

Self-Supervised Knowledge Triplet Learning for Zero-shot Question Answering

Pratyay Banerjee and Chitta Baral

Department of Computer Science, Arizona State University

pbanerj6, chitta@asu.edu

Abstract

The aim of all Question Answering (QA) systems is to be able to generalize to unseen questions. Current supervised methods are reliant on expensive data annotation. Moreover, such annotations can introduce unintended annotator bias which makes systems focus more on the bias than the actual task. In this work, we propose Knowledge Triplet Learning (KTL), a self-supervised task over knowledge graphs. We propose heuristics to create synthetic graphs for commonsense and scientific knowledge. We propose methods of how to use KTL to perform zero-shot QA and our experiments show considerable improvements over large pre-trained transformer models.

1 Introduction

The ability to understand natural language and answer questions is one of the core focuses in the field of natural language processing. To measure and study the different aspects of question answering, several datasets are developed, such as SQuAD (Rajpurkar et al., 2018), HotpotQA (Yang et al., 2018), and Natural Questions (Kwiatkowski et al., 2019) which require systems to perform extractive question answering. On the other hand, datasets such as SocialIQA (Sap et al., 2019b), CommonsenseQA (Talmor et al., 2018), Swag (Zellers et al., 2018) and Winogrande (Sakaguchi et al., 2019) require systems to choose the correct answer from a given set. These multiple-choice question answering datasets are very challenging, but recent large pre-trained language models such as BERT (Devlin et al., 2018), XLNET (Yang et al., 2019b) and RoBERTa (Liu et al., 2019b) have shown very strong performance on them. Moreover, as shown in Winogrande (Sakaguchi et al., 2019), acquiring unbiased labels requires a “carefully designed crowdsourcing procedure”, which adds to the cost of data annotation. This is also quantified in other

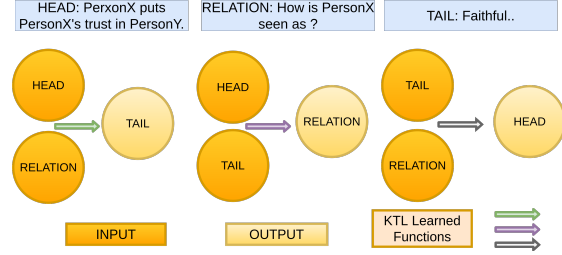


Figure 1: Knowledge Triplet Learning Framework, where given a triple (h, r, t) we learn to generate one of the inputs given the other two.

natural language tasks such as Natural Language Inference (Gururangan et al., 2018) and Argument Reasoning Comprehension (Niven and Kao, 2019), where such annotation artifacts lead to “Clever Hans Effect” in the models (Kaushik and Lipton, 2018; Poliak et al., 2018). One way to resolve this is to design and create datasets in a clever way, such as in Winogrande (Sakaguchi et al., 2019), another way is to ignore the data annotations and to build systems to perform unsupervised question answering (Teney and Hengel, 2016; Lewis et al., 2019). In this paper, we focus on building unsupervised zero-shot multiple-choice QA systems.

Recent work (Fabbri et al., 2020; Lewis et al., 2019) try to generate a synthetic dataset using a text corpus such as Wikipedia, to solve extractive QA. Other works (Bosselut and Choi, 2019; Shwartz et al., 2020) use large pre-trained generative language models such as GPT-2 (Radford et al., 2019) to generate knowledge, questions, and answers and compare against the given answer choices.

In this work, we utilize the information present in Knowledge Graphs such as ATOMIC (Sap et al., 2019a). We define a new task of Knowledge Triplet Learning (KTL) over these knowledge graphs. For tasks which do not have appropriate knowledge graphs, we propose heuristics to create synthetic knowledge graphs. Knowledge Triplet Learning

is like Knowledge Representation Learning and Knowledge Graph Completion but not limited to it. Knowledge Representation Learning (Lin et al., 2018) learns the low-dimensional projected and distributed representations of entities and relations defined in a knowledge graph. Knowledge Graph Completion (Ji et al., 2020) aims to identify new relations and entities to expand an incomplete input knowledge graph.

In KTL, as shown in Figure 1, we define a triplet (h, r, t) , and given any two as input we learn to generate the third. This tri-directional reasoning forces the system to learn all the possible relations between the three inputs. We map the question answering task to KTL, by mapping the *context*, *question* and *answer* to (h, r, t) respectively. We define two different ways to perform self-supervised KTL. This task can be designed as a representation generation task or a masked language modeling task. We compare both the strategies in this work. We show how to use models trained on this task to perform zero-shot question answering without any additional supervision. We also show how models pre-trained on this task perform considerably well compared to strong pre-trained language models on few-shot learning. We evaluate our approach on the three commonsense and three science multiple-choice QA datasets.

The contributions of this paper are summarized as follows:

- We define the Knowledge Triplet Learning over Knowledge Graph and show how to use it for zero-shot question answering.
- We compare two strategies for the above task.
- We propose heuristics to create synthetic knowledge graphs.
- We perform extensive experiments of our framework on three commonsense and three science question answering datasets.
- We achieve state-of-the-art results for zero-shot and propose a strong baseline for the few-shot question answering task.

2 Knowledge Triplet Learning

We define the task of Knowledge Triplet Learning (KTL) in this section. We define $G = (V, E)$ as a Knowledge Graph, where V is the set of vertices, E is the set of edges. V consists of entities which can be phrases or named-entities depending on the given input Knowledge Graph. Let S be a set of fact triples, $S \subseteq V \times E \times V$ with the format (h, r, t) ,

where h and t belong to set of vertices V and r belongs to set of edges. The h and t indicates the head and tail entities, whereas r indicates the relation between these entities.

For example, from the ATOMIC knowledge graph, $(\text{PersonX puts PersonX's trust in PersonY, How is PersonX seen as?, faithful})$ is one such triple. Here the head is *PersonX puts PersonX's trust in PersonY*, relation is **How is PersonX seen as?** and the tail is *faithful*. Do note V does not contain homogenous entities, i.e, both *faithful* and *PersonX puts PersonX's trust in PersonY* are in V .

We define the task of KTL as follows: Given input a triple (h, r, t) , we learn the following three functions.

$$f_t(h, r) \Rightarrow t, \quad f_h(r, t) \Rightarrow h, \quad f_r(h, t) \Rightarrow r \quad (1)$$

That is, each function learns to generate one component of the triple given the other two. The intuition behind learning these three functions is as follows. Let us take the above example: $(\text{PersonX puts PersonX's trust in PersonY, How is PersonX seen as?, faithful})$. The first function $f_t(h, r)$ learns to generate the answer t given the context and the question. The second function $f_h(r, t)$ learns to generate one context where the question and the answer may be valid. The final function $f_r(h, t)$ is a Jeopardy-style generating the question which connects the context and the answer.

In Multiple-choice QA, given the context, two choices may be true for two different questions. Similarly, given the question, two answer choices may be true for two different contexts. For example, given the context: *PersonX puts PersonX's trust in PersonY*, the answers *PersonX is considered trustworthy by others* and *PersonX is polite* are true for two different questions **How does this affect others?** and **How is PersonX seen as?**. Learning these three functions enables us to score these relations between the context, question, and answers.

2.1 Using KTL to perform QA

After learning this function in a self-supervised way, we can use them to perform question answering. Given a triple (h, r, t) , we define the following scoring function:

$$\begin{aligned} D_t &= D(t, f_t(h, r)), \quad D_h = D(h, f_h(r, t)), \\ D_r &= D(r, f_r(h, t)) \\ \text{score}(h, r, t) &= D_t * D_h * D_r \end{aligned} \quad (2)$$

where h is the context, r is the question and t is one of the answer options. D is a distance function

which measures the distance between the generated output and the ground-truth. The distance function varies depending on the instantiation of the framework, which we will study in the following sections. The final answer is selected as:

$$ans = \arg \min_t (score(h, r, t)) \quad (3)$$

As the scores are the distance from the ground-truth we select the choice that has the minimum score.

We define the different ways we can implement this framework in the following sections.

2.2 Knowledge Representation Learning

In this implementation, we use Knowledge representation learning to learn equation (1). In contrast to triplet classification and graph completion, where systems try to learn a score function $f_r(h, t)$, i.e., is the fact triple (h, r, t) true or false; in this method we learn to generate the inputs vector representations, i.e., $f_r(h, t) \Rightarrow r$. We can view equation 1 as generator functions, which given the two input encodings learns to generate a vector representation of the third. As our triples (h, r, t) can have a many to many relations between each pair, we first project the two inputs from input encoding space to a different space similar to the work of TransD (Ji et al., 2015). We use a Transformer encoder Enc to encode our triples to the encoding space. We learn two projection functions, M_{i1} and M_{i2} to project the two inputs, and a third projection function M_o to project the entity to be generated. We combine the two projected inputs using a function C . These functions can be implemented using feedforward networks.

$$\begin{aligned} I_{e1} &= Enc(I_1), I_{e2} = Enc(I_2), O_e = Enc(O) \\ I_{e1} &= M_{i1}(I_{e1}), I_{e2} = M_{i2}(I_{e2}), O_p = M_o(O_e) \\ \hat{O} &= C(I_{e1}, I_{e2}) \\ loss &= LossF(\hat{O}, O_p) \end{aligned}$$

where I_i is the input, \hat{O} is the generated output vector and O_p is the projected vector. M and C functions are learned using fully connected networks. In our implementation, we use RoBERTa as the Enc transformer, with the output representation of the $[cls]$ token as the phrase representation.

We train this model using two types of loss functions, L2Loss where we try to minimize the L2 norm between the generated and the projected ground-truth, and Noise Contrastive Estimation (Gutmann and Hyvärinen, 2010) where along with the ground-truth we have k noise-samples. These

noise samples are selected from other (h, r, t) triples such that the target output is not another true fact triple, i.e., (h, r, t_{noise}) is false. The NCELoss is defined as:

$$NCELoss(\hat{O}, O_p, [N_0 \dots N_k]) = -\log \frac{\exp(sim(\hat{O}, O_p))}{\exp(sim(\hat{O}, O_p)) + \sum_{k \in N} \exp(sim(\hat{O}, N_k))}$$

where N_k are the projected noise samples, sim is the similarity function which can be the L2 norm or Cosine similarity, \hat{O} is the generated output vector and O_p is the projected vector.

The D distance function (2) for such a model is defined by the distance function used in the loss function. For L2Loss, it is the L2 norm, and in the case of NCELoss, we use $1 - sim$ function.

2.3 Span Masked Language Modeling

In Span Masked Language Modeling (SMLM), we model the equation 1 as a masked language modeling task. We tokenize and concatenate the triple (h, r, t) with a separator token between them, i.e., $[cls][h][sep][r][sep][t][sep]$. For the function $f_r(h, t) \Rightarrow r$, we mask all the tokens present in r , i.e., $[cls][h][sep][mask][sep][t][sep]$. We feed these tokens to a Transformer encoder Enc and use a feed forward network to unmask the sequence of tokens. Similarly, we mask h to learn f_h and t to learn f_t .

We train the same Transformer encoder to perform all the three functions. We use the cross-entropy loss to train the model:

$$CELoss(h, r, mask(t), t) = -\frac{1}{n} \sum_{i=1}^n \log_2 P_{MLM}(t_i | h, r, t_{1..t_i..t_n})$$

where P_{MLM} is the masked language modeling probability of the token t_i , given the unmasked tokens h and r and other masked tokens in t . Do note we do not do progressive unmasking, i.e., all the masked tokens are jointly predicted.

The D distance function (2) for this model is same as the loss function defined above.

3 Synthetic Graph Construction

In this section we describe our method to create a synthetic knowledge graph from a text corpus containing sentences. Not all types of knowledge are present in a structured knowledge graph, such as ATOMIC, which might be useful to answer questions. For example, the questions in QASC

dataset (Khot et al., 2019) require knowledge about scientific concepts, such as, “Clouds regulate the global engine of atmosphere and ocean.”. The QASC dataset contains a textual knowledge corpus containing science facts. Similarly, the Open Mind Commonsense (OMCS) knowledge corpus contains knowledge about different commonsense facts, such as, “You are likely to find a jellyfish in a book”. Another kind of knowledge about social interactions and story progression is present in several story understanding datasets, such as RoC-Stories and the Story Cloze Test (Mostafazadeh et al., 2016). To be able to perform question answering using these knowledge and KTL, we create the following two graphs: Common Concept Graph and the Directed Story Graph.

Common Concept Graph In order to create the Common Concept Graph, we extract noun-chunks and verb-chunks from each of the sentences using Spacy Part-of-Speech tagger (Honnibal and Montani, 2017). We assign all the extracted chunks as the vertices of the graph, and the sentences as the edges of the graph. To generate training samples for KTL, we assign triples (h, R, t) as (e_1, e_2, v_i) where v_i is the common concept present in both the sentences e_1 and e_2 . For example, in the sentence *Clouds regulate the global engine of atmosphere and ocean.*, the extracted concepts are *clouds, global engine, atmosphere, ocean* and *regulate*. The triplet assignment will be, [*Warm moist air from the Pacific Ocean brings fog and low stratus clouds to the maritime zone.*, *Clouds regulate the global engine of atmosphere and ocean.*, **clouds**]. We create two such synthetic graphs using the QASC science corpus and the OMCS concept corpus. Our hypothesis is this graph and the KTL framework will allow the model to understand the concepts common in two facts, which in turn allows question answering.

Directed Story Graph This graph is created using short stories from the RoCStories and Story Cloze Test datasets. This graph is different from the above graph as this graph has a directional property and each individual story graph is disconnected. To create this graph, we take each short story with k sentences, $[s_1, s_2, s_3, \dots, s_k]$ and create a directed graph such that, all sentences are vertices and each sentence is connected with a directed edge only to sentences that occur after it. For example, s_1 is connected to s_2 with a directed edge but not vice

versa. We generate triples (h, R, t) by sampling vertices (s_i, s_j, s_k) such that there is a directed path between the sentences s_i and s_k through s_j . This captures a smaller story where the head is an event that occurs before the relation and the tail. This graph is designed for story understanding and abductive reasoning using KTL framework.

Random Sampling There are around 17M sentences in the QASC text corpus, similarly there are 640K sentences in the OMCS text corpus. Our synthetic triple generation leads to a significantly large set of triples that is in order of 10^{12} and more. To restrict the train dataset size for our KTL framework, we randomly sample triples and limit the train dataset size to be at max 1M samples, we refer to this as Random Sampling.

Curriculum Filtering Here we extract the noun and verb chunks from the context, question and answer options present in the question answering datasets. We filter triples from the generated dataset and keep only those triples where at least one of the entities is present in the extracted noun and verb chunks set. This filtering is analogous to a real-life human examination setting where a teacher provides the set of concepts upon which questions would be asked, and the students can learn the concepts. We perform the sampling and filtering only on the huge Common Concept Graphs generated from QASC and OMCS corpus.

4 Datasets

We evaluate our framework on the following six datasets: SocialIQA (Sap et al., 2019b), aNLI (Bhagavatula et al., 2019), CommonsenseQA (Talmor et al., 2018), QASC (Khot et al., 2019), OpenBookQA (Mihaylov et al., 2018) and ARC (Clark et al., 2018). SocialIQA, aNLI and CommonsenseQA require commonsense reasoning and external knowledge to be able to answer the questions. Similarly, QASC, OpenBookQA, and ARC require scientific knowledge. Table 1 shows the dataset statistics and the corresponding knowledge graph used to train our KTL model. Table 2 shows the statistics for the triples extracted from the graphs. From the two tables we can observe our KTL triples have different number of words when compared to the target question answering tasks. Especially, where the context is significantly larger and human annotated as in SocialIQA, increasing the challenge for unsupervised learning.

	ARC-Easy	ARC-Chall	QASC	OpenBookQA	CommonsenseQA	aNLI	SocialQA
Train Size	2251	1119	8134	4957	9741	169654	33410
Val Size	570	299	926	500	1221	1532	1954
Test Size	2377	1172	920	500	1140	-	-
C Length	-	-	-	-	-	9	15
Q Length	19.4	22.3	13	12	14	9	6
A length	3.7	4.9	1.5	3	1.5	9	3
# of Option	4	4	8	4	5	2	3
KTL Graph	QASC-CCG	QASC-CCG	QASC-CCG	QASC-CCG	OMCS-CCG	DSG	ATOMIC

Table 1: Dataset Statistics for the seven QA tasks. Context is not present in five of the tasks. The KTL Graph refers to the graph over which we learn. CCG is the Common Concept Graph. DSG is the Directed Story Graph. C,Q,A is the average number of words in the context, question and answer. aNLI and SocialQA Test set size is hidden.

	ATOMIC	QASC-CCG	OMCS-CCG	DSG
Train Size	893393	1662308	914442	1019030
Val Size	10000	10000	10000	10000
H Length	11.2	10.5	9.6	10.3
R Length	6.5	10.3	9.4	10.2
T Length	2	1.5	2	10.4

Table 2: Dataset Statistics for the generated Triples. For QASC and OMCS it is after Curriculum Filtering. H,R,T length refers to average number of words. For CCG, we show for the $[e_i, e_j, v]$ configuration.

4.1 Question to Hypothesis Conversion and Context Creation

We can observe the triples in our synthetic graphs, QASC-CCG and OMCS-CCG contain factual statements, and our target question answering datasets have questions that contain *wh* words or fill-in-the-blanks. We translate each question to a hypothesis using the question and each answer option. To create hypothesis statements for questions containing *wh* words, we use a rule-based model (Demszky et al., 2018). For fill-in-the-blank and cloze style questions, we replace the blank or concat the question and the answer option.

For questions that do not have a context, such as in QASC or CommonsenseQA, we retrieve top five sentences using the question and answer options as query and perform retrieval from respective source knowledge sentence corpus. For each retrieved context, we evaluate the answer option score using equation 2 and take the mean score.

5 Experiments

5.1 Baselines

We compare our models to the following baselines.

1. **GPT-2 Large** with language modeling cross-entropy loss as the scoring function. We concatenate the context and question and find

the cross-entropy loss for each of the answer choices and choose the answer which has the minimum loss.

2. **Pre-trained RoBERTa-large** used as is, without any finetuning or further pre-training, with scoring same as our defined SMLM model. We refer to it as Rob-MLM.
3. **RoBERTa-large** model further fine-tuned using the original Masked Language Modeling task over our concatenated fact triples (h, r, t) , with scoring same as SMLM. We refer to it as Rob-FMLM.
4. **IR Solver** described in ARC (Clark et al., 2016), which sends the context, question and answer option as a query to Elasticsearch. The top retrieved sentence which has a non-stop-word overlap with both the question and the answer is used as a representative, and its corresponding IR ranking score is used as confidence for the answer. The option with the highest score is chosen as the answer.

5.2 KTL Training

We train the Knowledge Representation Learning (KRL) model using both L2Loss and NCELoss. For NCELoss we also train it with both L2 norm and Cosine similarity. Both the KRL model (365M) and SMLM model (358M) uses RoBERTa-large (355M) as the encoder. We train the model for three epochs with the following hyper-parameters: batch sizes [512,1024] for SMLM and [32,64] for KRL; learning rate in range: [1e-5,5e-5]; warm-up steps in range [0,0.1]; in 4 Nvidia V100s 16GB. We use the transformers package (Wolf et al., 2019). All triplets from the training graphs are positive samples. We learn using these triplets. For NCE, we choose k equal to ten, i.e., ten negative samples. We perform three hyper-parameter trials using ten percent of the training data for each model, and

Models	ARC-E \uparrow			ARC-C \uparrow			OBQA \uparrow			QASC \uparrow		ComQA \uparrow		aNLI \uparrow		SocIQa \uparrow	
Random	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0	12.5	12.5	20.0	20.0	50.0	51.0	33.3	33.3
GPT-2 L	30.5	29.1	29.4	23.5	25.1	25.0	32.0	26.6	27.8	12.3	13.2	36.4	37.2	50.8	51.3	41.2	40.8
RoB-MLM	29.8	29.6	29.0	24.8	25.0	25.0	24.8	24.4	25.0	12.8	17.6	23.6	24.8	51.6	52.2	35.6	34.5
RoB-FMLM	31.0	31.2	30.6	24.6	22.1	23.8	23.4	24.2	23.8	14.2	19.7	23.2	26.1	51.2	51.4	36.9	36.1
IR	29.4	30.4	30.2	18.4	20.3	21.2	31.4	29.4	28.8	18.6	19.4	24.6	24.4	53.4	54.8	35.8	36.0
KRL-L2	28.8	29.6	29.8	26.7	26.8	25.6	29.6	28.8	29.2	20.4	20.8	31.4	30.6	57.6	57.4	43.2	43.8
KRL-NCE-L2	32.4	31.8	30.6	27.2	27.5	26.8	33.2	31.6	32.8	22.6	23.1	33.4	33.8	59.3	60.5	46.4	46.2
KRL-NCE-Cos	<u>32.8</u>	<u>32.0</u>	<u>31.8</u>	<u>27.4</u>	<u>27.9</u>	<u>27.8</u>	35.6	34.8	34.4	<u>23.2</u>	<u>24.4</u>	<u>36.8</u>	<u>37.1</u>	<u>60.4</u>	<u>60.2</u>	<u>46.6</u>	<u>46.4</u>
SMLM	33.2	33.4	33.0	27.8	28.4	28.4	<u>34.4</u>	<u>34.6</u>	<u>33.8</u>	26.6	27.2	38.2	38.8	64.7	65.3	48.7	48.5
Self-Talk	N/A			N/A			N/A			N/A		32.4		N/A		46.2	
BIDAF Sup.	50.1	49.8		20.6	21.2		49.2	48.8		31.8		32.0		67.8		51.2	
RoBerta Sup.	85.0			67.2			72.0			61.8		72.1		83.2		76.9	

Table 3: Results for the Unsupervised QA task. Mean accuracy on Train, Dev and Test is reported. For Self-Talk and BIDAF Sup. we report the Dev and Test splits, for Roberta Sup. we report Test split. Test is reported if labels are present. **Best** scores, Second Best.

train models with three different seeds. For each of our experiments we report the mean accuracy of the three random seed runs, and report the standard deviation if space permits. Code is available [here](#).

6 Results and Discussion

6.1 Unsupervised Question Answering

Table 3 compares our different KTL methods with our four baselines for the six question answering datasets on the zero-shot question answering task. We use Hypothesis Conversion, Curriculum Filtering and Context Creation for ARC, QASC, OBQA and CommonsenseQA for both, the baselines as well as our models. We compare the models on the Train, Dev and Test split if labels are available, to better capture the statistical significance.

We can observe our KTL trained models perform statistically significantly better than the baselines. When comparing the different KRL models, the NCELoss with Cosine similarity performs the best. This might be due to the additional supervision provided by the negative samples as the L2Loss model only tries to minimize the distance between the generated and the target projections. When comparing different KTL instantiations, we can see the SMLM model performs the best overall. SMLM and KRL differ in their core approaches. Our hypothesis is the multi-layered attention in a transformer encoder enables the SMLM model to better distinguish between a true and false statement. In KRL we are learning from both positive and negative samples, but the model still under-performs. On analysis, we observe the random negative samples may make the training task biased for KRL. Our future work would be to utilize alternative negative

Model	QASC \uparrow	OBQA \uparrow	aNLI \uparrow	ComQA \uparrow	SocIQa \uparrow
RoBerta	44.5 \pm 1.2	47.8 \pm 1.4	68.8 \pm 1.3	46.4 \pm 1.5	44.4 \pm 1.2
RoB-MLM	43.6 \pm 0.6	49.4 \pm 0.8	67.1 \pm 0.8	43.2 \pm 0.8	46.8 \pm 0.6
KRL-NCE-Cos	48.2 \pm 0.9	51.2 \pm 0.6	73.4 \pm 0.9	49.5 \pm 1.1	58.6 \pm 0.8
SMLM	49.8 \pm 0.6	55.8 \pm 0.6	76.8 \pm 0.6	51.2 \pm 0.7	69.1 \pm 0.4
RoBerta-Sup	59.40	71.0	84.3	71.4	76.6

Table 4: Accuracy comparison of the KTL pre-trained RoBerta encoder when used for Few-shot learning Question Answering task on the Validation split.

sampling techniques, such as selecting samples that are closer in contextual vector space.

The improvements on ARC-Challenge task is considerably less. It is observed that the fact corpus for QASC, although contains a huge number of science facts, does not contain sufficient knowledge to answer ARC questions. There is a substantial improvement in SocialQA, aNLI, QASC and CommonsenseQA as the respective KTL knowledge corpus contains sufficient knowledge to answer the questions. It is interesting to note that for QASC, we can reduce the problem from an eight-way to a four-way classification, as our top-4 accuracy on QASC is above 92%. Our unsupervised model outperforms previous approaches such as Self-Talk (Shwartz et al., 2020). It approaches prior supervised approaches like BIDAF (Seo et al., 2017), and even surpasses it on two tasks.

6.2 Few-Shot Question Answering

Table 4 compares our KTL pre-trained transformer encoder in the few-shot question answering task. We fine-tune the encoder with a simple feedforward network for a n -way classification task, the standard question answering approach using RoBerta with n being the number of answer options, while

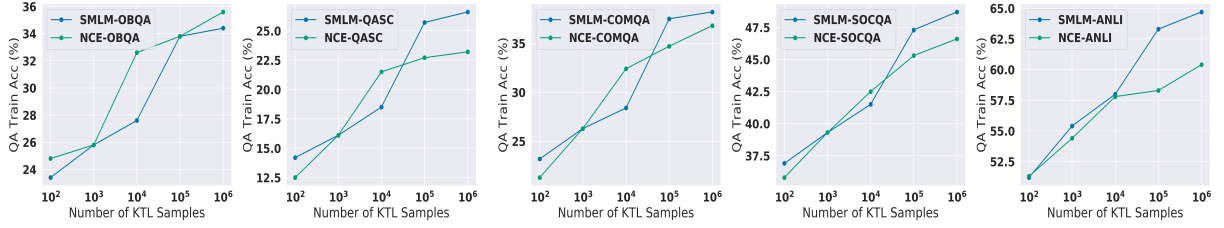


Figure 2: Effect of Increasing KTL training samples on the target zero-shot question answering Train split accuracy.

Model	QASC \uparrow	OBQA \uparrow	ComQA \uparrow	aNLI \uparrow	SocIQa \uparrow
SMLM - A	23.4 \pm 0.6	28.6 \pm 0.7	33.6 \pm 0.5	64.8 \pm 0.9	46.2 \pm 0.7
SMLM - Q	26.7 \pm 0.8	33.8 \pm 0.7	34.4 \pm 0.8	65.1 \pm 0.7	37.8 \pm 0.5
SMLM - C	22.8 \pm 1.1	29.8 \pm 1.3	31.9 \pm 0.9	64.9 \pm 0.8	47.1 \pm 0.8
SMLM - A*Q*C	27.2 \pm 0.6	34.6 \pm 0.8	38.8 \pm 0.6	65.3 \pm 0.7	48.5 \pm 0.6

Table 5: Accuracy comparison of using only Answer (A), Question (Q) and Context (C) distance scores.

training with only 8% of the training data. We train on three randomly sampled splits of training data and report the mean. We can observe our KTL pre-trained encoders perform significantly better than the baselines, and approaches the fully supervised model, with only 7.5% percent behind the fully supervised model on SocialQA. We also observe our pre-trained models have a lower deviation.

6.3 Ablation studies and Analysis

Effect of Context, Question, Answer Distance

In Table 5 we compare the effect of the three different distance scores. It is interesting to observe, in OpenBookQA, QASC and CommonsenseQA, the three datasets which don't provide a context, the model is more perplexed to predict the question when given a wrong answer option, leading to higher accuracy for only Question distance score. On the other hand, in aNLI all three distance scores have nearly equal performance. In SocialQA, the question has the least accuracy, whereas the model is more perplexed when predicting the context given a wrong answer option. This confirms our hypothesis that given a task predicting context and question can contain more information than discriminating between options alone.

Effect of Hypothesis Conversion, Curriculum Filtering and Context Retrieval

In Table 6 we observe the effect of hypothesis conversion, curriculum filtering and our context creation. Converting the question to a hypothesis provides a slight improvement, but a major improvement is observed when we filter our KTL training samples and keep only those concepts which are present in the target

Model	QASC \uparrow	OBQA \uparrow	ComQA \uparrow
SMLM - Hypo + CF	27.2 \pm 0.6	34.6 \pm 0.8	38.8 \pm 0.6
SMLM - Quesn + CF	26.5 \pm 1.2	32.2 \pm 1.1	35.4 \pm 1.3
SMLM - Hypo + Rand Sample	22.6 \pm 1.4	28.4 \pm 1.5	32.2 \pm 1.4
SMLM - Gold F+ Hypo + CF	72.4 \pm 0.8	75.2 \pm 0.7	-

Table 6: Effect of Question to Hypothesis Conversion (Hypo), Curriculum Filtering (CF) and providing the Gold Fact context on the Validation split.

question answering task, compared to when the KTL model is trained with a random sample of 1M. Curriculum filtering is impactful because there are a lot of concepts which are present in our source knowledge corpus, and the random sampled training corpus only contains 50% of the target question answering task concepts on an average. Another key thing to note in Table 6 is our KTL models can strongly perform like supervised models, when the gold knowledge context is provided, which are available in QASC and OpenBookQA. This indicates a better retrieval system for context creation can further improve our models.

Effect of Synthetic Triple corpus size Figure 2 compares our two modelling approaches when we train them with varying number of KTL training samples. NCE refers to our KRL model trained with NCELoss and Cosine similarity. We can observe our KRL model learns faster due to additional supervision, but the SMLM model performs the best, when trained with more samples. The performance tapers after 10^5 samples, indicating the models are overfitting to the synthetic data.

Error Analysis We sampled 50 error cases from each of our question answering tasks. Our KTL framework allows learning from knowledge graphs, that includes synthetic knowledge graphs. Both our instantiation, SMLM and KRL function as a knowledge base score generator, where given the inputs and a target the generator yields a score, how improbable is the target to be present in the knowledge base. Most of our errors are when all context,

question and answer-option have a large distance scores, and the model accuracy degenerates to that of a random model. This larger distance indicates the model is highly perplexed to see the input text. For aNLI and SocialIQA, we possess relevant context and our performance is significantly better in these datasets, but for other tasks we have another source of error, i.e., context creation. In several cases the context is irrelevant and acts as noise. Other errors include, when the questions require complex reasoning such as understanding negation, conjunctions and disjunctions; temporal reasoning such “6 am” being before “10 am”, and multi-hop reasoning. These complex reasoning tasks are required to answer a significant number of questions in the science and commonsense QA tasks. We also tried to utilize a text generation model, such as GPT-2, to generate and compare with ground truth text using our KTL framework, but preliminary results show the model is overfitting to the synthetic dataset, and has a significantly low performance.

Other Instantiations Our KTL framework can be implemented using other methods, such as using a Generator/Discriminator pre-training proposed in Electra (Clark et al., 2019), and sequence-to-sequence methods. The distance functions for sequence-to-sequence models can be similar to our SMLM model, the cross-entropy loss for the expected generated sequence. Discriminator based methods can adapt the negative class probabilities as the distance function. Studying different instantiations and their implications are some of the interesting future works.

7 Related Work

7.1 Unsupervised QA

Recent work on unsupervised question answering approach the problem in two ways, a domain adaptation or transfer learning problem (Chung et al., 2018), or a data augmentation problem (Yang et al., 2017; Dhingra et al., 2018; Wang et al., 2018; Alberti et al., 2019). The work of (Lewis et al., 2019; Fabbri et al., 2020; Puri et al., 2020) use style transfer or template-based question, context and answer triple generation, and learn using these to perform unsupervised extractive question answering. There is also another approach of learning generative models, generating the answer given a question or clarifying explanations and/or questions, such as GPT-2 (Radford et al., 2019) to perform unsu-

pervised question answering (Shwartz et al., 2020; Bosselut and Choi, 2019; Bosselut et al., 2019). In contrast, our work focuses on learning from knowledge graphs and generate vector representations or sequences of tokens not restricted to the answer but including the context and the question using the masked language modeling objective.

7.2 Use of External Knowledge for QA

There are several approaches to add external knowledge into models to improve question answering. Broadly they can be classified into two, learning from unstructured knowledge and structured knowledge. In learning from unstructured knowledge, recent large pre-trained language models (Peters et al., 2018; Radford et al., 2019; Devlin et al., 2018; Liu et al., 2019b; Clark et al., 2020; Lan et al., 2019; Joshi et al., 2020; Bosselut et al., 2019) learn general-purpose text encoders from a huge text corpus. On the other hand, learning from structured knowledge includes learning from structured knowledge bases (Yang and Mitchell, 2017; Bauer et al., 2018; Mihaylov and Frank, 2018; Wang and Jiang, 2019; Sun et al., 2019) by learning knowledge enriched word embeddings. Using structured knowledge to refine pre-trained contextualized representations learned from unstructured knowledge is another approach (Peters et al., 2019; Yang et al., 2019a; Zhang et al., 2019; Liu et al., 2019a).

Another approach of using external knowledge includes retrieval of knowledge sentences from a text corpora (Das et al., 2019; Chen et al., 2017; Lee et al., 2019; Banerjee et al., 2019; Banerjee and Baral, 2020; Mitra et al., 2019; Banerjee, 2019), or knowledge triples from knowledge bases (Min et al., 2019; Wang et al., 2020) that are useful to answer a specific question. Another recent approach is to use language model as knowledge bases (Petroni et al., 2019), where they query a language model to un-mask a token given an entity and a relation in a predefined template. In our work, we use knowledge graphs to learn a self-supervised generative task to perform zero-shot multiple-choice QA.

7.3 Knowledge Representation Learning

Over the years there are several methods discovered to perform the task of knowledge representation learning. Few of them are: TransE (Bordes et al., 2013) that views relations as a translation vector between head and tail entities, TransH (Wang et al., 2014) that overcomes TransE’s inability to model

complex relations, and TransD (Ji et al., 2015) that aims to reduce the parameters by proposing two different mapping matrices for head and tail. KRL has been used in various ways to generate natural answers (Yin et al., 2016; He et al., 2017) and generate factoid questions (Serban et al., 2016). The task of Knowledge Graph Completion (Yao et al., 2019) is to either predict unseen relations r between two existing entities: $(h, ?, t)$ or predict the tail entity t given the head entity and the query relation: $(h, r, ?)$. Whereas we are learning to predict including the head, $(?, r, t)$. In KTL, head and tail are not similar text phrases (context and answer) unlike Graph completion. We further modify TransD and adapt it to our KTL framework to perform zero-shot QA.

8 Conclusion

In this work, we propose a new framework of Knowledge Triplet Learning over knowledge graph entities and relations. We show learning all three possible functions, f_r , f_h , and f_t helps the model to perform zero-shot multiple-choice question answering, where we do not use question-answering annotations. We learn from both human-annotated and synthetic knowledge graphs and evaluate our framework on the six question answering datasets. Our framework achieves state-of-the-art in the zero-shot question answering task achieving performance like prior supervised work and sets a strong baseline in the few-shot question answering task.

9 Acknowledgements

The authors acknowledge support from the DARPA SAIL-ON program, and ONR award N00014-20-1-2332. The authors will also like to thank Tejas Gokhale, Arindam Mitra and Sandipan Choudhuri for their feedback on earlier drafts.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. [Synthetic QA corpora generation with roundtrip consistency](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Pratyay Banerjee. 2019. Asu at textgraphs 2019 shared task: Explanation regeneration using language models and iterative re-ranking. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods*

for Natural Language Processing (TextGraphs-13), pages 78–84.

- Pratyay Banerjee and Chitta Baral. 2020. Knowledge fusion and semantic knowledge ranking for open domain question answering. *arXiv preprint arXiv:2004.03101*.
- Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129.
- Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. [Commonsense for generative multi-hop question answering tasks](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.
- Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.
- Antoine Bosselut and Yejin Choi. 2019. Dynamic knowledge graph construction for zero-shot commonsense question answering. *arXiv preprint arXiv:1911.03876*.
- Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. [Reading Wikipedia to answer open-domain questions](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.
- Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. [Supervised and unsupervised transfer learning for question answering](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1585–1594, New Orleans, Louisiana. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. *arXiv preprint arXiv:1905.05733*.
- Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *ArXiv*, abs/1809.02922.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Bhuvan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. [Simple and effective semi-supervised question answering](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587, New Orleans, Louisiana. Association for Computational Linguistics.
- Alexander R Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. *arXiv preprint arXiv:2004.11892*.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.
- Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.
- Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 199–208.
- Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.
- Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696.
- Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*.
- Mandar Joshi, Kenton Lee, Yi Luan, and Kristina Toutanova. 2020. [Contextualized representations using textual encyclopedic knowledge](#).
- Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015.
- Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2019. Qasc: A dataset for question answering via sentence composition. *arXiv preprint arXiv:1910.11473*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.
- Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. [Unsupervised question answering by cloze translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

- Yankai Lin, Xu Han, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2018. Knowledge representation learning: A quantitative review. *arXiv preprint arXiv:1812.10901*.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-bert: Enabling language representation with knowledge graph. *arXiv preprint arXiv:1909.07606*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.
- Todor Mihaylov and Anette Frank. 2018. **Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.
- Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.
- Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. *arXiv preprint arXiv:1909.08855*.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.
- Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.
- Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. **Knowledge enhanced contextual word representations**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.
- Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191.
- Raul Puri, Ryan Spring, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. *arXiv preprint arXiv:2002.09599*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. *ArXiv*, abs/1811.00146.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. Socialliqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.
- Iulian Vlad Serban, Alberto Garcia-Duran, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598.
- Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. *arXiv preprint arXiv:2004.05483*.

- Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. [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)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.
- Damien Teney and Anton van den Hengel. 2016. Zero-shot visual question answering. *arXiv preprint arXiv:1611.05546*.
- Chao Wang and Hui Jiang. 2019. [Explicit utilization of general knowledge in machine reading comprehension](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2263–2272, Florence, Italy. Association for Computational Linguistics.
- Liang Wang, Sujian Li, Wei Zhao, Kewei Shen, Meng Sun, Ruoyu Jia, and Jingming Liu. 2018. [Multi-perspective context aggregation for semi-supervised cloze-style reading comprehension](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 857–867, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.
- An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019a. [Enhancing pre-trained language representations with rich knowledge for machine reading comprehension](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy. Association for Computational Linguistics.
- Bishan Yang and Tom Mitchell. 2017. [Leveraging knowledge bases in LSTMs for improving machine reading](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, Vancouver, Canada. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.
- Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. [Semi-supervised QA with generative domain-adaptive nets](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, Vancouver, Canada. Association for Computational Linguistics.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kgbert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.
- Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2972–2978.
- Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. [ERNIE: Enhanced language representation with informative entities](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.