A New Concept of Knowledge based Question Answering (KBQA) System for Multi-hop Reasoning

Yu Wang   Vijay Srinivasan   Hongxia Jin
Samsung Research America
{yu.wang1,v.srinivasan,hongxia.jin}@samsung.com

Abstract

Knowledge based question answering (KBQA) is a complex task for natural language understanding. Many KBQA approaches have been proposed in recent years, and most of them are trained based on labeled reasoning path. This hinders the system's performance as many correct reasoning paths are not labeled as ground truth, and thus they cannot be learned. In this paper, we introduce a new concept of KBQA system which can leverage multiple reasoning paths' information and only requires labeled answer as supervision. We name it as **M**utliple **R**easoning **P**aths KB**QA** System (MRP-QA). We conduct experiments on several benchmark datasets containing both single-hop simple questions as well as muti-hop complex questions, including WebQuestionSP (WQSP), ComplexWebQuestion-1.1 (CWQ), and PathQuestion-Large (PQL), and demonstrate strong performance.

1 Introduction

Knowledge-based question answering (KBQA) is the task of finding answers to questions by processing a structured knowledge base $\mathcal{KB}$. A $\mathcal{KB}$ consists of a set of entities $\mathcal{E}$, a set of relations $\mathcal{R}$, and a set of literals $\mathcal{S}$. A knowledge base fact is defined as (h, r, t), where $h \in \mathcal{E}$ is the head entity, $t \in \mathcal{E} \bigcup \mathcal{S}$ is the tail entity/literal, and $r \in \mathcal{R}$ is the directed relation between h and t. To answer a simple single-relation question (*i.e.* a 1-hop question) such as: *"Who is the president of the United States?"*, a typical KBQA system first identifies the entity (*i.e.* United States) and the relation (*i.e.* "president") asked in the question, and then searches for the answer entity by matching the entity-relation tuple $<$United States, president, ?$>$ over $\mathcal{KB}$.

While a single-hop question can be answered by searching a predicate relation in $\mathcal{KB}$, it is much harder to answer more complex multi-hop questions containing multiple entities and relations with constraints. For instance, for complex compositional questions, it is not easy to extract all the relations correctly together with their head and tail entities in the right order. For complex conjunction questions that requires a conjunction of multiple evidences, it is even more difficult to correctly extract all the reasoning paths included.

Most prior works on multi-hop KBQA focus on learning a single given ground truth reasoning path for each question, and outputting the most possible reasoning path during prediction (Zhou et al., 2018; Zhang et al., 2018; Yu et al., 2018; Lan et al., 2019). However, it is common that $\mathcal{KB}$ has many alternative paths leading to the correct answer, of various reasoning qualities. These alternative reasoning paths are usually not provided as ground truth by the human annotators.

For example, Figure 1 shows 7 reasoning paths $\mathbf{p}^n = e_0^n \rightarrow r_1^n \rightarrow e_1^n \rightarrow \cdots \rightarrow e_{ans}^n$ ($n = \{1, \ldots, 7\}$) leading to an answer set containing the correct answer *"West Lafayette"* for a given question *"What city is home to the University that is known for Purdue Boilermakers men's basketball?"*, but only the reasoning path $\mathbf{p}^1$ is labeled as the correct path in the dataset. A model trained with only $\mathbf{p}^1$ as supervision is likely to miss other paths which are also valid. For example, it will probably map a similar question *"What city is home to the stadium that is known for Los Angeles Lakers?"* to path $\mathbf{p}^1$, but fail to associate it with $\mathbf{p}^3$ or $\mathbf{p}^4$, because $\mathbf{p}^3$ or $\mathbf{p}^4$ contain different types of relations. However, $\mathbf{p}^1$ is a wrong reasoning path for that test question.

As the example shown in Figure 1, there are four paths ($\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3, \mathbf{p}^4$) pointing to the exact answer set containing only the answer entity, and thus can be treated as ground truth paths when training. Comparatively, reasoning paths $\mathbf{p}^5$ and $\mathbf{p}^6$ lead to a larger final entity set containing the correct answer

Q: What city is home to the University that is known for **Purdue Boilermakers men's basketball** ?
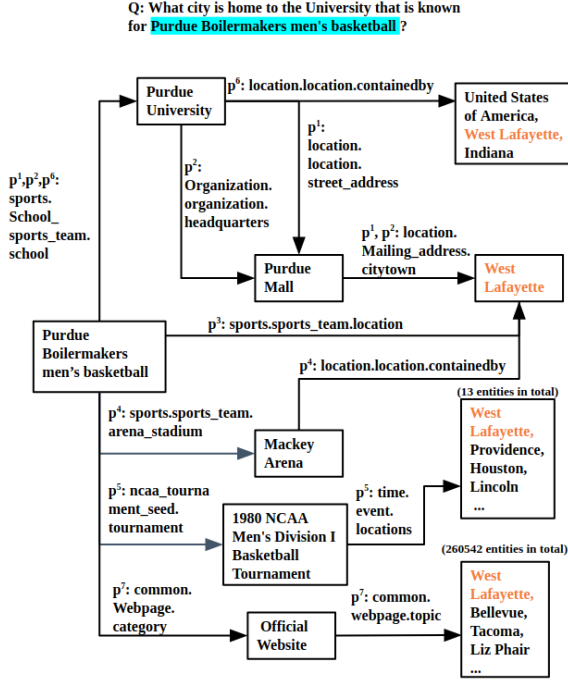
Figure 1: One QA example with Multiple Reasoning Paths (MRPs) from COMPLEXWEBQUESTION-1.1. The blue color highlighted is the extracted topic entity. Each square represents an entity, and the arrows represent the relations. Reasoning path $\mathbf{p}^1$ to $\mathbf{p}^4$ are the correct ones containing meaningful reasoning paths to the final answer. $\mathbf{p}^5$ and $\mathbf{p}^6$ are the "second choice" paths that generate a larger final answer set containing some wrong entities. $\mathbf{p}^7$ is the wrong one as its reasoning path is totally not interpretable and the answer set is huge.

"West Lafayette" but also other entities. These two paths can be considered as inferior to the top 4 paths; however, it is still worth including them in the training as a "second choice", as it is not difficult to extract the correct answer from final sets by additional post-processing. For example, a simple filter can be applied to filter out *"United States of America"* and *"Indiana"* from the predicted set, as they are not cities. Path $\mathbf{p}^7$ is bad because it is not interpretable, in addition to the final answer set being exaggeratedly large with invalid answers. Hence, path $\mathbf{p}^7$ should not be considered as a training path for this question. Unfortunately, it is not possible for any existing models to use multiple good/inferior paths, but not the bad ones, since current models are only trained with a single path for each question answer pair.

In this paper, we propose a novel multi-hop KBQA system, which can leverage the training information from multiple reasoning paths (MRPs)

without using any path annotations, we name it as **M**ultiple **R**easoning **P**aths KBQA System (MRP-QA). We model the reasoning path as a latent variable, and propose supporting training and prediction methods. The system can output diverse reasoning paths, and reward the "better" paths over the inferior ones by assigning "better" paths higher probabilities. Our method can be applied to most KBQA systems to predict the answer, and can be used with any model architecture. We achieve strong performance on three popular KBQA datasets. Experimental results show that our model performs especially well on multi-hop question, and in particular on complex questions that cannot be solved with a single reasoning path.

Our MRP-QA System does not need training paths annotation (only the question, and head and final entities), since it can sample the paths from the $\mathcal{KB}$ graph. This is of enormous pratical importance, because in practice questions and answers are easy to collect (sometimes for free), but path annotation is very labor-intensive and expensive.

2 Model

We first introduce some notations. For a given question q and its topic entity e_0 (identified by entity linking tool), a reasoning path is a sequence in the form $\mathbf{p} = (e_0, r_1, e_1, r_2, \cdots, e_{T-1}, r_T)$ that points to the answer entity $e_T = y$. That is, $\mathbf{p} \rightarrow e_T = y$. Each step (e_{t-1}, r_t, e_t) is a valid fact in the knowledge base $\mathcal{KB}$. Our goal is to build a model that can use multiple paths $\mathbf{p}$ to predict answer y given question q and topic entity e_0. In this section, we first present the design of our model architecture, and then explain the training and inference algorithms in detail.

2.1 MRP-QA Model Architecture

Figure 2 illustrates the architecture of our MRP-QA model. We model path probabilities using recurrent neural network with gated recurrent units (GRU). At a timestep t, the input hidden representations of GRU unit and predicted relation are denoted by h_{t-1} and r_t respectively. The model relies on the attention mechanism (Bahdanau et al., 2015) to produce a question context vector c_t. Specifically, all the words $w_0, w_1, \cdots, w_{|q|-1}$ in the given question q are first sent to a fixed embedding layer to acquire word embeddings $\varepsilon_w(w_0), \varepsilon_w(w_1), \cdots, \varepsilon_w(w_{|q|-1})$. Next we apply GRU to produce a temporary hidden state $h_t' =$
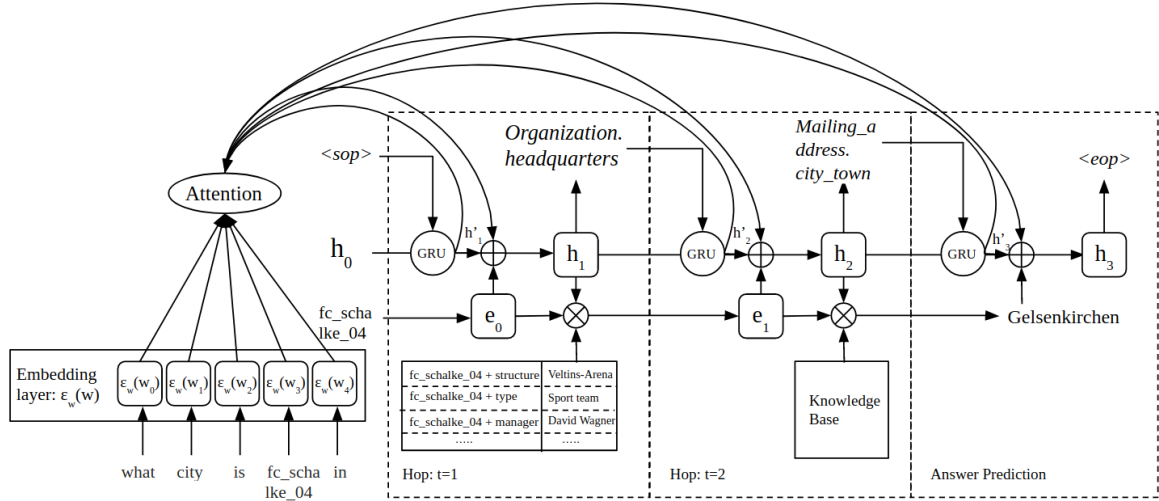
Figure 2: An illustration of how our MRP-QA model works with a QA pair *"What city is fc_schalke_04 in?"* and *Gelesenkirchen*. The entity linker extracts *fc_schalke_04* as the topic entity. We only show one possible paths here: r_1 is *Organization.headquarters* and r_2 is *Mailing_address.city_town*, our model can be used to output the probability of this given path. The symbol $\oplus$ represents concatenation, and $\otimes$ represents knowledge base lookup.

$GRU(h_{t-1}, r_{t-1})$, and then apply a parameterized feed-forward neural network a to calculate the similarity score $u_{tk} = a(h'_t, \varepsilon_w(w_k))$ of two inputs h'_t and $\varepsilon_w(w_k)$, and then these scores are normalized into attention weights $\alpha_{tk} = \frac{\exp(u_{tk})}{\sum_{0 \leq j \leq |q|-1} \exp(u_{tj})}$, which are used to produce the question context vector $c_t = \sum_{0 \leq j \leq |q|-1} \alpha_{tj} \varepsilon_w(w_j)$. In this fashion, word embeddings are combined in different ways based on attention weights to show different reasoning focuses at each timestep.

The model then concatenates temporary hidden state h'_t, entity representation $\varepsilon_e(e_{t-1})$, and question context c_t together, and passes the concatenation through a linear transformation f with ReLU activation to obtain the hidden state $h_t = ReLU(f([h'_t; \varepsilon_e(e_{t-1}); c_t]))$. This process is recurrently done until the model predicts a stop symbol $<eop>$[1]. Note that the vanilla RNN attention model only has h'_t and c_t when calculates h_t. We add entity representation into the calculation, since entity captures important information in the reasoning path.

2.2 Probabilities and Objective Function

The probability of predicting the k-th relation γ_k in $\mathcal{R}$ at timestep t is:

$$p(r_t = \gamma_k | q, e_0, r_1, \cdots, e_{t-1})$$
$$= \frac{\exp < h_t, \varepsilon_r(\gamma_k) >}{\sum_j \exp < h_t, \varepsilon_r(\gamma_j) >}$$

where ε_r is the embedding function, $<>$ is the dot product between two inputs.

Given the previous entity e_{t-1} and relation r_t, the next matched entity may not be unique when we query the knowledge base. For example, if e_{t-1}="united states", and r_t = "president of", then the resulting entity has 45 possibilities. Since we do not have additional constraints, all of them are equally likely to be selected, and hence we define:

$$p(e_t | e_{t-1}, r_t) \qquad (1)$$
$$= \begin{cases} 1/M & \text{if } e_t \text{ is one of the } M \text{ matched entities} \\ 0 & \text{if } e_t \text{ is not a matched entity} \end{cases}$$

Thus the probability of a path containing both entities and relations can be computed using the chain rule:

$$p(\mathbf{p}|q)$$
$$= \prod_{t=1}^{T-1} p(e_t | e_{t-1}, r_t) \prod_{t=1}^{T} p(r_t | q, e_0, r_1, \cdots, e_{t-1})$$
$$\qquad (2)$$

---

[1]This stop mechanism is the same as how it works in a vanilla RNN. Similarly, we also attach $<sop>$ to the beginning of each sequence to denote the start state. We will omit these symbols in formulas for simplicity.

We assume that there are multiple valid paths $\mathbf{p} \in \mathcal{P}$ that can lead to the correct answer y and they are not given by the annotator in the dataset. We treat these paths as hidden variables and we marginalize them out to compute the probability of getting the answer y:

$$
\begin{aligned}
&p(y|q) \\
&= \sum_{\mathbf{p} \in \mathcal{P}} [p(e_{T(\mathbf{p})} = y|\mathbf{p}, q)p(\mathbf{p}|q)] \\
&= \sum_{\mathbf{p} \in \mathcal{P}} \prod_{t=1}^{T(\mathbf{p})} [p(e_t|e_{t-1}, r_t)p(r_t|q, e_0, r_1, \cdots, e_{t-1})]
\end{aligned}
$$
(3)

where $\mathcal{P}$ is a set of all valid paths leading to the answer y, and $T(\mathbf{p})$ is the number of hops in the path $\mathbf{p}$.

To train our model, we would like to maximize the answer probability $p(y|q)$ using only the given answer for each training instance. To make prediction on each test case, we would like to find the answer y with the highest probability.

It is a novel way that we define answer probability as in (3) in the KBQA task. Most of the existing methods assume the availability of a single ground truth path annotation and aim to maximize the probability of the given path (Zhou et al., 2018). As we will demonstrate later in the Section 3.3, considering multiple paths leads to a better model performance.

2.3 System Training

In order to train our MRP-QA model by maximizing the marginalized answer probability given in (3), it requires summing over all valid reasoning paths from the topic entity to the answer entity in knowledge base. Thus computing this objective exactly can be intractable. As shown in the early example, some reasoning paths (R_5, R_6, R_7 in Figure 1) are not very helpful for training, thus should be either removed from training or assigned low probabilities. To achieve this goal, we first apply depth first search (DFS) algorithm with maximum 3 hops to get valid path candidates. The algorithm starts the traversal from the topic entity node, and ends at the answer entity node. All possible paths between the topic entity and the answer entity within 3 hops are extracted as candidates. We then set a threshold to remove paths which point to too many

entities at the last hop. To further filter out bad reasoning paths, we propose to dynamically choose reasoning paths deemed as most probable by the current model during training. The overall training procedure is summarized in Algorithm 1. Note that training with this algorithm does not require ground truth reasoning path label. Labeled reasoning path is a plus, but not necessary. If it is given, we can either include the ground truth paths in $\mathcal{P}$, or use them to initialize model training.

Algorithm 1: Training method for a MRP-QA System

Input : KBQA dataset
$(q^n, y^n, e_0^n), n = 1, 2, \cdots, N$,
Knowledge Base $\mathcal{KB}$,
Threshold k_1 and k_2.

Output : Trained model parameters

1 **foreach** *instance* (q^n, y^n, e_0^n) **do**
2   Use DFS algorithm to get a set of paths $\mathcal{P}^n$ from e_0^n to y^n.
3   Remove from $\mathcal{P}^n$ paths that point to more than k_1 entities.
4 **end**
5 **foreach** *batch* **do**
6   **foreach** (q^n, y^n, e_0^n) *in the batch* **do**
7   Get top k_2 paths in $\mathcal{P}$ sorted by $p(\mathbf{p}|q)$ based on current model: $\tilde{\mathcal{P}}^n = \{\mathbf{p}_1^n, \cdots, \mathbf{p}_{k_2}^n\}$
8   **end**
9   Update model parameters by maximizing
$$\sum_{(q^n, y^n, e_0^n)} \log \sum_{\mathbf{p} \in \tilde{\mathcal{P}}^n} p(y^n|\mathbf{p}, q^n)P(\mathbf{p}|q^n)$$
10 **end**

2.4 Prediction

During the prediction, we aim to select the answer y with the highest marginalized probability $p(y|q)$ as defined in (3). Similar to training, we need to approximate the sum with selected paths from $\mathcal{P}$. We use a modified beam search to find paths that have high probabilities. We add two constraints to standard beam search to only select the valid paths that match the knowledge base: (1) The first relation r_1 should connect to the topic entity e_0. (2) Each triple (e_{t-1}, r_t, e_t) should match a fact in KB. Given the set of paths collected as above, we can then collect a set of candidate answers that these paths point to. For each answer y, we evaluate

its probability $p(y|q)$ approximately using the collected paths, and among them we output the answer with the highest probability.

Additionally, we observe that it could be beneficial to de-emphasize the impact of the topic entity during prediction, as noted in (Li et al., 2016), which can improve inference performance by avoiding generating generic predictions and reducing overfitting. Specifically, instead of searching y^* that maximizes $p(y|q)$, we can find an answer that maximizes $\dfrac{p(y|q)}{p(y|e_0)}$, where $p(y|e_0)$ is the probability of getting the answer y when the question only contains the topic entity word. Mathematically, one can show that this is equivalent to maximizing the point-wise conditional mutual information $\text{PMI}((y; q \backslash e_0)|e_0)$ between y and $q \backslash e_0$ given e_0, where $q \backslash e_0$ stands for the question with the topic entity term removed. Further discussion can be found in Section 4.

3 Results and Analysis

3.1 Experimental Setup

We conduct experiments on 3 multi-hop KBQA datasets, WEBQUESTIONSP (WQSP) (Yih et al., 2015), COMPLEXWEBQUESTION-1.1 (CWQ) (Talmor and Berant, 2018), and PATHQUESTION-LARGE (PQL) (Zhou et al., 2018), and use the original train/dev/test split. WQSP is a dataset that has been widely used for relation extraction and end-to-end KBQA tasks, which contains 1 or 2 hops questions. CWQ dataset is designed to study complex questions by adding more constraints to questions in WEBQUESTIONSP. PQL is a small dataset used to study sequential questions. Its original release contains two subsets: PQL2H and PQL3H, which contains only 2-hop and 3-hop questions correspondingly. Chen et al. (2019) then combined these two subsets and renamed the unified dataset as PQL+. All of the three datasets use Freebase (Google, 2013) as the supporting knowledge base. Table 1 contains statistics of these datasets.

	#train	#valid	#test	max_hops	>1 path
WQSP	2677	297	1639	2	79.4%
CWQ	27639	3519	3531	6	83.4%
PQL2H	1275	159	160	2	12.5%
PQL3H	1649	206	207	3	45.2%
PQL+	2924	365	367	3	30.6%

Table 1: Statistics of datasets. To count the data percentage with more than one path, *i.e.* >1 path, we use graph search algorithm to calculate what percentage of QA pairs can be solved with MRPs.

For questions with multiple answers, we use each answer to construct a question-answer (QA) pair. For WQSP and CWQ, we build a subgraph in a similar way as in (Sun et al., 2018), in order to generate the entity and relation candidates. For PQL, the original paper provides a subgraph of the Freebase. We implement our model using TENSORFLOW-1.11.0 and choose S-MART (Yang and Chang, 2016) and AllenNLP (Gardner et al., 2017) as our entity linking tools. If multiple topic entities are extracted, we use each topic entity to construct a question-answer pair. We test three different graph embedding methods WORD2VEC (Mikolov et al., 2013), TRANSE (Bordes et al., 2013), and HOLE (Trouillon and Nickel, 2017), and decide to use TRANSE in our final experiment based on validation performance. The threshold k_1 is set to be: 15 plus the number of answers in the ground truth answer set, and k_2 is top 50%. We adopt the average F1 score and the set accuracy as our main evaluation metrics. It is worth noticing that: except our methods' results, all other experimental results are obtained from early published papers. Details of these models can be found from our referenced papers.

3.2 Experimental Results

In Table 2 we compare our MRP-QA method to state-of-the-art models. All comparisons are divided into two groups based on different training supervisions. The upper block shows methods that are only trained with final answer as supervision, and the second block contains methods using extra annotations such as parsing results of the query. Experimental results show that our MRP-QA model performs better than all other methods on WQSP.

Also although NSM and NSM_{+h} (2nd best result) only relies on answers to train their model, it requires many prior knowledges, such as a big vocabulary to train word embeddings and graph embeddings, type label of the entity and of the relation, and pre-defined templates. The experiments from their papers show that these knowledge play a very important role in the system, *e.g.* F1 score drops from 69.0 to 60.7 by not using the pretrained embeddings for NSM.

In contrast, our model supports a training method that takes only raw QA pairs and the facts in knowledge base, and does not rely on any additional labels and pre-defined knowledge.

To further disentangle the contribution of differ-

	WQSP	CWQ
STAGG_SP (Yih et al., 2016)	71.7	-
HR-BiLSTM (Yu et al., 2017)	62.3	31.2
KBQA-GST (Lan et al., 2019)	67.9	36.5
KV-MemNN* (Miller et al., 2016)	38.6	-
STAGG_answer* (Yih et al., 2016)	66.8	-
NSM* (Liang et al., 2017)	69.0	-
GRAFT-Net* (Sun et al., 2018)	62.8	26.0
PullNet*(Sun et al., 2019)	68.1	47.2
TransferNet*(Shi et al., 2021)	71.4	48.6
NSM$_{+h}$*(He et al., 2021)	74.3	48.8
MRP-QA-marginal_prob*	**74.9**	**49.9**

Table 2: We report F1 (%) on WQSP and CWQ test sets. Methods labeled with ∗ only require the final answer as the supervision, and they are directly comparable to our MRP-QA model. As references, We also report the performance of methods that requires extra supervisions in the first block.

Setting	Δ F1 (std)
MRP-QA $- entity\_in\_RNN$	-2.1 (0.21)
MRP-QA $- marginal\_prediction$	-1.8 (0.32)
MRP-QA $- inference\_in\_training$	-3.4 (0.15)
MRP-QA $- mutual\_information$	-1.8 (0.16)

Table 3: Feature ablation study on the dev set with a mean of 5 runs.

ent factors in our MRP-QA method, we present a feature ablation test on WQSP dataset shown in Table 3. The vanilla RNN structure only maintains a hidden state and the previous prediction in the loop. Here, we show the performance boost by considering entity features in KBQA task. Instead of using greedy algorithm or beam search to output the top prediction with the highest joint probability $P(y, \mathbf{p})$, we propose to make the prediction based on marginalized probability $P(y)$, which also improves the performance by 1.8%. In addition, we show the benefits of using inference during training (line 6 and 7 in algorithm 1) and mutual information objective (Section (2.4)). More discussions can be found in the Section 4.

3.3 Choices of paths

In the second set of experiment, we test our MRP-QA model with different objective functions and compare their results correspondingly. The objective functions are as defined in Table 4, where the paths used for training are given in the last column. The detailed explanations are given as following:

Single ground truth path. When one reasoning path is given for each QA pair in addition to the answer, we can train the model to fit the

given path and answer by maximizing $p(y, \mathbf{p}|q) = p(y|\mathbf{p}, q)p(\mathbf{p}|q)$. This objective ignores the fact that MRPs could be valid for the same answer (see Figure 1) and pushes all the probability mass to the single given one.

Single random path. Many existing methods require a ground truth path for each question in order to train the model. When only the ground truth answer but no path is given to each question, one can randomly sample a path that leads to the given answer and treat the sampled path as ground truth for training.

Multiple paths product. For many of the existing training methods which expect a single path leading to the answer as part of the input, it is also possible to make them incorporate multiple possible paths when no path annotation is given. The simplest way is to expand each (question, answer) pair into multiple training instances, each with a different path leading to the same answer, and then apply existing training method treating them as independent instances. This corresponds to the objective $\prod_{\mathbf{p} \in \mathcal{P}} p(y|\mathbf{p}, q)p(\mathbf{p}|q)$. This objective has an undesired consequence in practical model training: because of the multiplication operation, the model has to assign equally high probabilities to all given reasoning paths in order to maximize the product of the probabilities. If only some reasoning paths receive high probabilities while others receive low probabilities, the production will still be low. As a consequence, the model cannot differentiate bad reasoning paths from good ones by assigning distinguishable probabilities to them.

Multiple paths marginalization. Our proposed training objective replaces the multiplication operation by the summation operation, and this allows the model to concentrate only on good reasoning paths for each QA pair. It is easy to show that the model tends to assign high probability $p(\mathbf{p}|q)$ to a path $\mathbf{p}$ when the path leads to few possible answers and therefore the chance of getting the correct answer $p(y|\mathbf{p}, q)$ is high (see 2). Also, using Jensen's inequality, one can show that this marginal probability objective maximizes the answer probability directly which is the learning goal of KBQA task, while the previous one using product operation maximizes a lower bound.

We test different ways of choosing paths and defining training objectives on WQSP and CWQ datasets. We further divide the test samples into two groups, based on whether there exist multiple

Method	Objective	Path $\mathbf{p}$
single ground truth	$p(y|\mathbf{p}, q)p(\mathbf{p}|q)$	single ground truth path leading to y
single random	$p(y|\mathbf{p}, q)p(\mathbf{p}|q)$	single random path leading to y
multiple product	$\prod_{\mathbf{p}\in\mathcal{P}} p(y|\mathbf{p}, q)p(\mathbf{p}|q)$	all valid paths leading to y
multiple marginal (MRP-QA)	$\sum_{\mathbf{p}\in\mathcal{P}} p(y|\mathbf{p}, q)p(\mathbf{p}|q)$	all valid paths leading to y

Table 4: Different choices of paths and objectives.

	WQSP			CWQ		
	1 path	>1 path	all	1 path	>1 path	all
single ground truth	66.8	69.3	68.1	40.8	49.2	46.4
single random	65.7	64.1	64.8	40.8	46.9	44.9
multiple product	69.1	70.2	69.7	40.9	50.7	47.5
multiple marginal (MRP-QA)	73.0	76.3	74.9	43.7	53.0	49.9

Table 5: We break test set into two groups based on number of paths associated with them and report F1(%).

	PQL2H	PQL3H	PQL+
HR-BiLSTM (Yu et al., 2017)	97.5	87.9	92.9
IRN (Zhou et al., 2018)	72.5	71.0	52.9
ABWIM (Zhang et al., 2018)	94.3	89.3	92.6
UHop (Chen et al., 2019)	97.5	89.3	92.3
KV-MemNN* (Miller et al., 2016)	72.2	67.4	-
Our MRP-QA Method-marginal_prob*	**98.4**	**97.8**	**98.0**

Table 6: We report set accuracy (%) on PQL. Similar to Table 2, we use * to highlight the methods which only requires the answer as supervision.

possible paths between the topic entity and the answer based on KB. Table 5 show that our proposed method gives the best performance on both scenarios. The models trained with single path perform consistently worse than those trained with multiple paths. Using random path is worse than using the given ground truth path. Between two models trained using multiple paths, the result shows the advantage of using our proposed objective.

3.4 PathQuestion-Large

In the third set of experiments, we test our MRP-QA model on PATHQUESTION-LARGE (PQL) dataset. This dataset contains synthetic questions generated by templates, and is supported by a very small knowledge base (500,000 times smaller than the full freebase). Not surprisingly, we can see the average performance on this dataset is much better than it is on the other two datasets. Recall that PQL2H and PQL3H represents two subsets with only 2 hops and 3 hops questions respectively. Table 6 shows that our MRP-QA method's performance beats all the other approaches on all three subsets of PQL from 1% to 7.8% in terms of test accuracy. Especially the gap between our method to the previous state-of-the-art approach (*i.e.* UHop)

becomes larger when the number of hops increase from 2 to 3.

4 Case Study

Our model requires inference while using the current model to select training samples for next batch in training (see line 6 in Algorithm 1). This EM style training approach helps us filter out bad reasoning paths based on context information. For example, a sample question from WQSP is *who was the owner of kfc?*, the graph search algorithm can easily extract two "correct" paths starting from the topic entity *kfc* directing to the ground truth answer *Colonel Sanders*: *kfc* $\rightarrow$ *organization.organization.founders* $\rightarrow$ *Colonel Sanders* and *kfc* $\rightarrow$ *advertisingcharacters.product.advertising_characters* $\rightarrow$ *Colonel Sanders*. However, the second path is totally wrong given that the reasoning path is irrelevant to the given question. *Colonel Sanders* happens to be the advertising character of *kfc*, but this cannot be generalized to other cases. Without using the trained model to filter out this irreverent path, the model may learn incorrect map from *who is the owner...* to the relation *advertising_characters*. In our experiment, we observe that when we train our model with all reasoning paths generated from DFS algorithm without using this filtering strategy (*i.e.* $k_2 = inf$), the F1 score drops 3.4% as shown in Table 3. This shows the importance of using the filtering strategy.

Next we demonstrate the benefit of maximizing conditional mutual information instead of likelihood. A sample question in WQSP is *who did benjamin franklin get married to?*. We observe that there are 13 questions are using *Benjamin Franklin* as the topic entity in the training set, but most of them are related to his invention and none of them is about marriage. With such a strong prior on *Benjamin Franklin*, our experimental result shows that the model trained with maximum likelihood mistakingly maps this question to a path related to *invention*, while the model trained with mutual

Question: what state does romney live in?	Answer: Massachusetts	Topic entity: romney
SINGLE GROUND TRUTH	MULTIPLE PRODUCT	MULTIPLE MARGINAL (OUR)
.89:children	.29:education_institution/ state_province_region	**.83:places_lived/ location**
.06:government_positions/ jurisdiction_of_office	.25:places_lived/ location	**.12:government_positions/ jurisdiction_of_office**
.04:government_positions/ office_position_or_title	**.25:government_positions/ district_represented**	**.04:government_positions/ district_represented**
.00:government_positions/ district_represented	**.01:government_positions/ jurisdiction_of_office**	.01:place_of_birth/ state
.00:place_of_birth	.01:place_of_birth/ state	.00:education/ degree
.00:jurisdiction_of_office	.01:sibling/ place_of_birth	.00:election_campaigns

Table 7: A running examples from WEBQUESTIONSP dataset. We show the probability $P(r_0, \cdots, r_T | q)$ before the inferred relations. Paths that lead to the correct answers are highlighted in bold. We use / to split two relations. The three columns are corresponding to the results by using different training settings as it is in Table 4. Due to space limit, we only show the partial name of a relation in the example and the probability less than .01 is shown as .00. We do not show $P(e_0, \cdots, e_{T-1} | q)$ because they are not determined by our MRP-QA model.

information makes the correct prediction. Table 3 shows that we get a 1.8% performance boost by using mutual information.

We further show how generated probabilities look like with different choices of paths and objectives in Table 7. In the given example, only our MRP-QA method outputs the correct path, and one can also find that the top three results correspond to three different but correct reasoning processes. We observe that in many training questions *"live in"* co-occurs with word *"children"*, which explains why the first model makes wrong prediction. We can see that training with joint objective given a single relation path generates the most sharp relation path distribution, *i.e.* the gap between the top entity and the second one is larger than that using other objectives. It assigns most probability mass to the top relation path. In this case, the model does not have ability to identify multiple relation paths during inference. The other extreme is that the second model is trained with joint objective and MRPs, which distribute probabilities over many relation paths, hence the model cannot distinguish good relation paths from the bad ones. Between the above two extremes is the proposed marginal objective with MRPs, when the most probable path is assigned the largest probability, while the rest ones still get reasonable probability assignments.

5 Related Work

Most of the existing multi-hop KBQA systems approach this task by decomposing it into two subtasks: topic–entity linking and relation extraction. The topic–entity linking gives the system an entry point to start searching, and the relation extraction is used to search relation paths leading to the final answer. Following this track, a straightforward idea is to match the question to a candidate entity/relation directly via calculating the similar-ity between them (Zhang et al., 2018; Yu et al., 2018; Lan et al., 2019). This method is not ideal for multi-hop questions with long paths, because the number of candidate entity-relation combinations grows exponentially as the number of hops increases. To tackle this issue, methods are proposed to decompose the input question into several single-hop questions, and then use existing method to solve each simple question. The decomposition methods are based on semantic parsing (Abujabal et al., 2017; Luo et al., 2018) or templates (Ding et al., 2019). A similar idea is to encode the reasoning information hop by hop, and predict the final answer at the last hop (Miller et al., 2016; Zhou et al., 2018; Chen et al., 2019).

Another line of work has looked at solving KBQA task with only final answer as supervision. Liang et al. (2017) first propose to cast KBQA as a program generation task using neural program induction (NPI) techniques. They learn to translate the query to a program like logical form executable on the KB. As a follow up, Ansari et al. (2019) improves this idea by incorporating high level program structures. Both these NPI models do not require annotated relation path as supervision, but they need some prior knowledge to design the program templates. In other work, Min et al. (2019) recently proposed a latent variable approach which is similar to the one described here, but applied on text-based QA scenarios. The main difference between our work is that our method aims at finding multiple reasoning paths leading to the answer, while their method only focus on extracting single optimal solution. We employ inference during training to filter our irrelevant paths, while they use it to identify the optimal solution.

Besides knowledge based question answering models, there are also other question answering sytsems without using knowledge base/graph in-

formation. For examples, there are end-to-end question-answering models by using reinforcement learning(Wang and Jin, 2019), adversarial learning(Wang et al., 2021b, 2020b), and also the multi-model structure to handle the unknown entities in question answering(Wang et al., 2018, 2020a). Coreference understanding (Wang et al., 2021a) is also another research direction in designing question-answering systems.

6 Conclusion

In this paper, We introduce a novel MRP-QA knowledge based question answering system which can leverage information from MRPs. To train our model, we use a marginalized probability objective function. Experimental results show that our model achieve strong performance on popular KBQA datasets.

References

Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated template generation for question answering over knowledge graphs. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1191–1200.

Ghulam Ahmed Ansari, Amrita Saha, Vishwajeet Kumar, Mohan Bhambhani, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2019. Neural program induction for KBQA without gold programs or query annotations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4890–4896.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. Uhop: An unrestricted-hop relation extraction framework for knowledge-based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 345–356.

Jiwei Ding, Wei Hu, Qixin Xu, and Yuzhong Qu. 2019. Leveraging frequent query substructures to generate formal queries for complex question answering. *CoRR*, abs/1908.11053.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Google. 2013. Freebase data dumps. https://developers.google.com/freebase/data.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 553–561.

Yunshi Lan, Shuohang Wang, and Jing Jiang. 2019. Knowledge base question answering with topic units. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5046–5052.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119.

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

Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Q. Zhu. 2018. Knowledge base question answering via encoding of complex query graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2185–2194.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 1400–1409.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. *CoRR*, abs/1909.04849.

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. *arXiv preprint arXiv:2104.07302*.

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.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242.

Alon Talmor and Jonathan Berant. 2018. Repartitioning of the complexwebquestions dataset. *CoRR*, abs/1807.09623.

Théo Trouillon and Maximilian Nickel. 2017. Complex and holographic embeddings of knowledge graphs: A comparison. *CoRR*, abs/1707.01475.

Yu Wang and Hongxia Jin. 2019. A deep reinforcement learning based multi-step coarse to fine question answering (MSCQA) system. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, USA, January 27 - February 1, 2019*, pages 7224–7232. AAAI Press.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 309–314. Association for Computational Linguistics.

Yu Wang, Yilin Shen, and Hongxia Jin. 2020a. A bi-model approach for handling unknown slot values in dialogue state tracking. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 8019–8023. IEEE.

Yu Wang, Yilin Shen, and Hongxia Jin. 2020b. An interactive adversarial reward learning-based spoken language understanding system. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 1565–1569. ISCA.

Yu Wang, Yilin Shen, and Hongxia Jin. 2021a. An end-to-end actor-critic-based neural coreference resolution system. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 7848–7852. IEEE.

Yu Wang, Yilin Shen, and Hongxia Jin. 2021b. Multi-step spoken language understanding system based on adversarial learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 7673–7677. IEEE.

Yi Yang and Ming-Wei Chang. 2016. S-MART: novel tree-based structured learning algorithms applied to tweet entity linking. *CoRR*, abs/1609.08075.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1321–1331.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 571–581.

Yang Yu, Kazi Saidul Hasan, Mo Yu, Wei Zhang, and Zhiguo Wang. 2018. Knowledge base relation detection via multi-view matching. In *New Trends in Databases and Information Systems - ADBIS 2018 Short Papers and Workshops, AI*QA, BIGPMED, CSACDB, M2U, BigDataMAPS, ISTREND, DC, Budapest, Hungary, September, 2-5, 2018, Proceedings*, pages 286–294.

Hongzhi Zhang, Guandong Xu, Xiao Liang, Tinglei Huang, and Kun Fu. 2018. An attention-based

word-level interaction model: Relation detection for knowledge base question answering. *CoRR*, abs/1801.09893.

Mantong Zhou, Minlie Huang, and Xiaoyan Zhu. 2018. An interpretable reasoning network for multi-relation question answering. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2010–2022.