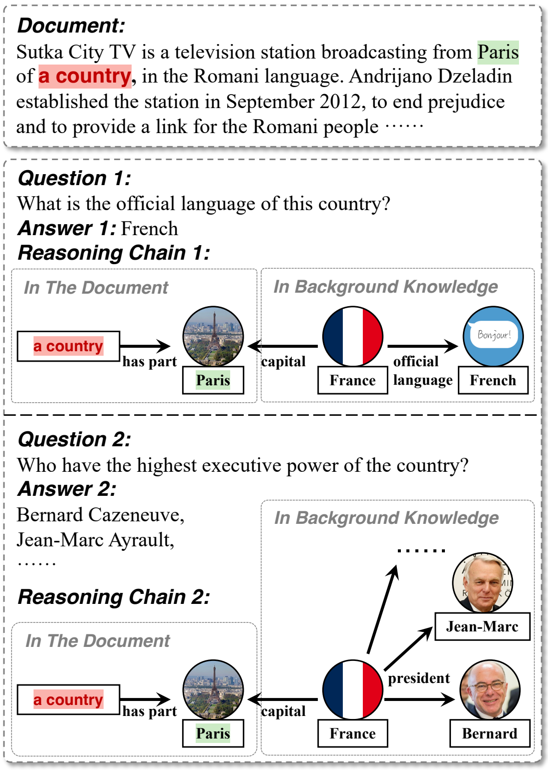
Zijun Yao[1,2*] **Yantao Liu**[3,4*] **Xin Lv**[1,2]
Shulin Cao[1,2] **Jifan Yu**[1,2] **Lei Hou**[1,2] **Juanzi Li**[1,2†]
Department of Computer Science and Technology,
[1]BNRist; [2]KIRC, Institute for Artificial Intelligence
Tsinghua University, Beijing 100084, China
[3]University of Chinese Academy of Sciences [4]Zhipu.AI
yaozj20@mails.tsinghua.edu.cn, {houlei,lijuanzi}@tsinghua.edu.cn

Abstract

Deep text understanding, which requires the connections between a given document and prior knowledge beyond its text, has been highlighted by many benchmarks in recent years. However, these benchmarks have encountered two major limitations. On the one hand, most of them require human annotation of knowledge, which leads to limited knowledge coverage. On the other hand, they usually use choices or spans in the texts as the answers, which results in narrow answer space. To overcome these limitations, we build a new challenging benchmark named KoRC in this paper. Compared with previous benchmarks, KoRC has two advantages, *i.e.,* broad knowledge coverage and flexible answer format. Specifically, we utilize massive knowledge bases to guide annotators or large language models (LLMs) to construct knowledgable questions. Moreover, we use labels in knowledge bases rather than spans or choices as the final answers. We test state-of-the-art models on KoRC and the experimental results show that the strongest baseline only achieves 68.3% and 30.0% F1 measure in the in-distribution and out-of-distribution test set, respectively. These results indicate that deep text understanding is still an unsolved challenge. The benchmark dataset, leaderboard, and baseline methods are released in https://github.com/THU-KEG/KoRC.

1 Introduction

Deep text understanding requires the integration of text information with its relevant background (prior) knowledge (Gough and Tunmer, 1986; Castles et al., 2018; Smith et al., 2021). It has been a long-pursued goal in natural language understanding (McCarthy, 1976; Norvig, 1987; Huang et al., 2019) for decades, and plays a key role in many real-world applications.



Document:
Sutka City TV is a television station broadcasting from Paris of **a country**, in the Romani language. Andrijano Dzeladin established the station in September 2012, to end prejudice and to provide a link for the Romani people ······

Question 1:
What is the official language of this country?
Answer 1: French
Reasoning Chain 1:

Question 2:
Who have the highest executive power of the country?
Answer 2:
Bernard Cazeneuve,
Jean-Marc Ayrault,
······
Reasoning Chain 2:

Figure 1: Examples of KoRC. Both question 1 and question 2 require to read the document and make connections to the background knowledge beyond the text.

Many benchmarks have been proposed to guide the development of deep text understanding skills. Early attempts formalize text understanding into machine reading comprehension (MRC) framework, such as SQuAD (Rajpurkar et al., 2016) and RACE (Lai et al., 2017). Readers are required to answer questions about the given document in MRC tasks. Recently proposed benchmarks further highlight the requirement of *deep* text understanding. To answer their questions, benchmarks such as CosmosQA (Huang et al., 2019), DREAM (Sun et al.,

[*] Yao and Liu contributes equally to KoRC. Work is done when Liu is an intern at Zhipu.AI.

[†] Corresponding author.

2019), and C^3 (Sun et al., 2020) have tapped into knowledge beyond the text. Moreover, it is necessary for deep text understanding to reason over a combination of different knowledge sources, as required by QAMPARI (Amouyal et al., 2022) and WikiHop (Welbl et al., 2018), *etc.* However, these benchmarks have encountered two limitations.

Limited Knowledge Coverage. Many of existing benchmarks are constructed based on knowledge provided by expert annotators (e.g., QUARTZ (Tafjord et al., 2019)) and knowledgeable questions written by question annotators from scratch (*e.g.,* CosmosQA (Huang et al., 2019)). The discrepancy between the limited background knowledge they cover and massive open-domain knowledge makes it difficult to measure deep text understanding skills at large. Fortunately, this can be mitigated by generating questions based on large-scale knowledge resources scattered across real-world knowledge bases.

Narrow Answer Space. As a compromise for easy construction and evaluation, a large portion of benchmarks ask multiple-choice questions (Lai et al., 2017; Sun et al., 2019) or have answers being spans in the provided reading material (Hewlett et al., 2016; Welbl et al., 2018; Amouyal et al., 2022). However, multiple-choice questions are processed simply as classification tasks. Questions based on span-extraction also increasingly become insufficient to challenge the state-of-the-art (SOTA) language models that already show great performance at information extraction (Xie et al., 2022).

Inspired by the common grounds on deep text understanding, we build a new challenging benchmark, KORC, for **K**nowledge **o**riented **R**eading **C**omprehension, as shown in Figure 1. Its most important feature is that both the reading material and external background knowledge are indispensable for every question within KORC. Readers must connect the document with their equipped prior knowledge and reason across both the text and the background knowledge to reach the final answers.

Different from previous benchmarks, KORC has two advantages. ***Broad knowledge coverage***. KORC does not require manual knowledge annotation from scratch. Instead, it uses off-the-shelf knowledge bases as its background knowledge sources to guide the construction of knowledgable questions. More exhilaratingly, KORC proves it feasible for LLMs to automatically generate high-quality questions following knowledge instructions. ***Flexible answer space***. The answers in KORC are labels in knowledge bases, rather than choices or spans from the text. In addition, questions in KORC have an in-determinant number of answers (*e.g.,* Question 2 in Figure 1). We propose two new metrics to facilitate easy evaluation of the variable number of answers.

KORC is constructed based on reasoning chains that weave together documents and background knowledge base. We provide three versions of KORC based on data annotation methods. They are KORC-T from **T**emplate-based generation, KORC-H from **H**uman annotation, and KORC-L from **LLM** annotation. The final version of KORC contains $9,074$ documents and $31,804$ questions. We establish the initial baselines for KORC. We find that even the strongest baseline model only achieves $68.3\%/30.0\%$ P-F1 (ID / OOD) on KORC-H, indicating that KORC brings new challenge to natural language understanding. We also find that LLM-annotated questions in KORC-L provide moderate supervision to answer human-generated questions in KORC-H, which suggests that models can be appropriately instructed to train themselves. The KORC dataset and codes for our baseline models will be released upon acceptance.

2 Task Definition

KORC shares a similar task format with traditional machine reading comprehension (MRC). The input includes a document d and a natural language question q. Models are required to output the answer $\mathbf{a}$ to the question after reading the document.

Different from traditional MRC tasks, KORC presents two key highlights. Firstly, KORC is augmented with an extra background knowledge base (KB), denoted as $\mathcal{K}$. Each semantic triple in the background KB $(e_h, r, e_t) \in \mathcal{K}$ describes the relation r between the head entity e_h and tail entity e_t. The questions cannot be answered solely within the document or the background KB, but a combination of the two. Readers need to reconstruct the reasoning chains, which weaves the document and the background KB together, to find the answers. Secondly, answers are an in-determinant number of entities in the background KB, *i.e.,* $\mathbf{a} = \{e_i | e_i \in \mathcal{K}\}$, $|\mathbf{a}| \geq 1$. Models are encouraged to output neither excessive nor insufficient predictions.

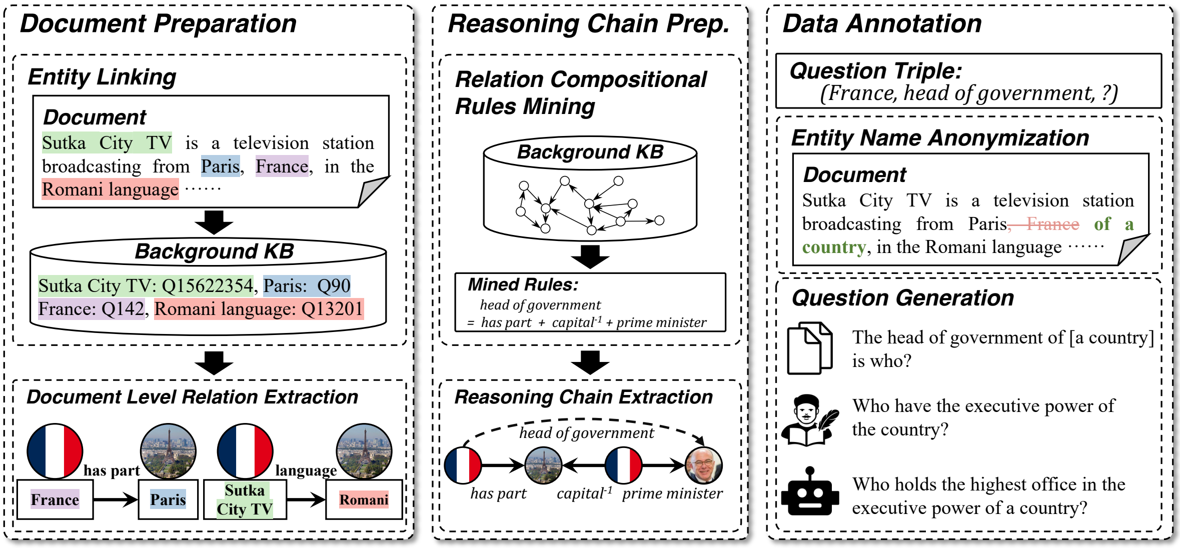

Document Preparation
Entity Linking
Document
Sutka City TV is a television station broadcasting from Paris, France, in the Romani language ······
Background KB
Sutka City TV: Q15622354, Paris: Q90
France: Q142, Romani language: Q13201
Document Level Relation Extraction
has part
language
France
Paris
Sutka City TV
Romani
Reasoning Chain Prep.
Relation Compositional Rules Mining
Background KB
Mined Rules:
head of government
= has part + capital-1 + prime minister
Reasoning Chain Extraction
head of government
has part
capital-1
prime minister
Data Annotation
Question Triple:
(France, head of government, ?)
Entity Name Anonymization
Document
Sutka City TV is a television station broadcasting from Paris, ~~France~~ of a country, in the Romani language ······
Question Generation
The head of government of [a country] is who?
Who have the executive power of the country?
Who holds the highest office in the executive power of a country?

Figure 2: The overall data collection process. In the data annotation step, we also show three real annotation cases from template-based generation, human annotation, and LLM annotation.

3 Dataset Construction

KoRC requires joint reasoning over text and background KB. It is constructed in three steps: (1) We prepare documents and align them to the background KB via entity linking and document level relation extraction; (2) We prepare reasoning chains that weave documents and background KB together. We first mine massive relation compositional rules from the background KB and then extract reasoning chains accordingly. (3) We annotate data by anonymizing the question entity e_q in the document to prevent reasoning shortcut and generate questions based on the reasoning chains. We design three different methods to annotate the data—template-based generation, human annotation, and large language model annotation. Figure 2 demonstrates the overall data construction process.

3.1 Step 1: Document Preparation

To provide broad knowledge coverage and facilitate knowledge reasoning, we sample documents from Wikipedia as the reading material and use Wikidata5M (Wang et al., 2021), a subset of Wikidata (Vrandecic and Krötzsch, 2014) consisting of all the entities in Wikipedia, as the background KB. To align documents from Wikipedia to Wikidata, we need to identify entity mentions in the documents and link them to their entity ID in Wikidata5M (*i.e.,* entity linking). We also need to extract semantic triples from the documents, which are weaved into the reasoning chains in Step 2.

Fortunately, DocRED (Yao et al., 2019) provides a large batch of documents from Wikipedia with extracted semantic triples. Specifically, each document in DocRED is released with extracted entity mentions and relations among the mentions, which comprise semantic triples. These semantic triples are manually annotated, which have a higher quality than algorithms-extracted ones. For entity linking, we first link mentions to Wikipedia entities via the existing hyperlink, or use the entity linking toolkit pre-trained on Wikipedia—BLINK (Wu et al., 2020). Then we use XLORE (Jin et al., 2019) to link Wikipedia entities to Wikidata entities. In total, $3,291$ documents with valid entity linking results in the training set and validation set of DocRED are used under the grant of MIT License.

3.2 Step 2: Reasoning Chain Preparation

A reasoning chain is a list of entities connected by their relations, denoted as $(e_q, r_1, e_1, \cdots, r_n, e_n)$. In particular, the reasoning chain starts from the document and ends at the background KB, which means $e_q \in d, e_n \in \mathcal{K}$. The reasoning chain deduces into a question triple $(e_q, r, ?)$ according to the compositionality of the relations, *i.e.,* $r = r_1 + \cdots + r_n$. The question triple can be paraphrased into natural language questions like *"Which entities have relation* r *with the question entity* e_q*?"*, such that e_n serves as the answer. To this end, we (1) mine relation compositional rules from massive semantic triples, and then (2) extract reasoning chains from the documents and the background KB according to the compositional rules.

Relation Compositional Rule Mining. Compositional rules of relations are induced from large-scale semantic triples in the background KB. We use BIMR (Lv et al., 2021), which provides high-quality compositional rules from human annotation. We supplement more rules mined by AnyBURL (Meilicke et al., 2019) from the background KB to further increase knowledge coverage.

Reasoning Chain Extraction. For semantic triple (e_q, r_1, e_1) extracted from document, if a compositional rule $r = r_1 + \cdots + r_n$ exists, we construct the reasoning chain $(e_q, r_1, e_1, \cdots, r_n, e_n)$ and its corresponding question triple $(e_q, r, ?)$. The resulting reasoning chain satisfies that e_q and e_1 are mentioned in the document, ***i.e.,*** $e_q, e_1 \in d$, and e_i are entities in the background KB, ***i.e.,*** $e_i \in \mathcal{K}, i \geq 1$. e_1 serves as the bridge entity between the document and the background KB.

It is worth noting that we filter out reasoning chains which end at the document, ***i.e.,*** $e_n \in d$, to prevent the reasoning process bypassing the background KB. The end entity e_n is identified from the document via entity linking.

3.3 Step 3: Data Annotation

Data annotation aims to (1) anonymize the question entity e_q mentioned in the document to prevent reasoning shortcut and (2) generate questions about the anonymized question entity.

In question entity name anonymization, reasoning shortcut means that the document is bypassed and questions can be answered without reading the document. For example, the answer of questions like *What is the official language of France?* does not require the document as in Figure 1. Thus, we substitute the mentions of e_q in the document with their anonymized name and polish the document to fluency. Question name anonymization requires ***anonymity*** and ***uniqueness***. Anonymity prunes reasoning shortcut and avoids answer leakage. Uniqueness guarantees that the anonymized name does not refer to other entities mentioned in the text.

The question generation process requires ***consistency*** and ***diversity***. Semantic information of the natural language question should be consistent with its corresponding question triple. Besides, diverse syntactic structures for the same relation in different question triples are desired. For example, question triples $(e_q, r, ?)$, where r=*"birth place"* can be converted into *"Where was e_q born?"* and *"In which place did e_q see the first sunrise of his life?"*. These two questions expect similar answers though differ in syntactic.

We design 3 different methods to accomplish the data annotation following the above principles.

Template-based Generation. For question entity anonymization, we substitute entity mentions with their most fine-grained class name in Wikidata. We also add a unique suffix to the class name to guarantee uniqueness so that it will not refer to entities in the document of the same class. For question generation, we manually annotate $1-4$ question templates for each relation, which has a placeholder for the question entity. Given a question triple $(e_q, r, ?)$, the questions are generated via substituting the placeholder in the template of relation r with the anonymized entity name for e_q. We provide example templates in Appendix A.1.

Human Annotation. We recruit annotators, who has at least passed Test for English Majors-Band 4 (TEM-4) to annotate the data. We train them to make sure they are aware of the aforementioned data annotation principles. We implement a visualized annotation platform to assist the data annotation process, as shown in Appendix A.2.2.

Large Language Model Annotation is inspired by the success of LLMs in generating datasets (Liu et al., 2022a). We prompt LLM with demonstrations (Liu et al., 2022b; Brown et al., 2020) and instructions (Sanh et al., 2022; Wei et al., 2022) to anonymize the question entity, generate questions, and conduct quality inspection. The provided demonstrations include 2 manually annotated examples for anonymization and questions. In particular, we implement the LLM with `text-davinci-003`, a variant of GPT-3 (Brown et al., 2020). Prompts are shown in Appendix A.3.

After dataset construction, we obtain a total of $9,086$ documents after anonymization and $31,804$ questions. Notice that each document could have more than one question entities. They are thus paraphrased into multiple different documents after anonymization. According to the data annotation method, we present three versions of KoRC, namely KoRC-T (Template-based generation), KoRC-H (Human annotation), and KoRC-L (LLM generation). We consider KoRC-H as the standard subset of KoRC.

4 Dataset Analysis

We perform a detailed analysis of KoRC. We first design two evaluation metrics where the number of answers are in-determinant. Then, we investigate sophisticated data splitting strategy. Finally, we conduct comprehensive analysis with regard to the data distribution in KoRC.

4.1 Evaluation Metric

We extend exact match accuracy and f1 measure to evaluate machine reading comprehension performance from Rajpurkar et al. (2016) by introducing penalized exact match accuracy (P-ACC) and penalized f1 measure (P-F1). Since the answer is a set of entities, the metrics need to match the predictions to the ground truth answers with Hungarian algorithm using editing distance. We define a penalty term in case that the model outputs excessive or insufficient predictions:

$$\text{penalty} = \frac{\min\{\#\text{prediction}, \#\text{label}\}}{\max\{\#\text{prediction}, \#\text{label}\}}$$

P-ACC and P-F1 are defined by multiplying the penalty term with the mean accuracy and F1 measure of each matched predictions, respectively.

4.2 Data Split

We are mainly concerned with three issues in splitting the data. (1) The training set should be sufficient to train a modern MRC model until convergence; (2) The test set should avoid any possible data leakage; (3) How to split the test set into in-distribution (ID) subset and out-of-distribution (OOD) subset for more detailed evaluation?

Training Data Sufficiency. We conduct pilot experiment on KoRC-H with BART-base. We vary the ratio of questions from 10% to 70% for training and use 30% of held-out questions for both validating and testing. The performance curve is shown in Figure 4, which flattens after 50%. Thus, we use 50% for training.

Leakage Avoidance. In the test set, for documents that have multiple question entities, we randomly select one question entity and keep it along with its questions. The remaining question entities are discarded with their associated questions. This strategy avoids possible leakage of the name of the anonymized entities.

Test Set Splitting. Questions in the test set are labeled as ID (OOD) when its question triple $(e_q, r, ?)$ does (not) appear in the training set. OOD questions are more challenging than ID questions.

4.3 Statistic Analysis

The general statistics of KoRC is shown in Table 1. Answers require reasoning chains of an average of 2.80 hops to reach the answer beyond the document, including the chains within the document. Figure 3 compares the prefix trigram pattern among different ways of data annotation in Step 3. It shows that human annotated questions provides the best diversity compared to template based questions and LLM generated questions. Although LLM annotated questions show lower diversity than template generated questions, we find that LLM can occasional spark novel questions, as the examples shown in Figure 2.

5 Experiments

We establish the initial baselines for KoRC and use KoRC to analyze the deep text understanding ability of these baseline models. More experiments, analysis, and benchmark results are included in the project repository.

5.1 Baseline Models

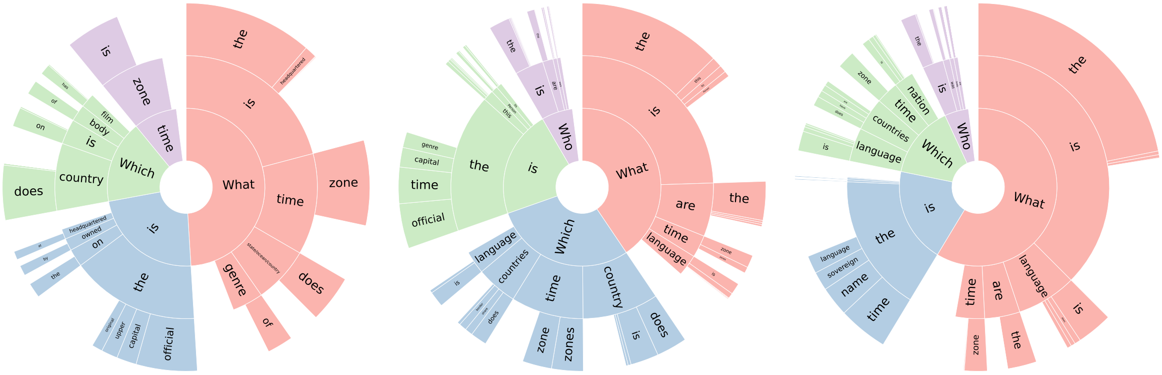
We design and implement the initial baselines in the following 4 categories.

Fine-tuned Language Models. It has been shown that pre-trained language models are rich in knowledge (Petroni et al., 2019; AlKhamissi et al., 2022). Fine-tuning on dataset that requires knowledge reasoning (Talmor et al., 2020; West et al., 2022) elicit the knowledge within LMs. We view KoRC as a sequence-to-sequence task, which can be directly processed by an encoder-decoder language model, such as **BART-base** (Lewis et al., 2020a) and **Flan-T5-base** (Chung et al., 2022). We also train and evaluate **Flan-T5-XXL** (Chung et al., 2022), which scales up to 11B parameters and is trained with task descriptions. Particularly, the input of the encoder is a concatenation of the anonymized document and the question. The answers are output as coma separated entity labels.

In-Context Learning (ICL) Prompting. Prompting is another thread of attempts that stimulate the pre-trained language models to perform complex reasoning task without tuning. To construct prompts, we use examples in the training set as demonstrations. The demonstration examples are dynamically selected according to sentence similarity of the question and its associated document, which is computed with sentence embedding model MPNet (Song et al., 2020). We

Split	Train	Valid	Test-ID	Test-OOD	All
#Document (Unique)	7, 260 (2, 332)	4, 637 (2, 074)	546 (546)	516 (516)	9, 086 (3, 291)
#Relation (Unique)	208 (117)	185 (113)	121 (90)	162 (111)	212 (119)
#Question	18, 945	7, 574	3, 432	1, 853	31, 804
Average Hops per Answer	2.80	2.80	2.84	2.81	2.80

Table 1: Statistics of the final version of KoRC. Unique documents is the number of documents before anonymization. Unique relation considers the inverse relation the same as the forward relation. They are shown in the parenthesis.



(a) KoRC-T

(b) KoRC-H

(c) KoRC-L

Figure 3: Distribution of trigram prefixes of questions in KoRC-T, KoRC-H, and KoRC-L.

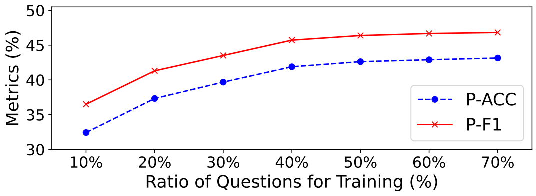


Figure 4: Training curve.

implement in-context learning prompting with **GPT-3** (Brown et al., 2020) (`text-davinci-002`) and **GLM-130B** (Zeng et al., 2022).

Retrieval Augmented Models. There are opinions on language models alone being insufficient to answer knowledge intensive questions. To facilitate reasoning requiring knowledge beyond the input text, they propose to augment language models with an external retrieval module, which searchs for the background knowledge from the open-domain Internet, such as RAG (Lewis et al., 2020b). We test on **RAG-seq**, which generates intermediate answers with multiple searching results and synthesis them into the final answer, and **RAG-token**, which synthesis the searching results and generate the answer. In KoRC, we use the document and the question to search for knowledge and mingle the original document with the searching results to generate the answer.

Joint Reasoning over Text and KB. These methods align document and questions to the background KB (*i.e.,* Wikidata5M) and perform the knowledge reasoning on the background KB. **EmbedKGQA** (Saxena et al., 2020) converts documents and questions into vectors in the embedding space of the background KB and performs the knowledge reasoning with operations on the embedding vector, where we use ComplEx (Trouillon et al., 2016). We also implement EmbedKGQA with trainable knowledge representations (**EmbedKGQA***). However, limited by computational memory, we only use a subset of the background KB with entities recalled by entity linking. **TransferNet** (Shi et al., 2021) uses documents and questions as attention queries in GAT (Veličković et al., 2018) to perform explicit knowledge reasoning on the background KB.

5.2 Main Results

Table 2 shows all the baseline results on KoRC-H—the standard subset of KoRC. The strongest baseline achieves 52.8% average P-ACC and 55.8% average P-F1 by Flan-T5-XXL, which suggests that fine-tuned large language models have strong capability to use background knowledge. RAG-seq and EmbedKGQA also achieve competitive performance, which have the ability to retrieve background knowledge from the open-domain Internet or access the background KB. Although language

KoRC-H	P-ACC			P-F1		
	ID	OOD	Mean	ID	OOD	Mean
BART-base	50.3	24.9	41.4	52.9	30.2	44.9
Flan-T5-base	33.5	24.0	30.2	35.8	27.5	32.9
Flan-T5-XXL	**63.8**	**32.3**	**52.8**	65.8	**37.2**	**55.8**
GPT-3	18.2	24.6	20.5	22.2	30.2	25.0
GLM-130B	9.9	14.9	11.6	12.7	18.8	14.8
RAG-seq	61.7	25.9	49.2	63.7	30.0	51.9
RAG-token	57.4	23.5	45.5	59.1	27.2	47.9
EmbedKGQA	61.2	21.9	47.4	**68.3**	28.9	54.5
EmbedKGQA*	34.0	13.6	26.9	41.6	21.8	34.6
TransferNet	32.7	12.9	25.8	37.7	16.6	30.3

Table 2: Baseline results on KoRC-H. Baseline results on KoRC-L and KoRC-L are shown in Appendix C.

BART-base	KoRC-T	KoRC-H	KoRC-L
KoRC-T	**48.7**	39.4 (9.3 ↓)	37.5 (11.2 ↓)
KoRC-H	41.7 (3.2 ↓)	**44.9**	40.8 (4.1 ↓)
KoRC-L	40.7 (6.4 ↓)	42.3 (4.8 ↓)	**47.1**
GPT-3	KoRC-T	KoRC-H	KoRC-L
KoRC-T	**24.5**	23.6 (0.9 ↓)	23.2 (1.3 ↓)
KoRC-H	23.7 (1.3 ↓)	**25.0**	24.9 (0.1 ↓)
KoRC-L	23.0 (0.9 ↓)	23.8 (0.1 ↓)	**23.9**
RAG-seq	KoRC-T	KoRC-H	KoRC-L
KoRC-T	**51.3**	40.8 (10.5 ↓)	38.6 (12.7 ↓)
KoRC-H	46.5 (5.4 ↓)	**51.9**	47.9 (4.0 ↓)
KoRC-L	46.7 (8.2 ↓)	48.1 (6.8 ↓)	**54.9**
EmbedKQGA	KoRC-T	KoRC-H	KoRC-L
KoRC-T	**58.5**	44.1 (14.4 ↓)	38.5 (20.0 ↓)
KoRC-H	53.6 (0.9 ↓)	**54.5**	47.8 (6.7 ↓)
KoRC-L	49.5 (6.0 ↓)	51.5 (4.0 ↓)	**55.5**

Table 3: Cross evaluation results among KoRC-T, KoRC-H, and KoRC-L in terms of P-F1 (%) averaged over IID set and OOD set. The left most column shows where the training data are from.

model pre-training brings large-scale knowledge into the model, ICL prompted LLMs do not provide a satisfactory performance on KoRC, which indicates that precise recalling of background knowledge plays a key role in answering our questions. These results show that KoRC serves its designing purpose to test deep text understanding skills.

Evaluation results show a performance drop around $20\% - 40\%$ from ID set to OOD set on KoRC-H. This discrepancy suggests that these models mainly learn to ***remember*** the answers, rather than ***generalize*** to different query triples. Meanwhile, knowledge representation based EmbedKGQA is superior or comparable to knowledge retrieving based RAG-seq on ID sets while it is outmatched on OOD sets. This occurs because knowledge representations are constructed based on relation compositional rules, thus easy to overfit the ID questions. Splitting the test set in KoRC provides a new way to evaluate the true deep text understanding skills.

ICL prompted LLMs are observed to perform better on the OOD set than the ID set. This counter-intuitive result is caused by the notorious repetition problem (Xu et al., 2022). ID shares a similar distribution to the training set so LLMs directly copy the results from the demonstrations, while the OOD set urges the model to think independently. Another abnormal model is EmbedKGQA*. Although its knowledge representation can be updated, it falls short of EmbedKGQA by a large margin due to its limited background knowledge that can be held into the random access memory of GPUs, which further reflects the broad knowledge coverage of KoRC.

5.3 Cross Evaluation

We conduct cross evaluation among KoRC-T, KoRC-H, and KoRC-T to verify whether automatically generated questions can be used as distant supervision to learn deep text understanding skills. In particular, we train models on one of the three versions of datasets, and evaluate on the test set of all the three versions. Cross evaluation results are shown in Table 3.

As expected, all the cross evaluation results drop compared to the those where training data and test data are produced by the same data annotation method. Nevertheless, among all the three versions, KoRC-H brings more sophisticated deep text understanding skills to the model, with even as marginal as a 0.9% performance drop for EmbedKGQA on KoRC-T in terms of average P-F1. This is attributed to the diversity of the questions generated by our annotators. Meanwhile, training on KoRC-L only results in a moderate performance drop on KoRC-T and KoRC-H. By contrast, models trained on KoRC-T struggle with test questions in KoRC-H and even KoRC-L. This suggests a feasibility to instruct LLMs with massive real-world knowledge to generate high-quality questions. These questions can then be used as distant supervision to train models to achieve deep language understanding.

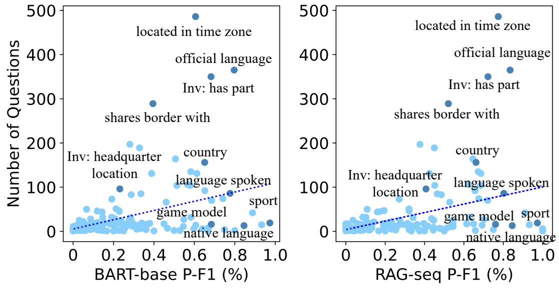


Figure 5: Error analysis. Each point corresponds to a relation with its number of questions in KoRC and average P-F1 recorded on BART-base and RAG-seq. The dashed lines indicate linear regression results. We highlight and label several representative relations.

KoRC-H	Original	-Document	-Anon.
BART-base	44.9	24.5 (20.4 ↓)	55.1 (10.2 ↑)

Table 4: Ablation results on KoRC-H with BART-base in terms of P-F1 (%) averaged over IID and OOD sets.

5.4 Analysis

We further conduct empirical analysis on KoRC, including error analysis and ablation study.

Error Analysis. Each question in KoRC-H corresponds to a question triple $(e_q, r, ?)$, which contains a relation r. We examine the error distribution with regard to relations. Figure 5 plots the scatter charts for each relation in KoRC. Each point represents a relation with its question number and average P-F1 on BART-base and RAG-seq.

To better demonstrate the correlation between question number and P-F1, we run least square error regression and show in dashed line. The regression results indicate the trend that relations with fewer questions (long tail relations) are more difficult than relations with abundant questions. However, there are outlier relations scattered in the top left (bottom right) corner, which means they have many (few) questions in KoRC that are difficult (easy) to answer. We label a few of these outlier relations in Figure 5. We find that top-left-relations are mostly equipped with multiple answers. For example, questions involving the inverse relation of *headquarter location* usually ask *Which organizations are headquartered in this place?* are difficult to recall all the correct answers. For the bottom-right relations, they usually construct single-answer questions, such as *native language* and *sport*.

Ablation. We remove documents from KoRC-H, which makes KoRC-H degenerate into a question answering benchmark. We also experiment whether the entity name will result in reasoning shortcut without anonymization. The original name of the question entity is appended to the document. Table 4 shows the ablation study results.

We find that removing document significantly undermines the results of BART-base with a performance drop at 20.4% in P-F1. This shows that text information is indispensable in KoRC. Readers are not encouraged to directly answer the questions without reading the given document. When we provide the entity name as part of the reading material, the P-F1 of BART-base increases from 44.9% to 55.1%. This shows that entity name contains direct clues to answering the question and annotating anonymized entity name cannot be omitted.

6 Related Work

Machine Reading Comprehension. Devising intelligent systems to answer questions on knowledge in text form has long been a challenge in Natural Language Understanding (NLU) (Welbl et al., 2018), and the MRC task plays an important part in evaluating NLU (Ho et al., 2022). Abundant datasets have been proposed to advance research in MRC. One of the earliest work is MCTest (Richardson et al., 2013), a multiple-choice reading comprehension dataset. Following works have surged to advance more challenging text understanding with more complicated answer formats. Based on the answer format, MRC datasets can by grouped into four types: span extraction (Hewlett et al., 2016; Welbl et al., 2018; Amouyal et al., 2022), multiple-choice (Sun et al., 2019; Tafjord et al., 2019; Huang et al., 2019; Amouyal et al., 2022), cloze style (Mostafazadeh et al., 2016), and free-form (Khashabi et al., 2018) answer.

Deep Text Understanding. Background knowledge integration is regarded as the key ingredient of deep text understanding. Different kinds of background knowledge have been employed, such as commonsense knowledge (e.g., ATOMIC (Sap et al., 2019)), and world knowledge (e.g., Wikidata (Vrandecic and Krötzsch, 2014)). Representative works include WikiReading (Hewlett et al., 2016) which aims to predict textual values from Wikidata by reading the corresponding Wikipedia text, DREAM (Sun et al., 2019) whose questions requires unspoken commonsense knowledge, QuARTz (Tafjord et al., 2019) that requires understanding and applying qualitative knowledge, and CosmosQA (Huang et al., 2019) that requires contextual commonsense reasoning.

Compared with the existing datasets, KoRC is constructed with the instruction from real-world large-scale knowledge base. The answers of our KoRC are labels in the knowledge bases, and the number of answers is in-determinant, challenging MRC more. Most importantly, both the reading materials and external background knowledge are indispensable for every question in KoRC, which prevents reasoning shortcut effectively.

7 Conclusion

In this paper, we propose a new benchmark—KoRC for deep text understanding with broad knowledge coverage and flexible answer format. Our contributions are not only the dataset itself, but also we demonstrate the feasibility to guide LLMs to generate deep text understanding questions with the help of large-scale background KB. Our baseline experiments demonstrates to which extent existing powerful models can leverage background knowledge to understand passages by trying to solve KoRC. In the future, we plan to extend KoRC to more complicated knowledge, such as literal knowledge and qualifier knowledge in common knowledge bases. It is intriguing to design more skillful reader models via connecting the document with background knowledge.

Limitations

We propose and construct KoRC as a new benchmark dataset for deep text understanding. The limitations are two folds. First, in the benchmark design, KoRC do not take more complicated knowledge into consideration, including literal knowledge and qualifier knowledge. We leave extending KoRC to these knowledge in future work. Second, in the dataset construction, we examine automatic name anonymization and question generation strategy, and present KoRC-L. KoRC-L relies on large language models. Rather than medium-scaled language models that can be maintained by a single machine, GPT-3 is used via its online APIs. Although the service of GPT-3 is currently available, we still need to find a substitution for better reproducibility. Besides, although LLM saves human effort, the execution of LLMs potentially consumes more energy power. It would be better if we can preserve the high question generation quality and propose a small model to proceed data annotation.

Ethics Statement

Our proposed dataset, KoRC, is constructed with the knowledge guidance from Wikidata. As a crowd-sourced knowledge base, it is possible that Wikidata contains bias knowledge and even poisonous information. For example, Wikidata contains more information in the English. It is possible that KoRC also inherit the bias from Wikidata. Another ethical concern raises from the payment of our annotators. All the annotators are payed equally according to the number of documents and questions they annotated. We hope that KoRC can be properly used to guide the development of deep text understanding models after we release it.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona T. Diab, and Marjan Ghazvininejad. 2022. A review on language models as knowledge bases. *CoRR*, abs/2204.06031.

Samuel Joseph Amouyal, Ohad Rubin, Ori Yoran, Tomer Wolfson, Jonathan Herzig, and Jonathan Berant. 2022. QAMPARI: : An open-domain question answering benchmark for questions with many answers from multiple paragraphs. *CoRR*, abs/2205.12665.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Anne Castles, Kathleen Rastle, and Kate Nation. 2018. Ending the reading wars: Reading acquisition from novice to expert. *Psychological Science in the Public Interest*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Philip B Gough and William E Tunmer. 1986. Decoding, reading, and reading disability. *Remedial and special education*.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WikiReading: A novel large-scale language understanding task over Wikipedia. In *ACL*.

Xanh Ho, Johannes Mario Meissner, Saku Sugawara, and Akiko Aizawa. 2022. A survey on measuring and mitigating reasoning shortcuts in machine reading comprehension. *ArXiv*, abs/2209.01824.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *EMNLP*.

Hailong Jin, Chengjiang Li, Jing Zhang, Lei Hou, Juanzi Li, and Peng Zhang. 2019. XLORE2: large-scale cross-lingual knowledge graph construction and application. *Data Intelligence*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *EMNLP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. WANLI: worker and AI collaboration for natural language inference dataset creation. *CoRR*, abs/2201.05955.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *DeeLIO*.

Xin Lv, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Yichi Zhang, and Zelin Dai. 2021. Is multi-hop reasoning really explainable? towards benchmarking reasoning interpretability. In *EMNLP*.

John McCarthy. 1976. An example for natural language understanding and the ai problems it raises. *Formalizing Common Sense: Papers by John McCarthy*, 355.

Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Anytime bottom-up rule learning for knowledge graph completion. In *IJCAI*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL*.

Peter Norvig. 1987. *A Unified Theory of Inference for Text Understanding*. Ph.D. thesis, EECS Department, University of California, Berkeley.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *EMNLP*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *ICLR*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI Conference on Artificial Intelligence*.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *ACL*.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. TransferNet: An effective and transparent framework for multi-hop question answering over relation graph. In *EMNLP*.

Reid Smith, Pamela Snow, Tanya Serry, and Lorraine Hammond. 2021. The role of background knowledge in reading comprehension: A critical review. *Reading Psychology*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *NeurIPS*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *TACL*.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2020. Investigating prior knowledge for challenging Chinese machine reading comprehension. *TACL*.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. QuaRTz: An open-domain dataset of qualitative relationship questions. In *EMNLP*.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In *NeurIPS*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *ICLR*.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *TACL*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *ICLR*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *NAACL*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir R. Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *CoRR*, abs/2201.05966.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. 2022. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. In *NeurIPS*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *ACL*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2022. GLM-130B: an open bilingual pre-trained model. *CoRR*, abs/2210.02414.

A Data Annotation Details

A.1 Question Templates

In Section 3.3, we introduced three different ways to annotate data. They are template-based generation, human annotation, and LLM generation. Here we supplement more technical details on these three methods.

A.2 Human Annotation Details

A.2.1 Annotator Recruiting

We recruit professional annotators who have English as their second language. These annotators are employees of data provider. All the annotators working for KoRC-H have passed Test for English Majors-Band 4 (TEM-4). In particular, TEM-4 is a national Test for students majoring in English in the end of their second year at university in China. This qualification ensures that they can correctly read our document, paraphrase the document after anonymization, and write fluent questions according to the question triples.

A.2.2 Annotation Platform

We design visualized annotation platform to help annotators to better annotate data. The annotation platform aims to (1) track editing history and (2) provide knowledge information such as anonymization name recommendations.

Entity Name Anonymization. Figure 6 shows the screenshot of our GUI for entity name anonymization. The annotators are asked to anonymize the question entities by modify the input box right below "Document After Anonymization". We provide information, including question entity names, entity mentions, and recommended anonymization name in colored cards. Annotators could easily identify which spans are deleted (marked by red background) and which spans are newly added (marked by green background). In the screenshot, we delete span [country_2] and add span of a country .

Question Annotation. Figure 7 shows the screenshot for question annotation. The annotators are provided with the question triple and the corresponding answers. They are required to write questions accordingly.

A.3 Prompt Design for LLM Annotation

We use in-context learning to instruct LLMs, where we use GPT-3, to proceed data annotation. For entity name anonymization, we provide LLM with the class name of the question entity and ask LLM to select the optimal class name, which will not leak any information to the answer, to paraphrase the document. For question generation, we first instruct LLM to generate multiple candidate questions. Then, we design another instruction to select the optimal questions, which is similar to the quality control step in data engineer.

Question Generation. Prompts for question generation are shown in Table 6 and Table 7. Notice that for question triples involving forward relations and inverse relations, we design different prompts. They are mainly different in the example.

Question Selection. For question selection, we provide LLM with all the questions generated from previous step. The quality control protocals are included in the instructions, as shown in Table 8.

B Experiment Implementation Details

B.1 In-Context Learning Prompt

The ICL prompt consists of two parts. First, we give the task description in the instruction. Then, we provide 4 demonstration examples. The overall prompts are shown in Table 9.

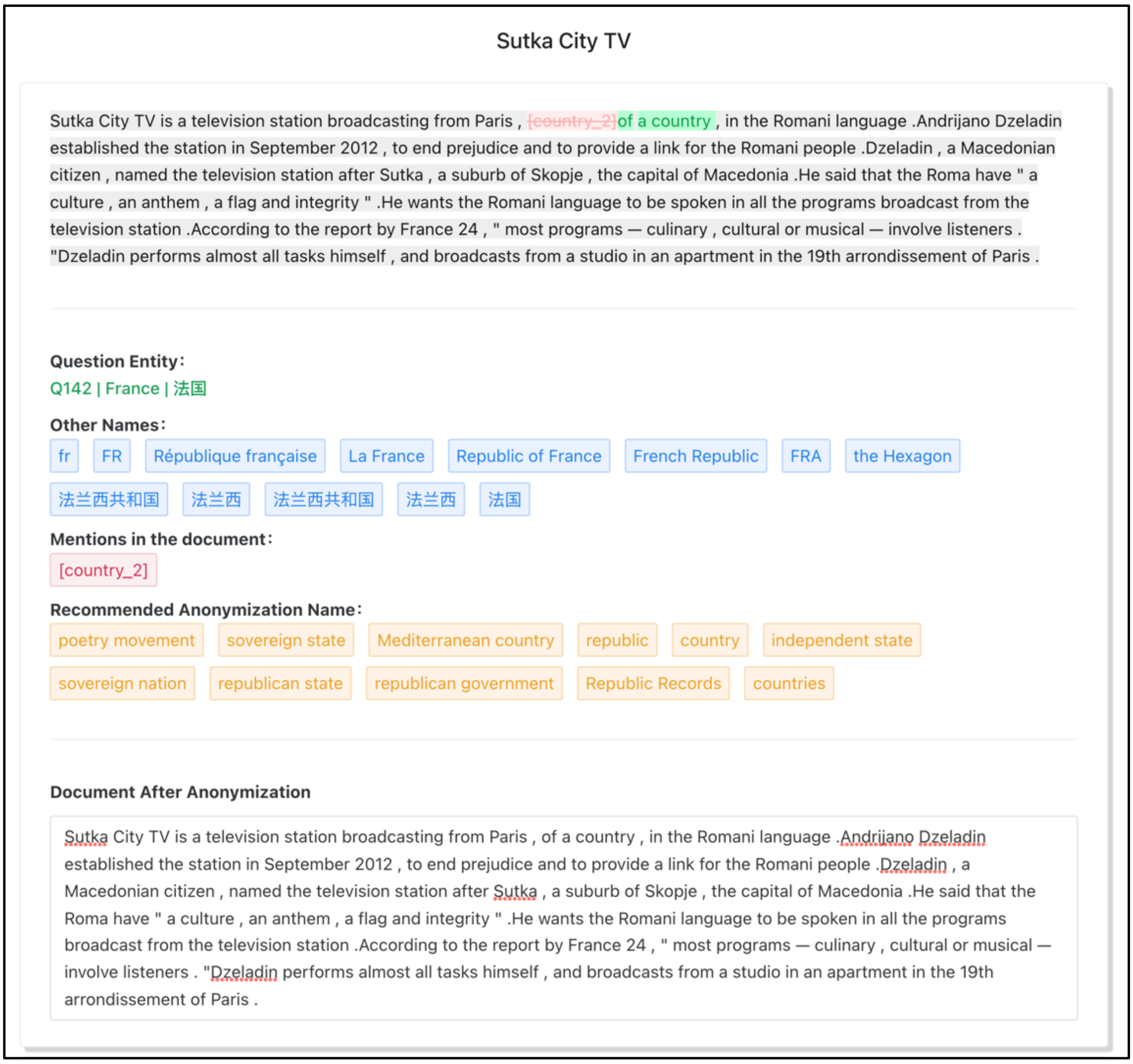
C Supplementary Experiments

We evaluate our baseline models on KoRC-T, KoRC-H, and KoRC-L. The results are shown in Table 10, as a supplementation to Table 2.

We observe that KoRC-T, as a template-generated dataset, is the simplest among all the three versions. Baselines generally achieve higher performance on KoRC-T compared to KoRC-H and even KoRC-L. We also find that LLMs failed to successfully answer questions generated by themselves on KoRC-L. This is because the questions are generated according to external knowledge guidance beyond LLM itself.

Relation Direction	Relation Label	Template
Forward	r = *member of political party*	What political party was $[e_q]$ a member of? Which political party does $[e_q]$ belong to?
	r = *place of burial*	Where is the burial place of $[e_q]$? Where was $[e_q]$ buried after his/her death?
	r = *cast member*	$[e_q]$ is a cast member of which movie? What movies or work has $[e_q]$ been in?
	r = *country of citizenship*	Which country does [x] come from? What nationality does [x] hold?
Forward	r = *Inv: producer*	Which work is produced by $[e_q]$? Which work did $[e_q]$ produce?
	r = *Inv: parent organization*	Whose parent organization is $[e_q]$? Which subsidiaries does $[e_q]$ have?

Table 5: Example question templates for data annotation of KoRC-T.



Sutka City TV

Sutka City TV is a television station broadcasting from Paris , ~~[country_2]~~of a country , in the Romani language .Andrijano Dzeladin established the station in September 2012 , to end prejudice and to provide a link for the Romani people .Dzeladin , a Macedonian citizen , named the television station after Sutka , a suburb of Skopje , the capital of Macedonia .He said that the Roma have " a culture , an anthem , a flag and integrity " .He wants the Romani language to be spoken in all the programs broadcast from the television station .According to the report by France 24 , " most programs — culinary , cultural or musical — involve listeners . "Dzeladin performs almost all tasks himself , and broadcasts from a studio in an apartment in the 19th arrondissement of Paris .

Question Entity:
Q142 | France | 法国

Other Names:
fr | FR | République française | La France | Republic of France | French Republic | FRA | the Hexagon | 法兰西共和国 | 法兰西 | 法兰西共和国 | 法兰西 | 法国

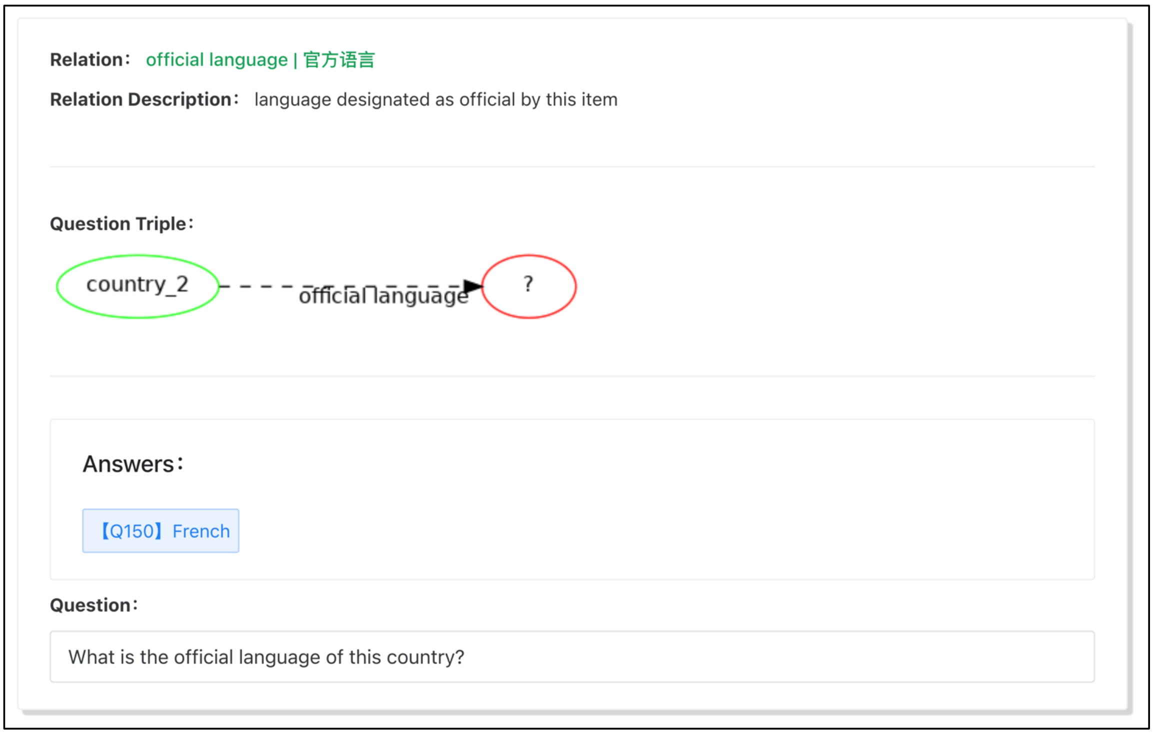
Mentions in the document:
[country_2]

Recommended Anonymization Name:
poetry movement | sovereign state | Mediterranean country | republic | country | independent state | sovereign nation | republican state | republican government | Republic Records | countries

Document After Anonymization

Sutka City TV is a television station broadcasting from Paris , of a country , in the Romani language .Andrijano Dzeladin established the station in September 2012 , to end prejudice and to provide a link for the Romani people .Dzeladin , a Macedonian citizen , named the television station after Sutka , a suburb of Skopje , the capital of Macedonia .He said that the Roma have " a culture , an anthem , a flag and integrity " .He wants the Romani language to be spoken in all the programs broadcast from the television station .According to the report by France 24 , " most programs — culinary , cultural or musical — involve listeners . "Dzeladin performs almost all tasks himself , and broadcasts from a studio in an apartment in the 19th arrondissement of Paris .

Figure 6: Screenshot of our annotation platform for entity name anonymization.



Relation: official language | 官方语言

Relation Description: language designated as official by this item

Question Triple:

Answers:

【Q150】French

Question:

What is the official language of this country?

Figure 7: Screenshot of our annotation platform for question generation.

Prompt for forward relation.

Instruction: A semantic triple describe the relation between one head entity and one tail entity. For example, Job Biden -> native language -> English is one semantic triple which means Job Biden (head entity)'s native language (relation) is English (tail entity), now you are given one incomplete semantic triple where the tail entity is missing and one hint which would tell what all the possible missing entity is. your task is to design 5 questions based on the given semantic triple and the hint to find out the missing tail entity.
Notice: the given hint could be utilized to design more accurate questions with respect to the given possible missing entities, but any part of the hint should not be contained in the generated question!

Example 1:
Input:
Question Triple: independent state F -> shares border with (countries or administrative subdivisions, of equal level, that this item borders, either by land or water. A single common point is enough.) -> missing entity
hint: possible missing entity could be: "Paraguay","Chile","Uruguay","Bolivia","Brazil"

Output:
1. Which countries does independent state F border?
2. What countries do the boundaries of independent state F touch?
3. Who are the neighboring countries of independent state F?
4. What states share a border with independent state F?
5. To which countries does independent state F have a frontier?

Example 2:
Input:
Question Triples: person F -> occupation (occupation of a person; see also "field of work" (Property:P101), "position held" (Property:P39)) -> missing entity
hint: possible missing entity could be: "actor", "singer"

Output: [*LLM output*]

Table 6: Prompt for generating questions involved with forward relation.

Prompt for inverse relation.

Instruction: A semantic triple describe the relation between one head entity and one tail entity. For example, Job Biden -> native language -> English is one semantic triple which means Job Biden (head entity)'s native language (relation) is English (tail entity), now you are given one incomplete semantic triple where the head entity is missing and one hint which would tell what all the possible missing entity is. your task is to design 5 questions based on the given semantic triple and the hint to find out the missing head entity.
Notice: the given hint could be utilized to design more accurate questions with respect to the given possible missing entities, but any part of the hint should not be contained in the generated question!

Example 1:
Input:
Question Triple: missing entity -> has part(s) (part of this subject; the inverse property of "part of" (P361). See also "has parts of the class" (P2670).) -> country A
hint: possible missing entity could be: "Northern America", "North American Football Union", "G20", "Allies of the Second World War", "Procurement G6", "North America"
Output:
1. What international organizations and events have country A participated in?
2. What international congregations and activities have the country A partaken in?
3. To what foreign associations and interactions have country A contributed?
4. What external associations and proceedings have country A been a part of?
5. What associations and episodes on the international level have country A been a part of?

Example 2:
Input:
Question Triple: missing entity -> award received (award or recognition received by a person, organisation or creative work) -> order of chivalry
hint: possible missing entity could be: "Theobald Bethmann-Hollweg", "Abdul Karim", "Abraham Moyshevich Hekkelman", "Gerald Lloyd-Verney", "Faisal of Saudi Arabia", "Peter Westmacott", "John Simon, 1st Viscount Simon", "Johan E. Mellbye", "Francisco Craveiro Lopes", "Alfred Munnings", "Vyvyan Holt", "Arthur Sullivan", "Mary Curzon, Baroness Curzon of Kedleston"

Output: [*LLM output*]

Table 7: Prompt for generating questions involved with inverse relation.

Prompt for question selection.

Instruction: You are given several questions, which share similar semantics and same answers. Their corresponding answers are also provided. Your task is to pick out the most accurate, the smoothest, the most novel question from the given questions with respect to given answers based on the given information. Notice, any part of the corresponding answers should not be contained in the selected question and the selected question should not be simply answered by "yes" or "no"!

1. What language(s) does the person speak?
2. What language(s) can the person read, write and sign?
3. What language(s) is the person familiar with?
4. What is the person's first language?
5. Does the person understand English?

Corresponding Answers: "English"

Output: [*LLM output*]

Table 8: Prompt for question selection in automatic quality control.

Prompt for in-context learning.

Instruction: you are given one document and one anonymized real-world entity with one or more mentions in the passage. Then we will ask your a question about this anonymized entity. The questions cannot be answered solely within the document or the background knowledge. Your task is to leverage world knowledge you have like Wikipedia or wikidata as background knowledge combined with the given document to answer the question related to the anonymized entity. You must output all answers in the end.

Document:"[TV show A]" is the third episode of the first season of the American comedy television series The Office. Written by Paul Lieberstein, who also acts in the show as Toby Flenderson, and directed by Ken Whittingham, the episode first aired in the United States on April 5, 2005 on NBC. In this episode, Michael (Steve Carell) is tasked with choosing a new and inexpensive health care plan. He immediately hands it off to enthusiastic volunteer Dwight (Rainn Wilson). Dwight ruthlessly cuts nearly all benefits in the new plan, angering the rest of the office staff. Meanwhile, Pam (Jenna Fischer) and Jim (John Krasinski) make up fake diseases, much to Dwight's chagrin. In an attempt to appease them, Michael promises the entire office a surprise and then spends the rest of the day scrambling to come through with his promise. The employees wait for Michael's surprise, which he awkwardly never delivers. Jenna Fischer later called "[TV show A]" her favorite season one episode. During one particular scene, Rainn Wilson kept improvising new fake diseases. The laughter that resulted in his ad-libs was not scripted, as they were in fact the cast's genuine reaction to Wilson's fake diseases. The episode received a 2.9/7 in the Nielsen ratings among people aged 18–49 garnered 5.8 million viewers overall. In addition, the episode retained 100 % of its lead - in 18–49 audience and ranked, along with the other first - season episodes of The Office, as NBC's highest - rated Tuesday night program since February 1, 2005. The episode received positive reviews.
Question: What is the series of TV show A? **Answer**: "The Office" <stop>

Here we omit other examples for better viewing.

Document: "Insane" is the twelfth episode of the third season of the American animated sitcom [TV show A]. It originally aired on the Fox network in the United States on April 8, 2001. The episode was written by Bill Odenkirk and directed by Peter Avanzino. In the episode, Fry and Bender are admitted to an insane asylum for robots after being charged for their roles in holding up a bank. Fry's attempts to convince the asylum's staff that he is a human fail; he is eventually made to believe that he is a robot, and is deemed "cured" and released from the asylum. After being released, the Planet Express crew try to make him rediscover his humanity; these attempts fail, until Fry bleeds and realizes he is in fact, human. The episode introduces the recurring [TV show A] character Roberto.
Question: What is the publisher of TV show A?
Answer: [*LLM output*]

Table 9: Prompt for question selection in automatic quality control.

KoRC-T	P-ACC			P-F1		
	ID	OOD	Mean	ID	OOD	Mean
BART-base	55.8	25.6	45.2	58.3	30.9	48.7
Flan-T5-base	40.1	25.8	35.1	42.4	29.6	37.9
GPT-3	17.3	24.8	19.9	21.2	30.6	24.5
GLM-130B	9.0	16.8	11.7	11.5	20.5	14.7
RAG-seq	60.6	**26.7**	48.7	62.1	**31.2**	51.3
RAG-token	64.0	24.2	50.0	65.9	28.4	52.7
EmbedKGQA	**66.7**	22.9	**51.3**	**73.7**	30.2	**58.5**
EmbedKGQA*	39.9	15.5	31.3	46.8	23.4	38.6
TransferNet	35.8	14.9	28.5	40.7	19.2	33.2

KoRC-H	P-ACC			P-F1		
	ID	OOD	Mean	ID	OOD	Mean
BART-base	50.3	24.9	41.4	52.9	30.2	44.9
Flan-T5-base	33.5	24.0	30.2	35.8	27.5	32.9
GPT-3	18.2	24.6	20.5	22.2	30.2	25.0
GLM-130B	9.9	14.9	11.6	12.7	18.8	14.8
RAG-seq	**61.7**	**25.9**	**49.2**	63.7	**30.0**	51.9
RAG-token	57.4	23.5	45.5	59.1	27.2	47.9
EmbedKGQA	61.2	21.9	47.4	**68.3**	28.9	**54.5**
EmbedKGQA*	34.0	13.6	26.9	41.6	21.8	34.6
TransferNet	32.7	12.9	25.8	37.7	16.6	30.3

KoRC-L	P-ACC			P-F1		
	ID	OOD	Mean	ID	OOD	Mean
BART-base	52.0	27.9	43.6	54.7	**33.1**	47.1
Flan-T5-base	36.6	26.6	33.1	38.9	30.2	35.8
GPT-3	16.4	24.1	19.1	20.5	30.3	23.9
GLM-130B	9.2	14.1	10.9	11.6	17.9	13.8
RAG-seq	**64.8**	**28.7**	**52.2**	66.7	**33.1**	54.9
RAG-token	56.8	21.8	44.5	58.6	25.7	47.1
EmbedKGQA	62.7	22.4	48.6	**69.7**	29.2	**55.5**
EmbedKGQA*	42.8	18.9	34.4	49.6	26.0	41.3
TransferNet	31.8	12.7	25.1	36.8	16.2	29.6

Table 10: Baseline results on KoRC-T, KoRC-H, and KoRC-L. EmbedKGQA* updates the knowledge representations during training, while EmbedKGQA uses freezed knowledge representations.